

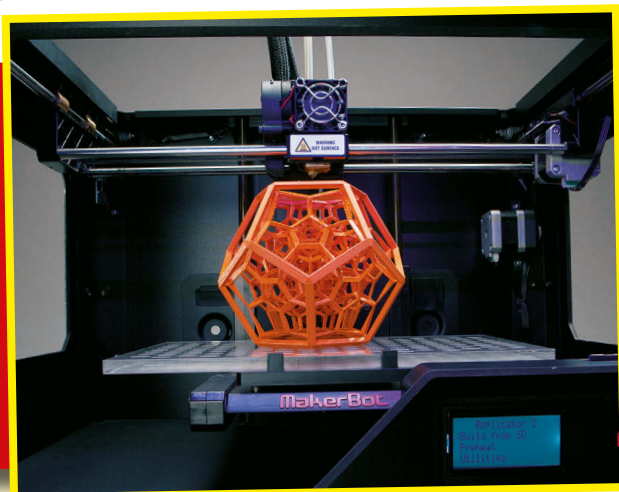


Développez pour tablettes !

*Android, iOS,
Windows 8*



100% GEEK



Les imprimantes 3D

Formation

La guerre des écoles

- **42** : le pari d'une école gratuite et sans diplôme !
- **Epita, Epitech, ETNA** : l'éducation en mode projet



- **Google I/O** et Android Studio
- Comprendre **/proc de Linux**
- Gérer la persistance **iOS / Windows 8**
- Utiliser un service **Hadoop**

Printed in EU - Imprimé en UE - BELGIQUE 6,45 €
SUISSE 12 FS - LUXEMBOURG 6,45 € DOM Surf 6,90 €
Canada 8,95 \$ CAN - TOM 940 XPF - MAROC 50 DH

M 04319 - 164 - F: 5,95 €



WINDEV

ATELIER DE GÉNIE LOGICIEL
PROFESSIONNEL



DÉVELOPPEZ VOS APPLIS POUR TABLETTE 10 FOIS PLUS VITE

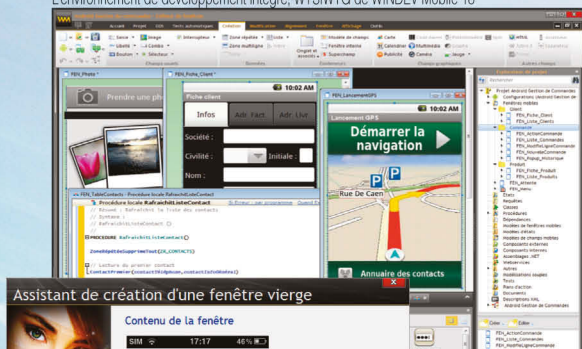
* IPAD, IPHONE, ANDROID, WINDOWS 8 & RT

DÉVELOPPEZ TRÈS VITE VOS APPLICATIONS MOBILES
(TABLETTE & SMARTPHONE) AVEC WINDEV MOBILE 18 :
EN GÉNÉRAL QUELQUES JOURS SUFFISENT .

VOUS POSSEDEZ UN EXISTANT WINDEV ? IL EST PORTABLE
DIRECTEMENT : RETAILLEZ LES FENÊTRES, RECOMPILÉZ.

BASE DE DONNÉES SQL INCLUSE.

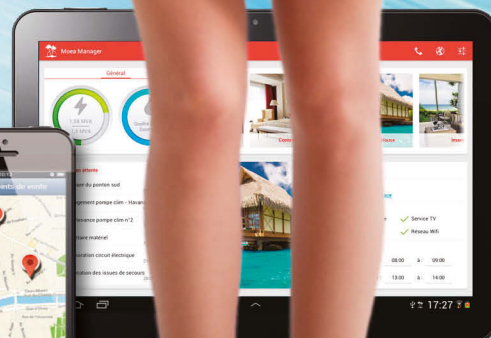
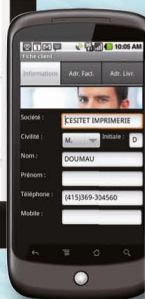
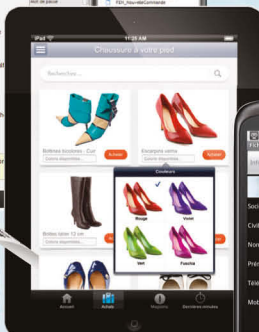
L'environnement de développement intégré, WYSIWYG de WINDEV Mobile 18



WINDEV 18

Atelier de développement
multi-plateformes: Windows
8, RT, 7, Vista, XP..., Linux,
Mac, Internet, Cloud,
Android, iPhone, iPad...

Vos applications sont compatibles



**VERSION
EXPRESS
GRATUITE**
Téléchargez-la !

DEMANDEZ VOTRE DOSSIER GRATUIT Sans engagement

260 pages - 100 témoignages - DVD Tél: **04.67.032.032** info@pcsoft.fr

www.pcsoft.fr



Fournisseur Officiel de la Préparation Olympique

La NSA dévoile son petit manuel 2006 sur la recherche Internet

« **U**ntangling the web » (démêler le Web) est un document confidentiel destiné à un usage strictement interne à la NSA, l'agence américaine de sécurité. Mais une décision de justice a obligé l'agence à le rendre public ! Objectif : trouver les (bonnes) informations nécessaires sur Internet en utilisant aux mieux les arcanes du réseau. Il est d'ailleurs décrit comme un labyrinthe et ce document serait un fil d'Ariane. Pour l'auteur, ou les auteurs, ce livre doit aider un utilisateur à comprendre comment se servir plus efficacement d'Internet et pourquoi Internet est une ressource incroyable (à exploiter). Le rapport pointe du doigt le fait qu'Internet change constamment, qu'une partie des pages seulement est réellement indexée par les moteurs de recherche.

Basiquement, il faut déjà maîtriser deux notions : disposer d'une bonne liste de marque-pages et acquérir l'expérience de la recherche des autres personnes. Bien entendu, le document aborde Google, Yahoo, Live Search, Gigablast mais aussi Exalead et Ask.

Une des premières notions à acquérir est de savoir comment fonctionne un moteur de recherche. Et on se rend compte en quelques minutes que ce « *Untangling the web* » est une petite mine d'or pour trouver des ressources. Rien que sur la partie moteur de recherches, une vingtaine de références (sites, blogs...) sont données. Nous poursuivons notre découverte par les sites de metarecherches.

L'auteur pointe du doigt que ces outils n'utilisent pas les moteurs les plus performants et que leurs performances

sont donc à relativiser. Mais ils peuvent servir comme première recherche pour appréhender le problème de la taille de la recherche. Et là encore, on nous propose plusieurs liens intéressants : Clusty, Jux2, Dogpile, Mamma, etc. Avec à chaque fois, des conseils, une méthode d'utilisation.

UN LIEU DE FRUSTRATIONS ?

Lent, peu disponible, frustrant, difficile à utiliser, etc. Internet n'est pas toujours simple à utiliser pour trouver la bonne information. Et nous le constatons chaque jour. La NSA le confirme (merci à eux !). Mais dans le même temps, Internet permet d'accéder à une multitude de ressources, de media et d'interroger des sites dans le monde entier...

Quelques règles de base pour vous en sortir :

Règle 1 : utilisez le bon outil pour faire le job !

Règle 2 : laissez d'autres personnes faire le travail pour vous autant que possible (utilisez leurs guides, leurs FAQ)

Règle 3 : bookmarquez constamment les pages, organisez-les et archivez-les comme si votre vie en dépendait !

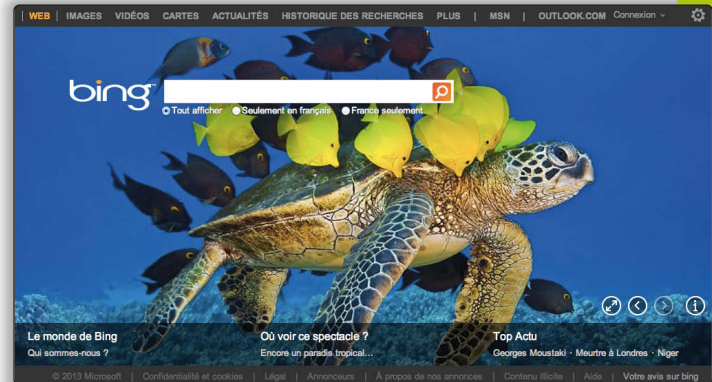
Règle 4 : utilisez plusieurs moteurs de recherche.

Règle 5 : lisez les instructions

L'auteur martèle fortement la règle 4 : utiliser un seul moteur de recherche est une erreur à ne pas faire ! Car chaque moteur a ses spécificités, ses propres performances. « *Par rapport aux répertoires et métamoteurs, les moteurs de recherche individuels offrent une plus grande souplesse et beaucoup*

plus d'options, comme la capacité de recherche en utilisant des expressions booléennes. » (extrait page 45). Bref, certains moteurs peuvent aller très loin dans les outils et paramètres de recherche. Mais parfois, il faut aller chercher les spécificités dans les instructions.

Internet est-il le meilleur endroit pour démarrer (une recherche) ?



ET APRÈS ?

C'est là que le document devient intéressant, à partir de la page 47. Le « guide » décrit les bonnes pratiques pour chaque moteur de recherche : comment il fonctionne, la personnalisation des préférences, la recherche basique (et les limites), la recherche avancée (et pas uniquement sur les pages HTML), utiliser des fonctions spéciales de recherche (ex. : dictionnaire, traduction...), les techniques de hacks pour chaque moteur...

Pour bien hacker, il faut regarder dans les API, les documentations techniques. Ce sont des mines d'informations et d'outils comme le Hub Finder (Yahoo), comment manipuler les caches de moteur (le copier localement par exemple, faire sauter la limitation des 32 mots). Le guide conseille aussi d'aller voir dans les services annexes des moteurs car ils peuvent aider à aller plus loin, à mieux cerner la recherche.

Mais ce livre ne se limite pas à la simple recherche, il aborde aussi la messagerie, la vidéo, la téléphonie, les podcast, les flux sociaux, les whois des domaines, etc. avec à chaque fois, des conseils, des exemples, et de nombreux liens de ressources.

En 642 pages, la NSA offre une bonne synthèse de l'état de l'art des techniques de recherches sur Internet et on peut très rapidement s'en servir pour du cyber-hacking. A consulter d'urgence.

Adresse du document :

http://www.nsa.gov/public_info/_files/Untangling_the_Web.pdf

BUZZWORD

La NSA dévoile son petit manuel 2006
sur la recherche Internet.....3

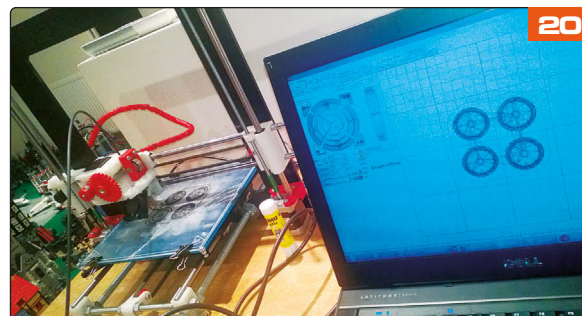
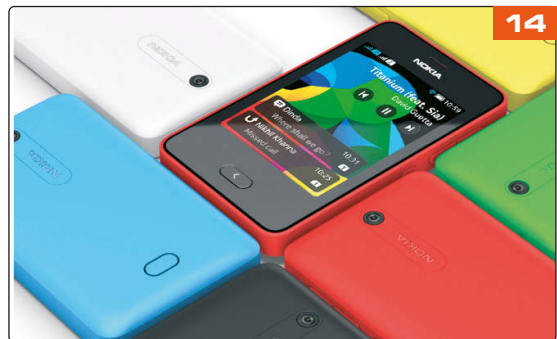
ACTUS

Actu en bref6
Retour sur Devovx France 2013 : les interviews8
Google I/O : IDE Google, nouvelles cartes !10

RESSOURCES

Événements, agenda.....12

MATERIEL



DEVELOPPEUR DU MOIS

Nathanaël Marchand :
aucun jour ne ressemble à un autre18

VEILLE TECHNO

Impressionné par l'impression 3D ?20

SECURITÉ

Les URL courtes24

CLOUD

Choisir son PaaS Java26

OUTILS

Les outils indispensables du développeur.....30

DOSSIER

Développer pour les Tablettes



HTML vs Code natif vs Hybride37
Développer des applications Android pour tablettes40
Je débute sur Android et iOS.....44
Découverte de Windows 8.....50
Créer des applications hybrides avec PhoneGap54

A lire dans le prochain numéro

N°165-Juillet/Août 2013, à paraître le 29 juin 2013

• Leap Motion

Contrôler son PC avec la main

• IE10

Tout savoir sur le navigateur
et les outils pour les développeurs

• Webmaster

Déployer un serveur web

Editeur : Go-02 sarl, 21 rue de Fécamp 75012 Paris - diff@programmez.com.
Rédaction : redaction@programmez.com - **Directeur de la Rédaction** : Jean Kaminsky. **Rédacteur en Chef** : François Tonic - f tonic@programmez.com.
Ont collaboré à ce numéro : F. Mazué, S. Saurel, C. Calais. **Experts** : F. Bellahcene, P.H. Gache, V.T. Saluzzo, F. Muiseroux, B. Dechoux, L. Rebours, C. Autexier, T. Cavin, S. Civetta, Y. Grenzinger, T. Guérin, L. Ellerbach, A. Talavera, F. Dupont, A.-L. Agoyé, A. Bouadma, L. Probst, G. Bougenière, C. Villeneuve
Illustration couverture : D.R.

Publicité : Régie publicitaire, K-Now sarl.
Pour la publicité uniquement : Tél. : 01 41 77 16 03 - diff@programmez.com.
Dépôt légal : à parution - Commission paritaire : 0717K78366 ISSN : 1627-0908. Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles Belgique.
Directeur de la publication : J-C Vaudecrane
Ce numéro comporte un encart jeté Component Source sur les abonnés et une partie du tirage.

CARRIERE

42 : born to code58

EPITA, Epitech, ETNA :
l'éducation en mode projet60

JE DEBUTE □□□□□

La persistance connectée
dans les applications mobiles62Une application Windows
Store réactive en 5 étapes68

CAS D'USAGE

Développer un job Map/Reduce pour Hadoop71



PRATIQUE

Comment choisir une stratégie
de Build et l'intégrer dans le Cloud74Comprendre et exploiter le système
de fichiers /proc de Linux77

BUGTRACK

Spring Security : Détecter les timeouts
de session utilisateur lors d'appels Ajax81La hiérarchie
des développeurs

Avertissement : je ne suis pas responsable des dépressions post-édo, ni des changements de carrières et encore moins d'un usage abusif de quiches.

Parfois, en facebookant avec des amis connectés, on découvre des posts énormes. Et là, impossible de ne pas en parler à Programmez !. Quelle est donc la hiérarchie des langages et surtout des développeurs entre eux. Certains se sentent-ils supérieurs aux autres ? C'est une question que je me pose de temps en temps en regardant le « temps restant » (enfin quand le système daigne l'indiquer correctement).

Steve Rowe sur son blog MSDN a relayé un schéma hiérarchique de Luke Welling (<http://goo.gl/CIOZb>). Et en lisant, honnêtement, cela m'a bien fait rire. Bien entendu, précaution d'usage : le prendre avec humour et non pas au 1er degré.

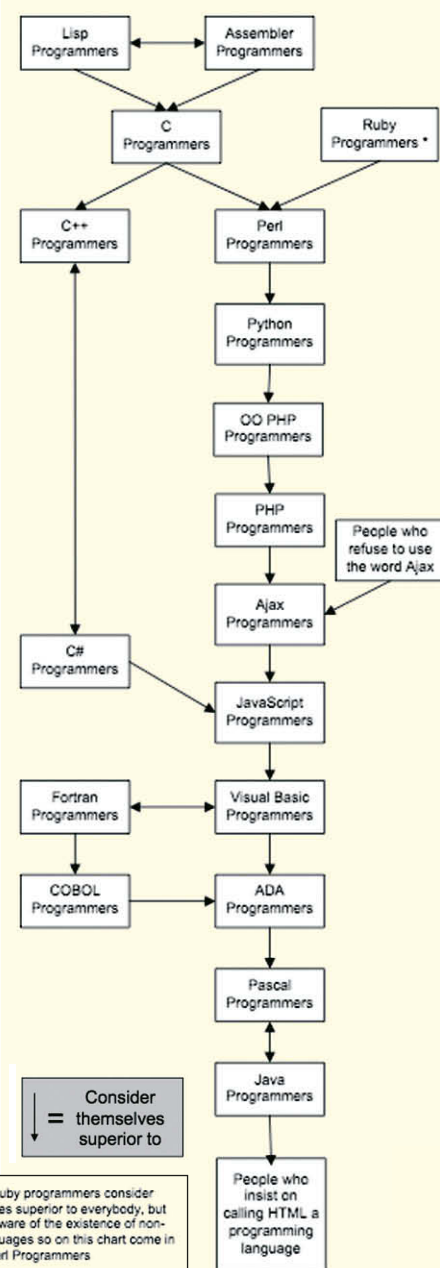
Cette hiérarchie de la considération est très amusante. Tout en haut, nous avons, enfin ceux qui se considèrent les meilleurs au monde, les programmeurs Lisp, Assembleur, puis C et C++. Ah le fameux développeur Assembleur qui manipule le plus bas niveau.

La secte Ruby

Je me souviens encore des manipulations d'adresses sur ResEdit (pour les connaisseurs). Visiblement, une secte a envahi l'univers Ruby car ils se croient supérieurs à tout le monde et aucun autre langage n'existe et surtout pas les langages non-web. Remarquez, un développeur C++ ou Java dit souvent la même chose. « *Quoi, un autre langage que le mien existe dans le monde ? Non ? tu déconnes ?* ». Et c'est du vécu. Que mon binaire m'en soit témoin.

Après, il y a des cas désespérés qui devraient faire une cure de Cobol ou de C# : ceux qui persistent à appeler HTML, un langage de développement ! Enfin là, y'a plus grand chose à faire, à part leur injecter du Node.JS. Bref, mieux vaut ne pas dépasser les 2 ou 3 premières cases (du haut de la hiérarchie).

Bon après, vous pouvez toujours passer au niveau ++ du geek avec la fameuse hiérarchie geek (Brunching Shuttlecocks) et là, vous ne serez pas déçus du voyage (re-attention aux yeux). Sinon, optez pour les 56 geeks ! Très drôle aussi...

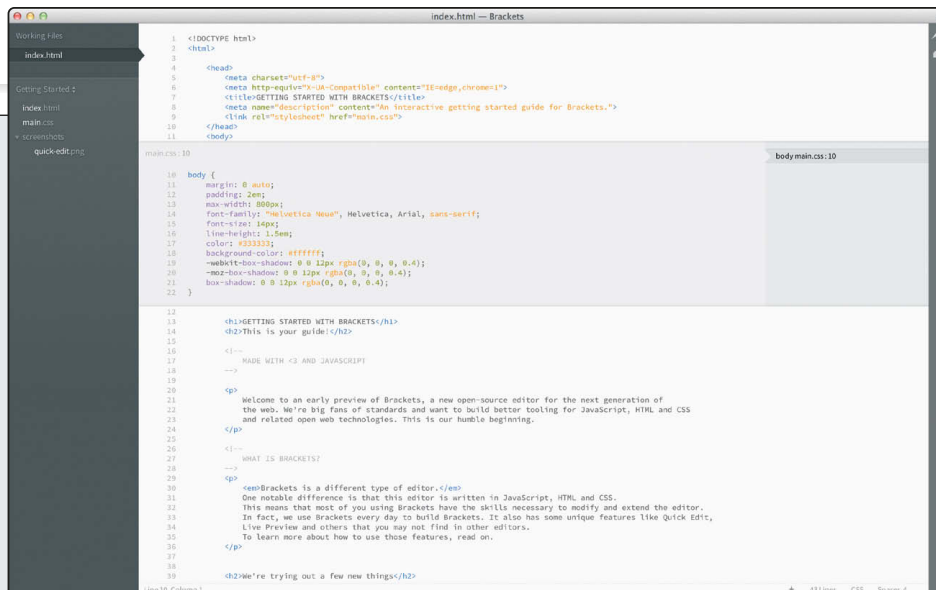


#dev geek
@programmez !

» Adobe mise tout sur la Creative Cloud !

L'éditeur Adobe a annoncé l'arrêt des boîtes pour ses logiciels au-delà de la Creative Suite 6. Cette version sera donc la dernière disponible en licence perpétuelle. Il faudra passer par la Creative Cloud dont les nouvelles versions seront disponibles courant juin. La Creative Cloud est disponible sous abonnement et valable sur deux postes (pour les installations). L'éditeur annonce cependant la continuation du support de la CS 6. Adobe met en avant les services supplémentaires : synchronisation des données, espace de stockage en ligne, gestion des versions des fichiers, digital publishing suite, accès aux polices de caractères, etc. En revanche, toujours aucune « suite » dédiée aux développeurs.

L'éditeur continue à investir et à développer des outils HTML 5 / CSS 3 / JavaScript. En mars dernier, une nouvelle version du toolkit for CreateJS était lancée. Il permet d'exporter des projets Flash vers HTML 5. Flash n'est pas abandonné. Les nouvelles versions



des outils ont été en grande partie réécrites. Mais aujourd'hui, l'avenir pour l'éditeur n'est plus (uniquement) sur Flash. Adobe l'a bien compris en multipliant les outils pour les web designer et les web développeurs autour du Responsive Design, de HTML 5, des tablettes. On dispose ainsi d'un éditeur HTML, CSS et JavaScript 100 % open source, du projet Brackets ou encore d'un puissant outil de tests des projets Responsive, Edge Inspect (la gamme Edge est

déclinée en plusieurs outils dédiés design et développeurs). Sur le support de Windows 8 / Windows Phone 8, il se fera selon les outils. Par exemple, sur Edge Inspect, le support de Windows 8 se fera quand le marché décollera. Sur Phonegap, l'éditeur est ravi de son dynamisme et de la communauté autour de l'outil. Mais une des difficultés avec PhoneGap est la conception des interfaces. Pour aider le développeur, Adobe propose PhoneGap Build.

» **Google** livre la version 1.1 de son langage Go. Celui-ci est disponible depuis un an mais il a bien du mal à intéresser les développeurs. Cette v1.1 propose quelques nouveautés : meilleures performances, optimisation de la compilation et du linker, amélioration du garbage collector. La syntaxe a été légèrement modifiée. Pour en savoir plus : <http://blog.golang.org/2013/05/go-1.1-is-released.html>

» Oracle a confirmé les retards de Java 8 et des futures versions.

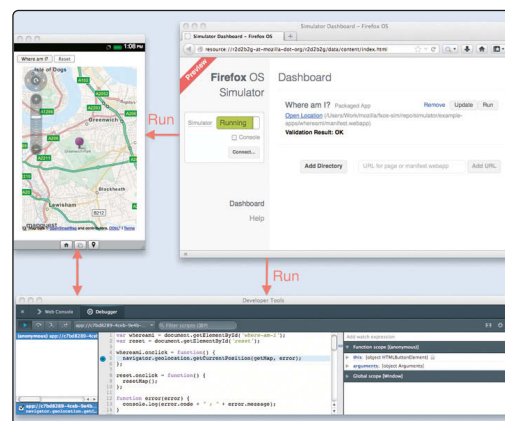
Suite à diverses fuites de sécurité et au retard de développement, Oracle a confirmé un

décalage d'agenda sur Java 8. Désormais, la version finale n'est pas attendue avant mars 2014, au lieu de septembre 2013. Java 9 est repoussé à début 2016 (au mieux). Cette version doit inclure le projet Jigsaw qui devait initialement être intégré à Java 8.

» **Enalean**, l'éditeur de Tuleap Open ALM annonce une nouvelle version de sa forge logicielle. Cette version inclut : une gestion de projet Agile, un plugin Eclipse Mylyn, l'intégration de Gerrit et une compatibilité Debian. « Cette nouvelle version de Tuleap est le résultat d'un an de collaboration avec nos clients. Nous co-contruisons la roadmap de Tuleap avec eux. En mettant en place les

approches Agile et DevOps, nous travaillons étroitement avec nos clients pour les aider à identifier et prioriser leurs besoins. Nous livrons tous les mois une mise à jour immédiatement utilisable, disponible pour tout le monde » explique Manuel Vacelet, directeur technique chez Enalean.

» Mozilla améliore les outils de développement pour Firefox OS.



La fondation a livré en début mai la version 3 de son simulateur dédié à son système mobile. Elle incorpore les dernières versions de Gaia et du navigateur, une simulation d'une API de géo-localisation, simulation de la rotation, validation du Manifest, etc. pour en savoir plus : <http://goo.gl/bxWWW>

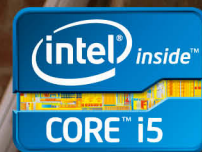
» AMD pousse la fondation HSA sur le multicœur.

Heterogeneous System Architecture (HSA) a pour ambition d'aider, de faciliter l'usage de systèmes multicœurs (CPU, GPU, DSP) et de se focaliser sur les outils et méthodes de développements. Aujourd'hui, l'univers

des systèmes parallèles est hétérogène et utilise de nombreux cœurs, qu'ils soient CPU ou GPU, etc. HSA veut créer et proposer des standards ouverts et des interfaces communes de programmation. HSA dispose déjà de différents composants pour le développeur : HSA Runtime, définition de la plateforme HSA. Intel n'est pas membre, ce qui relativise de facto la portée réelle de HSA.

Site : <http://hsafoundation.com>





Des stations de travail puissantes au service d'idées créatives.

Passez rapidement du concept à la réalité avec les nouvelles stations de travail Dell Precision dotées du processeur Intel® Core™ i5 de 3e génération.

Primés, ces produits Dell les plus performants sont spécialement adaptés pour les applications les plus exigeantes. Ils sont testés et certifiés pour une fiabilité optimale et couverts par des solutions d'assistance complètes Dell pour un temps d'inactivité minimal.

Découvrez les possibilités qui s'offrent à vous sur **Dell.fr/precision** ou appelez le **0 825 801 693**.

Numéro Indigo : 0.15 €^{TTC}/min.



AUTODESK
AUTOCAD
2014

Les stations de travail
Dell Precision sont
certifiées par AutoCAD.



Le pouvoir d'en faire plus

Offres réservées aux petites et moyennes entreprises de moins de 200 salariés, à la France métropolitaine, dans la limite des composants et pièces disponibles. En fonction de la taille de la mémoire système et d'autres facteurs, une quantité de mémoire système importante peut être nécessaire pour prendre en charge les cartes graphiques. Un système d'exploitation 64 bits est requis pour prendre en charge au moins 4 Go de mémoire système. La capacité réelle du disque dur varie en fonction du matériel préinstallé et du système d'exploitation et peut être inférieure aux valeurs indiquées. Ultrabook, Celeron, Celeron Inside, Core Inside, Intel, Logo Intel, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, Logo Intel Inside, Intel vPro, Itanium, Itanium Inside, Pentium, Pentium Inside, vPro Inside, Xeon, Xeon Phi, et Xeon Inside sont des marques de commerce d'Intel Corporation aux Etats-Unis et dans d'autres pays. De 18h00 à 09h00, le service est dispensé en anglais. Microsoft® et Windows® sont des marques déposées ou des marques commerciales de Microsoft Corporation aux Etats-Unis et dans d'autres pays. Precision est une marque déposée de Dell Inc. Offres soumises aux Conditions Générales de Vente et de Service Dell, disponibles sur www.dell.fr. Dell S.A. Capital : 1 782 769 €, 1 Rond Point Benjamin Franklin - 34938 Montpellier Cedex 9 France. RCS Montpellier N° 351 528 229 - APE 4651 Z.

Retour sur **Devoxx France 2013** : les interviews

Dans le numéro précédent de Programmez !, nous avons fait un long compte-rendu de l'événement Devoxx. Ce mois-ci, dernière étape Devoxx avec trois conférenciers techniques !



© Devoxx

> Rencontre avec **Eric Bottard**

Peux-tu te présenter en quelques mots ?

Je suis Developer Advocate chez Pivotal, une société dont le but est de faciliter la création d'un nouveau type d'applications, partant du constat que l'on doit de plus en plus gérer de

grosses quantités de données (Big Data) et ce en quasi-temps-réel (Fast Data), le tout sur des infrastructures scalables et sans contraintes (indépendance vis-à-vis du Cloud). Pivotal est issue de la rencontre des meilleurs produits et équipes de VMware et d'EMC dans ces domaines. Mon job consiste donc à présenter à la communauté de développeurs les possibilités et évolutions de nos solutions, qu'elles soient open source (comme Spring et Cloud Foundry) ou non.

Peux-tu nous parler de la session que tu as présentée à Devoxx France ?

J'ai parlé des bonnes et mauvaises pratiques d'architecture logicielle, qui prennent une importance capitale lorsque l'on souhaite migrer (ou créer de toutes pièces) une application destinée à tourner sur un PaaS (Platform as a Service, tel que Cloud Foundry, c'est-à-dire une plateforme de Cloud sur laquelle on ne se soucie pas de l'installation des machines et de leur mise à jour, mais où l'on déploie une "application" en quelques commandes ou clics). J'ai notamment insisté sur le fait qu'il faut éviter de coupler son

code à une quelconque solution, afin de s'assurer de pouvoir, par exemple, passer d'un Cloud public à un Cloud privé, ou inversement.

Que retiens-tu de cette seconde édition de Devoxx France ?

Habitué de Devoxx Anvers, je n'avais pas pu participer à l'édition française de l'année dernière. Eh bien, je dois dire que j'ai été impressionné ! A la fois par l'organisation, qui a semblé sans faille, mais surtout par le contenu. Non seulement les speakers internationaux étaient très bons, mais les sessions en français (qui représentent la majorité du programme) étaient d'un excellent niveau. Ceci prouve que nous n'avons pas à rougir face aux environnements anglosaxons notamment, et que de nombreux projets à haute valeur ajoutée technique sont exécutés en France. Il ne tient qu'à chacun de venir s'y intéresser. D'ailleurs, ce conseil vaut aussi pour les managers de tout bord, car il est clair que le frein n'est pas du côté des développeurs. Rendez-vous donc l'année prochaine !

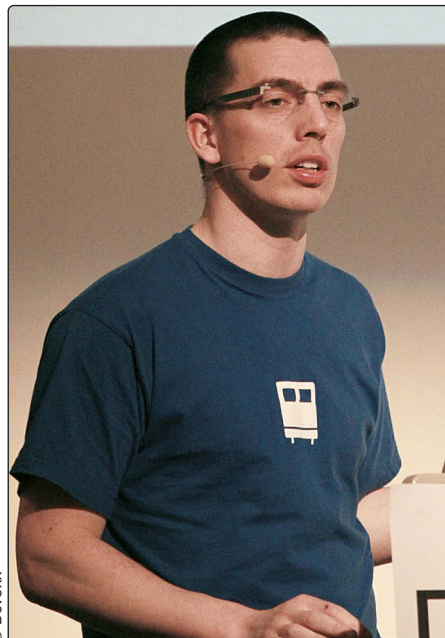
> Rencontre avec **Cyril Mottier**

Peux-tu te présenter en quelques mots ?

Je suis diplômé de l'INSA de Rennes. A l'issue de mes études - plutôt généralistes - je me suis orienté vers la mobilité, un domaine qui regroupe deux de mes passions : les contraintes et le design. Après un rapide passage en société de service, j'ai découvert le monde des startups et c'est pour mon plus grand plaisir que je travaille aujourd'hui chez Capitaine Train en tant qu'ingénieur développement mobile.

Peux-tu nous parler de la session que tu as présentée à Devoxx France ?

Cette année, à Devoxx, j'ai présenté une session intitulée "Mastering Android Drawables". L'objectif principal derrière cette conférence était de décrire une partie malheureusement méconnue du SDK Android. De mon point de

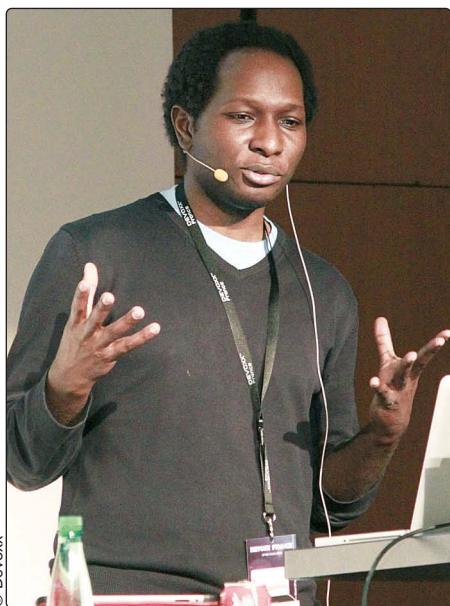


© Devoxx

vue, parfaitement comprendre le système de fonctionnement des Drawables sur Android est une condition sine qua non à la réalisation d'une application Android de qualité.

Que retiens-tu de cette seconde édition de Devoxx France ?

Tout ce que je peux dire sur cette édition 2013 c'est qu'elle m'a permis de rapidement faire un tour d'horizon des technologies qui buzzent et même de rentrer dans les détails dans certains cas. Participer à Devoxx France est vraiment la meilleure façon d'avoir un état de l'art du paysage technologique actuel et ceci dans une ambiance conviviale et amicale. Une conférence à ne pas rater. Enfin, je dirais que la conférence que j'ai préférée était "What Every Hipster Should Know About Functional Programming" de Bodil Stokke.



© Devvxx

> Rencontre avec Sam Bessalah

Peux-tu te présenter en quelques mots ?

Je suis développeur, diplômé d'un Master en informatique de Paris V, et passionné par les problématiques sur les datas, englobant les architectures d'analyse des données à plus ou moins large échelle et les algorithmes qui permettent cette analyse (Data

Mining, Machine Learning...). En ce moment je travaille pour Next Capital, une entreprise qui fait de l'investissement dans les marchés financiers.

Peux-tu nous parler de la session que tu as présentée à Devvxx France ?

Ma conférence était intitulée "Structure de données exotiques". J'ai voulu aborder des structures de données un peu moins connues du commun des développeurs, souvent bien cachées dans des bibliothèques utilisées au quotidien, mais qui ont des propriétés intéressantes en termes de performance et d'espace occupé. C'est ainsi donc que j'ai parlé des skip lists, variante des Linked List, puis des hash array mapped tries, très utilisées dans les langages fonctionnels comme Clojure ou Scala. Et pour terminer, j'ai abordé les structures de données probabilistes, notamment les bloom filters ou le count min sketch, qui sont deux structures de données très utilisées dans les systèmes Big Data, que ce soit avec les bases de données NoSql comme Cassandra ou HBase, ou pour faire de l'analytique en temps réel sur des flux continus de données.

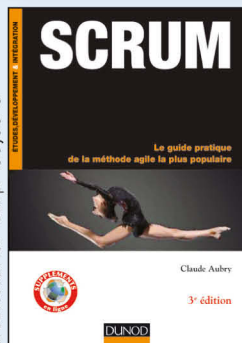
Que retiens-tu de cette seconde édition de Devvxx France

Je suis un grand fan des conférences de développeurs par définition. Et Devvxx est une messe pour les passionnés de notre métier. Je pense que la touche unique de Devvxx France est sa diversité. En effet, bien que cette conférence émane du Paris JUG, cette dernière couvre un large éventail de technologies qui ne se limitent pas à Java et son écosystème. Cette année on a vu du html5, Javascript, iOS, Cloud, Big Data, Machine Learning et même du Haskell. J'ai beaucoup apprécié la présence des speakers de renommée internationale, spécialistes dans leurs domaines, comme Martin Odersky, Martin Thompson, Ted Dunning, Bodil Stokke, Matt Raible et beaucoup d'autres. Discuter avec eux a été très instructif en ce qui me concerne.

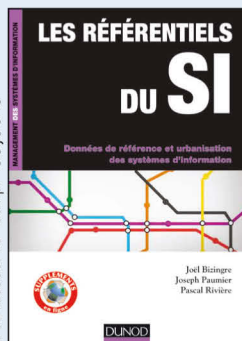


Propos recueillis par Ludwine Probst, consultante Soat (@nivdul)

DÉVELOPPEZ VOS COMPÉTENCES



Le guide pratique de la méthode agile la plus populaire

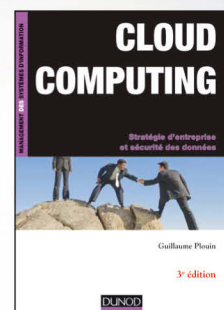


Une vue d'ensemble des référentiels : sujet de base dans la conception d'un SI



Réconcilier les chefs de projet avec les outils de management de projet

Tout savoir sur le Cloud Computing



Google I/O : IDE Google, nouvelles cartes !

La conférence développeur Google, Google I/O s'est tenue mi-mai. Cette année, Google a surtout voulu s'adresser aux développeurs et utilisateurs. Les nouveautés matérielles ont été presque absentes, tout comme les futures versions d'Android. Ce qui n'est pas forcément une mauvaise chose. L'équipe Android a tout de même précisé qu'elle travaillait toujours à améliorer la fluidité du système

> Google Maps : une longueur d'avance

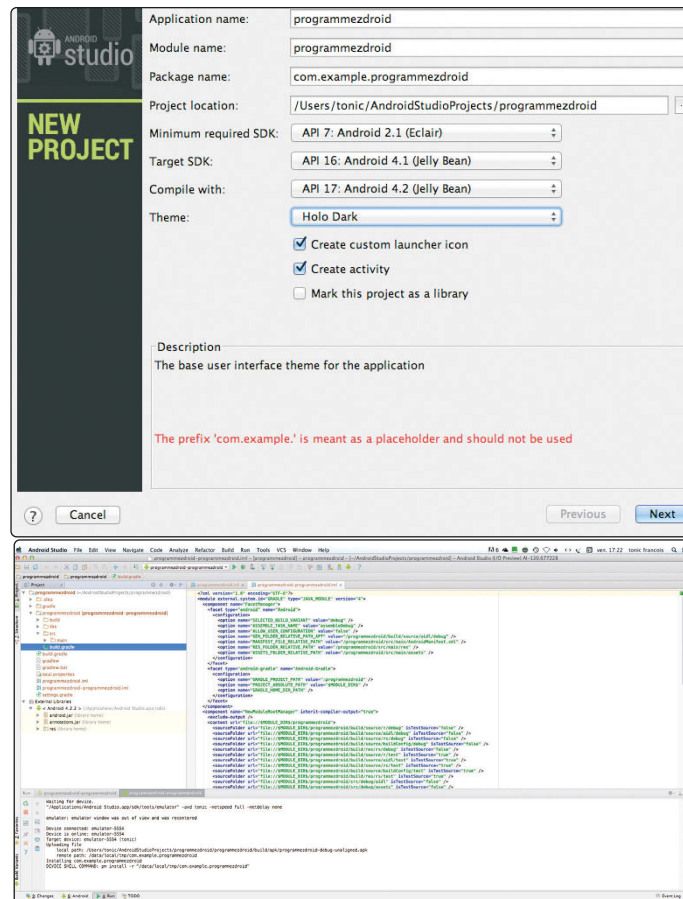
Google a dévoilé la nouvelle version de Maps, avec les SDK qui vont bien. Google rappelle que les données sont vitales pour un tel projet. Aujourd'hui 200 pays sont couverts soit par Google, soit par des partenaires, Street View est disponible dans 50 pays. Aujourd'hui plus d'un million de sites utilisent les API Maps. La partie mobile est importante aussi bien sur Android que iOS. L'avenir de Maps est sans doute dans le mobile et Google le pousse à toutes les sauces. Le prochain SDK promet beaucoup : nouveau design (notamment avec la granularité des cartes 2D/3D) et plus de simplicité, données temps réel. Sur le desktop, Maps s'appuie désormais sur WebGL (assez impressionnant) pour les vues satellites et on peut visiter les lieux intérieurs (s'ils sont disponibles).

> Web Components : le futur du web

Durant la première journée, Google a montré les Web Components même si le projet n'est pas nouveau en soi. L'objectif est de simplifier la réutilisation de composants, l'utilisation de composants, et la programmation liée. Il faut pouvoir aider le développeur à aller vite à construire des composants facilement transposables ailleurs et éviter d'écrire des milliers de ligne de code. Les Web Components sont écrits en HTML et CSS. Sur-tout, ces composants peuvent être utilisés sur n'importe quel navigateur.

> Android Studio : Google veut son IDE !

Jusqu'à présent, pour développer des applications Android, on utilisait le plus souvent Eclipse, avec les avantages et les inconvé-



nients que cela présente. Durant la conférence, Google a dévoilé son propre IDE, basé sur la version communautaire de IntelliJ, un IDE Java reconnu pour ses qualités.

Officiellement, Google ne remplace pas Eclipse par Android Studio. Il sera toujours

possible de développer avec. Mais, Google cherche à proposer un environnement de développement intégré à l'instar de Microsoft Visual Studio pour Windows / Windows Phone ou d'Apple XCode pour iOS. Android Studio se veut très proche dans la philosophie d'Eclipse + ADT Plugin :

- refactoring spécifique à Android
- template de créations rapides
- éditeur « riche » pour le design
- prévisualisation des designs sur une multitude de configurations d'écrans

L'assistant de création de nouveaux projets est pratique. On choisit les bases de son projet et il génère le squelette de l'application. Pour faciliter la migration d'Eclipse vers Android Studio, Google a mis en ligne un petit guide pratique. Cependant, attention, Android Studio est encore en développement et disponible en version 0.1, bref en alpha ! Google prévient que

des fonctions sont incomplètes ou absentes. L'outil est disponible sur Windows, Linux et OS X.

Pour en savoir plus :

<http://developer.android.com/sdk/installing/studio.html>

Du quantique chez Google





Google travaille sur de nombreux domaines. Le quantique et l'informatique quantique rentrent dans cette recherche fondamentale qui débouchera sur quelque chose dans 5 à 10 ans. L'éditeur a acheté un ordinateur quantique D-Wave dans le cadre d'une collaboration avec la NASA. Ce travail commun est très concret car les deux sociétés ont créé le Quantum Artificial Intelligence Lab. Le quantique peut aider à résoudre des problématiques actuelles et futures dans le monde informatique et notamment sur la notion de machine à apprendre. Très complexe à appréhender, Google et la NASA pensent que le quantique peut les aider dans cette recherche. Annonce : <http://googleresearch.blogspot.fr/2013/05/launching-quantum-artificial.html>

WINDOWS OU LINUX CHOISISSEZ !

Chez 1&1, vous êtes libre de choisir entre un hébergement Linux ou Windows.

Avec la **dernière version d'ASP.NET**, vous bénéficiez d'une palette d'outils innovants, pour des sites Web de qualité professionnelle dans un environnement Windows.

Pour les adeptes de Linux, les packs 1&1 offrent un environnement de développement idéal avec le **nouveau PHP 5.4** et un grand choix d'applications Click & Build : WordPress, Joomla!, Drupal et bien plus encore !

 1&1 Illimité Windows	 1&1 Illimité Linux
Trafic illimité et bande passante de 100 Mbps	
Espace Web illimité	
500 comptes email (2 Go) et 500 accès FTP	
3 noms de domaine inclus pendant toute la durée de votre pack (au choix : .fr, .com, .info, .net, .org, .eu)	
1&1 Sécurité : géo-redondance, 1&1 SiteLock (en option)	
Service expert 6j/7 via hotline non surtaxée et email	
et bien plus encore...	
NOUVEAU : ASP.NET / .NET Framework 4, 4.5  (compatibles avec les versions précédentes)	NOUVEAU : PHP 5.4 , Zend Framework, Perl, Python
NOUVEAU : ASP.NET MVC 3 et 4	Protection de vos données : Restauration de l'espace Web
NOUVEAU : 10 bases de données MS SQL (1 Go) 	100 bases de données MySQL 5 (1 Go)
NOUVEAU : Applications en pools isolés	Accès à 65 applications Click & Build au choix : WordPress, Joomla!, Drupal...
6 MOIS GRATUITS puis 6,99 € HT/mois (8,36 € TTC/mois)*	6 MOIS GRATUITS puis 6,99 € HT/mois (8,36 € TTC/mois)*

Découvrez toutes nos autres offres sur notre site Web.



SÉCURITÉ MAXIMALE

Vos données sont hébergées en parallèle dans 2 centres de données 1&1 haute performance situés à différents endroits. Le « plus » : sauvegardes automatiques quotidiennes.

SERVICE EXPERT 6j/7

En tant que client 1&1, vous bénéficiez d'une assistance non surtaxée (hotline et email), assurée par des professionnels de l'hébergement.

1&1

1and1.fr



DOMAINES | EMAIL | HÉBERGEMENT | E-COMMERCE | SERVEURS

 **0970 808 911** (appel non surtaxé)

En juin

DevParty@Nantes 2013

Cette manifestation régionale de référence est organisée à l'Epitech de Nantes par le JUG Nantes, le GDG Nantes et Kskills, et se déroulera le 4 juin 2013. Elle réunira plusieurs dizaines de développeurs qui pourront concourir à un challenge unique sur la région et mettre en avant leur expertise. Concrètement, après avoir répondu à différents quizz, les développeurs prendront part à des challenges de programmation en utilisant le langage de leur choix : Java, C, Python, JavaScript, C# et PHP. Bien entendu, cette soirée se réalisera dans un cadre convivial et les participants seront récompensés en recevant de nombreux lots.

Site : <https://guestlistapp.com/events/159942>

EclipseCon débarque en France

Les 5 et 6 juin prochains se tiendra à Toulouse, la première conférence EclipseCon France ! Cette conférence parlera bien entendu d'Eclipse et de son riche écosystème. Pour les conférences (l'agenda sera disponible courant mai), elles tourneront autour de l'industrie, la communauté et les modèles économiques, le développement web et mobile. Il sera possible de voir des projets nouveaux et innovants basés sur Eclipse.

Site officiel : <http://www.eclipsecon.org/france2013/>

Ch'ti JUG, le 10 juin

Le JUG du Nord propose une belle soirée autour d'Angular JS en mode live coding. Voilà une soirée hardcoding comme on les

Concours

Moov' in the city



Le concours Moov' in The City est lancé depuis quelques jours. Le but de ce concours de développement est simple : « En utilisant les données de la Ville de Paris, de la RATP, de JCDecaux ou de la SNCF, participez à l'amélioration de la vie des usagers des transports parisiens. Vous êtes développeur, designer ou

entrepreneur, alors participez au concours Moov'In The City et créez un site internet, une application mobile ou de la data visualization et tentez de gagner plus de 25 000 € de dotation ». La finale se déroulera début juillet !

Site : <http://moovinthecity.fr>

aime à Programmez !. « BackboneJs, EmberJs, KnockoutJS, CanJS ... les développeurs Front-End ne manquent pas de frameworks JavaScript pour les aider à développer des applications Web modernes. Juin 2012, AngularJS – estampillé 1.0 – sort des usines Google et vient rejoindre cette liste déjà assez longue des frameworks JavaScript. "Mais pourquoi dois-je m'intéresser à AngularJS ? Qu'apporte-t-il de plus par rapport aux autres frameworks ?" Autant de questions que vous vous posez sans doute », dit le JUG. Session en français.

Lien : <http://chtijug.org>

(DSI, RSSI, RSI) et les experts techniques du hacking, se déroulera du 17 au 21 juin prochain au Centre de Congrès de Disneyland Paris.

Site : <http://www.hackinparis.com>

DrupalCamp Paris 2013



Les 21, 22 et 23 juin, le plus grand événement Drupal français aura lieu à Issy les Moulineaux. De nombreuses sessions techniques mais aussi des retours d'expériences. Vous pourrez toucher Drupal 8, la livraison continue, les blocs sous Drupal, Twig, les tests.

Site : <http://paris2013.drupalcamp.fr>

Et aussi :

- Coding Breakfast #3 : 5 juin à Paris. Autour d'un petit déj', discuter technique, code et des problèmes rencontrés. <http://www.meetup.com/altnetfr/events/116746562/>

- 1er meeting Paris.py : 12 juin à Paris. Soirée orientée Python / Django ! <http://www.meetup.com/Paris-py-Python-Django-friends/events/119858032/>

agenda

➤ JUIN

- Le 7 juin, Théâtre des Variétés, 7 Boulevard Montmartre 75002, **CloudConf 2013**, Un événement pour les développeurs et les DevOps uniquement, suivi d'une journée d'atelier le 8 juin. <http://cloudconf.eu>
- Du 12 juin au 13 juin 2013, Porte de Versailles, **i-expo 2013**. Les grandes problématiques autour du Web 2.0 seront abordées, sous la forme de conférences plénières, ateliers, tables rondes ou encore conférences d'application exposants. <http://www.i-expo.net>
- Le 25 juin, MAS-Paris 75013, **CMSDay 2013**, le premier événement majeur, en France et en Europe, dédié à la gestion de contenu open source. <http://www.cmsday.fr/Accueil>

Hack in Paris 2013 !

Organisé pour la troisième fois, l'événement Hack in Paris, unique événement en France exclusivement en anglais et réunissant les professionnels de la sécurité informatique





FILE APIs

Api pour manipuler les fichiers

CONVERTIR
IMPRIMER
CRÉER
COMBINER
MODIFIER



Files
from your
CODE

les Fichiers depuis votre CODE

Aspose.Words

DOC, DOCX, RTF, HTML, PDF, XPS et d'autres formats de document.

Aspose.Cells

XLS, XLSX, XLSM, XLTX, CSV, SpreadsheetML et formats d'images.

Aspose.BarCode

JPG, PNG, BMP, GIF, TIF, WMF, ICON et d'autres formats d'images.

Aspose.Pdf

PDF, XML, XLS-FO, HTML, BMP, JPG, PNG et d'autres formats d'images.

Aspose.Email

MSG, EML, PST, EMLX et d'autres formats.

Aspose.Slides

PPT, PPTX, POT, POTX, XPS, HTML, PNG, PDF et d'autres formats.

... et bien d'autres fonctions!



Scannez pour
remise de -20%



ASPOSE

Your File Format Experts

Suivez nous sur
Facebook et Twitter

Obtenez votre version d'évaluation gratuite sur www.aspose.com

US Sales: +1 888 277 6734
sales@aspose.com

EU Sales: +44 141 416 1112
sales.europe@aspose.com

AU Sales: +61 2 8003 5926
sales.asiapacific@aspose.com

> Nokia lance les nouveaux Lumia, Asha, et dévoile une mise à jour de WP8

Le constructeur Nokia a dévoilé le 14 mai dernier le nouveau Lumia, terminal dédié à Windows Phone 8 : Lumia 925. Ce modèle doit répondre aux critiques formulées sur le 920 et notamment sur le poids, l'autonomie, le design.

Outre quelques dizaines de grammes en moins, le 925 hérite d'une optique caméra améliorée qui doit augmenter la qualité et la précision de la vidéo et de la photo. C'est d'ailleurs l'argument mis en avant par le constructeur. Tous les Lumia pourront aussi accéder au tuner FM (excepté le 620).

Dans la foulée de l'annonce, Nokia dévoile une importante mise à jour de Windows Phone 8, nom de code « Amber » (prévue pour cet été). Amber doit apporter à WP8 les fonctions manquantes de WP7 telles que le tuner FM. Cette version tirera profit de la nouvelle caméra du 925 (support jusqu'à 3200 ISO, meilleure gestion du bruit, mise à jour de l'application caméra). Seul le 620 ne pourra pas en profiter ! Amber doit aussi apporter beaucoup de petites améliorations et corrections. Pour le développeur, peu de changement à prévoir sauf sur la partie caméra (les exemples sur le site développeur Nokia indiquent pour le moment les modèles 820 et 920).

Pour en savoir plus : <http://goo.gl/jNQiI>

Et voir aussi la partie développeur :

<http://www.developer.nokia.com/Develop/asha/>

Nokia n'abandonne pas les « features phone », bien au contraire, il fait toujours un important volume sur ce marché. Sur le smartphone (chiffres Gartner), Nokia arrive à 4,9 % soit 5,1 millions d'unités contre 64 millions à Samsung et 38 millions à Apple ! C'est pour cela

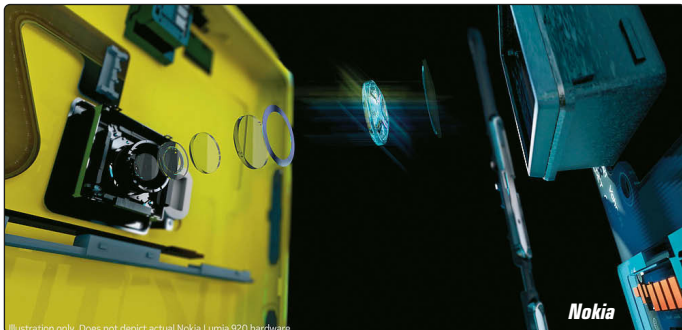
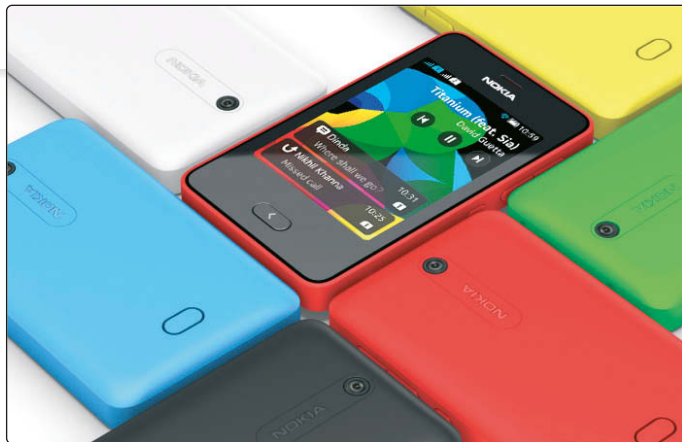


Illustration only. Does not depict actual Nokia Lumia 920 hardware.

que Nokia a dévoilé la gamme Asha avec un environnement de développement dédié. Asha reprend quelques principes des Lumia mais fonctionne avec un système spécifique et Java ME. Il possède un écran 3", des capteurs et capacités 2D/3D. Nokia utilise les dernières versions de Java ME d'Oracle. L'interface a été développée pour le Asha. Pour les développeurs, Nokia propose un Asha SDK (incluant du code, la documentation, un IDE). Les API utilisent celles des terminaux série 40 avec quelques rajouts : notification, contacts, état du réseau. L'émulateur a été corrigé pour proposer de nouvelles fonctions telles que le diagnostic réseau. Bon à savoir, Nokia IDE fonctionne aussi bien en Eclipse qu'en NetBeans.

Site développeur Asha : <http://www.developer.nokia.com/Develop/asha/>



> NVidia : la console portable Shield bientôt disponible

NVidia travaille depuis longtemps au projet Shield, une console portable, fonctionnant sous Android. Doté d'un design discret mais agressif dans les lignes, elle intègre un processeur Tegra 4, une manette intégrée (pad), un écran tactile (5 pouces), connexion wifi, GPS intégré. Shield devrait arriver courant juin à un prix annoncé de 370 \$. La plupart des applications Android devraient fonctionner et des dizaines de jeux seront spécifiquement optimisés. Un mode streaming PC sera proposé dans les prochains mois. Le joueur pourra alors jouer sur son PC depuis la console (à condition de posséder une GeForce GTX compatible). Il s'agit d'un pari assez audacieux car les smartphones et tablettes ont investi ce marché. Site : <http://shield.nvidia.com>

> BlackBerry : nouveaux terminaux et BB 10.1

Mi-mai, BlackBerry dévoilait ses nouveautés printanières et particulièrement la version 10.1 du système qui est disponible dès maintenant pour le Z10 et qui arrivera selon les opérateurs dans les prochaines semaines. La 10.1 corrige et améliore plusieurs modules : Hub, le centre de notification, la caméra (la prise de vue en mode HDR apparaît), amélioration du calendrier, etc. Pour le développeur, l'impact sera limité, voire nul. La grosse nouveauté est l'annonce du Q5, un modèle d'entrée de gamme pour les pays émergents, un marché en pleine explosion. Il reprend un clavier physique. Les autres fonctions : BBM vidéo, Hub, AppStore. Il embarque le système 10.1. L'écran de 3"1 est tactile.



BlackBerry

DANS LA JUNGLE DU CLOUD, MIEUX VAUT CHOISIR LE BON PARTENAIRE.



Aruba Cloud, les solutions IaaS qui répondent à chacun de vos besoins.

CLOUD COMPUTING

- Créez, activez et gérez vos VM.
- Choisissez parmi nos 3 hyperviseurs.
- Maîtrisez et planifiez vos ressources CPU, RAM et espace disque.
- Uptime 99,95% garanti par SLA.

CLOUD OBJECT STORAGE

- Créez vos espaces et stockez vos données en toute sécurité.
- Une solution qui s'adapte à vos besoins : Pay as you Go, ou formule prête à l'emploi.
- Bande passante et requêtes illimitées.

LE CLOUD PAR ARUBA

- Ubiquité : choisissez votre pays et datacenter.
- Interopérabilité : API et connecteurs.
- Agnosticisme : choisissez votre hyperviseur.
- Scalabilité : étendez votre infrastructure à l'infini.
- Transparence : pas de coûts d'activation, ni coût caché.
- Pay as you Go : ne payez que ce que vous consommez.

Aruba, le bon partenaire pour bénéficier de la puissance d'un acteur majeur qui considère que chaque client, dans chaque pays, est unique. **MY COUNTRY. MY CLOUD.**



arubacloud.fr | TÉL : 0810 710 300
(COÛT D'UN APPEL LOCAL)

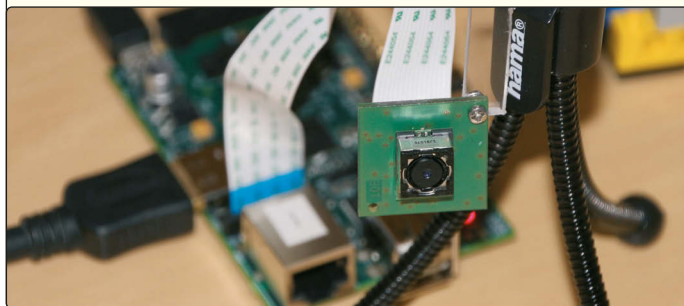


> Le dock Thunderbolt Express de Belkin enfin en vente !

Ouf ! le dock Thunderbolt Express de Belkin arrive. Il possède deux ports Thunderbolt. Il permet de connecter des périphériques USB 3, Firewire, Ethernet et audio. Les flux annoncés par le constructeur sont : x20 plus rapide qu'en USB 2, avec des transferts pouvant atteindre 10 Gbps. La station d'accueil Thunderbolt Express utilise une seule connexion à haut débit pour le transfert de données ultra rapide entre l'ordinateur portable et jusqu'à huit périphériques, et permet aux utilisateurs de connecter en chaîne jusqu'à 5 appareils Thunderbolt supplémentaires. La station d'accueil représente une solution astucieuse pour obtenir un espace de travail plus propre, plus rapide et plus productif, ainsi qu'une connectivité fiable aux périphériques et à Internet. Dommage que son prix de vente soit si élevé : 349 € !

> La caméra Raspberry Pi disponible !

Attendue depuis des mois, la caméra dédiée au Raspberry Pi est désormais disponible chez RS Components. Pour 20,36 €, le module caméra comprend un capteur d'image CMOS de 5 mégapixels OmniVision, construit sur l'architecture propriétaire OmniBSI d'illumination arrière des pixels de 1,4 microns. Il est capable de prendre des photos de 2592 x 1944 pixels et de capturer des vidéos HD jusqu'à 60 images par seconde en 720p. Le module caméra se connecte par l'intermédiaire de l'embase CSI du Raspberry Pi. Il permet également aux utilisateurs d'enregistrer en 1080p à 30 images par seconde au format vidéo H.264.



> SSHD : nouvelles gammes WD et Sandisk

Les deux constructeurs annoncent des disques hybrides 2,5 pouces. SanDisk fournit les mémoires (iSSD) aux disques WD Black.



« Avec l'aide de SanDisk, nous avons pu concrétiser la vision de WD de produits combinant le meilleur de nos deux technologies », confirme Matt Rutledge, vice-président de la division Informatique client de WD. « Les disques durs SSHD de WD sont

une parfaite illustration de la révolution en marche dans le stockage SSHD/flash. »

Le but : des disques hybrides plus performants. Pour le moment, la capacité sera de 500 Go.

> Archos continue sur Android

Le constructeur français a dévoilé le 16 mai dernier sa nouvelle tablette Android : l'Archos 80 xenon. Elle fonctionne bien entendu avec Google Play. Elle intègre Google Maps, le GPS, un écran 8 pouces, Android 4.1, les applications mobiles Google. "Nous avons conçu l'ARCHOS 80 xenon, pour répondre à la demande de plus en plus importante des utilisateurs qui recherchent un appareil

alliant grand écran et connectivité." indique Loïc Poirier, Directeur Général d'Archos. "Grâce à de nouvelles offres de souscription 3G plus abordables, l'ARCHOS 80 xenon s'emporte partout pour se connecter à tout moment, à un prix raisonnable." Archos tente de profiter d'un marché qui explose, mais que seuls quelques constructeurs occupent.

> Nouvelles stations Dell Precision

Dell propose de nouvelles stations de travail dans la gamme Precision : le T1700 et le R7610. Ce dernier est une station au format rack. En plus de ces deux stations de travail, la gamme compte également une

nouvelle station d'entrée de gamme le T1700 Mini-Tower. Ces modèles sont disponibles dès maintenant (ou le seront dès le 4 juin). Le modèle R7610 est vendu 2242 €. Les autres tarifs seront connus le 4 juin.

> Google Glass : nouvelles fonctions dans le SDK

Durant la conférence Google I/O, Google a annoncé plusieurs nouveautés importantes dans le kit de développement (Mirror API) : applications utilisables en mode déconnecté et accès direct aux couches matérielles. Des API natives devraient être aussi disponibles dans le futur. D'autre part, Google veut développer tout un écosystème autour des Google Glass.

> Zappiti Player Mini : un lecteur multimédia français

Le Zappiti Player Mini est un lecteur multimédia de salon Full HD 3D. Il permet d'y connecter un disque dur, possède plusieurs ports USB, HDMI, Ethernet, Wifi, etc. Il supporte une grande variété de formats et codecs vidéo et audio. La partie logicielle a été

développée en France, le matériel est d'origine mixte française et taiwanaise. Des applications mobiles (iOS / Android) permettent de profiter du lecteur et des contenus sur son iPhone, iPod Touch et iPad.

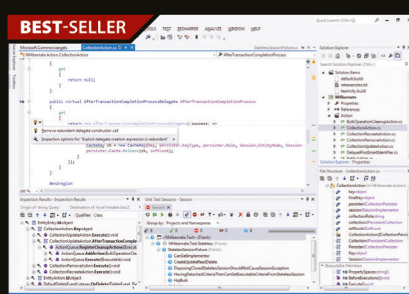
Site : <http://zappiti.com>




Aspose.Words for .NET à partir de € 764


Lisez, modifiez et écrivez des documents Word sans Microsoft Word.

- Création de documents, manipulation du contenu/formatage, puissante capacité de fusion de courrier et exportation en DOC/HTML
- Accès détaillé à tous les éléments d'un document par programmation
- Support les formats de fichiers: DOC, DOCX, WordprocessingML, RTF, HTML, OOXML, OpenDocument, PDF, XPS, EMF et EPUB


ReSharper à partir de € 308

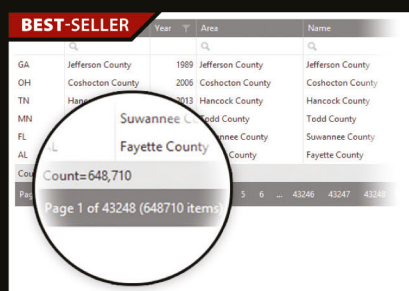

Outil de développement Microsoft Visual Studio 2005-2012 pour améliorer la productivité.

- Analyse de qualité du code à la volée dans C#, VB.NET, XAML, ASP.NET, MVC, JS, CSS, HTML, XML
- Accès ultrarapide à tout fichier, type ou membre dans votre base de code
- Signale les problèmes de code et propose des solutions rapides de correction
- Assistants de codage incluant IntelliSense et transformations instantanées du code
- Refactorisations de la solution pour changer la base de code en toute sécurité


TX Text Control .NET for Windows Forms/WPF à partir de € 911


Composants complets de traitement de texte pour Windows Forms et WPF.

- Le traitement de texte professionnel pour vos applications
- Chargez, modifiez et enregistrez au format MS Word (DOCX, DOC, RTF)
- WYSIWYG, tableaux imbriqués, cadres, en-têtes, pieds de pages, images, puces, listes numérotées, zoom, sauts de section, etc.
- Exploitez la liaison de données XAML pour lier les barres d'outils et de ruban


DevExpress Universal Suite à partir de € 1 683


Outils et contrôles WinForms, ASP.NET, WPF, Silverlight et Windows 8.

- Construire des applications pour Windows, web, mobile et tablette
- Exploitez vos compétences pour développer des applications tactiles sur toutes les plateformes
- Créez des applications métier avec l'esthétique Windows 8
- Inclut des fonctions d'aide décisionnelle, de rapport et d'analyse
- Comprend CodeRush et eXpress Application Framework (XAF) tout en un ensemble intégré

Nathanaël Marchand : aucun jour ne ressemble à un autre !

Ingénieur en informatique, automatique et instrumentation (Polytech'Savoie), Nathanaël Marchand, 26 ans, est aujourd'hui consultant à So@t Expert .Net. Il évoque sa passion pour l'informatique mais aussi pourquoi et comment il est devenu développeur.

Comment es-tu tombé dans l'informatique et plus spécialement dans le développement ?

Pour moi, le développement est, paradoxalement, un hasard de mon parcours mais aussi une voie toute tracée. Un hasard, car au départ, je souhaitais faire une carrière de pilote de chasse. Etant myope, j'ai dû changer d'avis et je suis parti sur des classes prépas orientées mécanique. Pendant mon école d'ingénieur, qui était généraliste, je me suis dirigé vers la spécialité électronique et informatique embarquée. C'est grâce à un coup de cœur pour une start-up lors du choix de mon stage de fin d'études que je me suis lancé dans le .Net. Mais c'était également une voie toute tracée car depuis assez jeune, ce domaine m'intéressait : en primaire j'ai eu mon premier ordinateur, au collège je développais du VBA sous Excel et sous Access puis du VB6 car mon père me l'avait ramené du bureau. Ensuite, j'ai continué avec PHP, C et C++, en parallèle de mes études. Contrairement à beaucoup, je n'ai que très peu été intéressé par le développement de jeux vidéo, c'était plutôt les applications de gestion qui me fascinaient.

Je trouve qu'il y a également une grande ressemblance entre le développement informatique et les LEGO, le but est toujours d'assembler les différentes briques pour bâtir un édifice toujours plus complexe. J'ai passé des heures à assembler ces morceaux de plastique et aujourd'hui, cela me fait le même effet lorsque les différentes briques de mes architectures s'emboîtent.

Enfin, je suis quelqu'un de très curieux : j'ai toujours besoin de comprendre ce qui se cache dans une boîte noire, comprendre les mécanismes. Ce qui m'a toujours amené à essayer de démonter, décortiquer les mécanismes et autres objets complexes. Je retrouve la même chose dans mon métier.

Pour toi, qu'est-ce qui fait que l'on aime toujours et encore le développement, la technique ?

Le développement, c'est une remise en question perpétuelle. C'est vraiment ce qui me plaît là-dedans. Il faut toujours chercher à améliorer son code, optimiser, concevoir à nouveau afin de l'adapter à de nouvelles contraintes, le faire évoluer vers les nouvelles versions. La recherche d'une solution à un problème, la modélisation de l'environnement, trouver le meilleur enchaînement logique entre les choses, ce sont toutes des problématiques qui permettent de se creuser le cerveau et cette sensation est extrêmement enivrante.

Tu as gardé un regard très geek : gadget, veille techno. C'est important pour ton job et ta passion ?

Essentiel ! Je pense que, pour réussir dans ce secteur, il est très important de toujours être à la page, de toujours anticiper ce qui se profile à l'horizon. D'ailleurs, plus on avance, plus on se retrouve avec des scénarios très intéressants qui se démocratisent pour améliorer la vie quotidienne ! Les tablettes, smartphones sont aujourd'hui extrêmement présents, ce qui représente un marché et des possibilités incroyables pour nous, développeurs. Et puis, c'est assez rigolo de rentrer et d'avoir son Nabaztag qui souhaite la bienvenue, comme un fidèle animal de compagnie, en moins encombrant.

Etre développeur n'est pas toujours facile : pression, évolution constante, frustration des projets et des "chefs", c'est quoi pour toi être développeur aujourd'hui ? Le job a-t-il changé depuis tes débuts ?

J'ai la chance d'exercer un poste assez spécial puisque la cellule d'expertise de So@t



intervient sur des projets particuliers : vitrines technologiques (Kinect, Windows 8, etc.) ou missions pointues. Par conséquent, je n'ai jamais le temps de m'ennuyer, on est toujours confronté à de nouveaux challenges. Cependant, j'ai pu constater que selon là où on intervient, le rapport à l'informatique n'est pas le même, j'ai commencé dans une start-up de e-commerce : petits effectifs, ambiance sweat à capuche et baskets, avec l'IT comme pierre angulaire de l'entreprise, puis j'ai effectué des missions en banque d'investissement où là l'informatique est plutôt un centre de coûts pour le business et les process sont verrouillés. Il faut sans cesse s'adapter à l'environnement ! Entre deux missions, le job peut totalement changer.

Et en dehors du boulot, qu'est-ce que tu aimes faire ? Comment trouves-tu l'équilibre entre travail, vie privée, passion, famille ?

Il est vrai que mon travail m'occupe beaucoup et qu'il empiète largement sur mes soirées et week-end. Ça n'est pas dérangeant étant donné que j'ai la chance de faire un travail qui me passionne. Toujours est-il que parfois il m'est nécessaire de me couper de tout et de me changer les idées. Pour cela,

rien de tel que des sensations fortes : paintball, conduite automobile, surf...

J'adore aussi voyager et plus particulièrement pour découvrir les Etats-Unis. Je prévois de bientôt y retourner pour remonter toute la côte ouest dans un « muscle car » typique. Je viens également de terminer un important projet de ma vie qui m'a occupé pendant une année complète, la rénovation de mon appartement. J'ai dû jongler pendant mes week-end pour préparer mes différents projets et mener à bien les travaux.

Enfin, je participe souvent aux soirées organisées par So@t, elles me permettent de rencontrer d'autres personnes et notamment les autres consultants que je n'ai pas l'occasion de voir en mission. Les thèmes sont toujours assez rigolos, en ce moment c'est un tournoi par équipes en trois manches (bowling, fléchettes et karting).

Peux-tu nous présenter ton quotidien en quelques mots ? Comment vois-tu ton job évoluer ?

Les jours ne se ressemblent que très peu, il y a toujours du changement, tout dépend de mes activités. Soit je suis chez un client pour avancer sur des problématiques bien précises : audit, formation, point technique particulier. Je ne travaille pas intégré totalement aux équipes pour une longue durée, je viens en renfort ponctuellement aux différents moments importants de la vie d'une application. Soit je travaille au bureau pour répondre à des appels d'offres en faisant des maquettes, des formations internes, des projets internes, etc. Il m'arrive souvent d'accompagner les équipes commerciales pour des présentations. C'est toujours apprécié par le client d'avoir quelqu'un connaissant la technique afin de mieux



Hackathon Windows Phone de décembre 2012. © Bernard Lachaud

mesurer les problématiques plutôt que quelqu'un qui n'a pas eu de formation dans ce domaine. Il nous arrive aussi de faire des démonstrations de nos différents prototypes pour montrer notre savoir-faire.

Enfin, troisième point clef de mon travail : la capitalisation. La société offre à ses experts une quarantaine de jours par an (cette année) qu'ils peuvent aménager comme ils souhaitent. C'est grâce à ces journées que je peux préparer les conférences que je donne, écrire des articles pour la communauté, animer les différents forums et groupes d'utilisateurs, réaliser des projets personnels comme par exemple Showrizo, une application Windows 8 que j'ai réalisée avec Cyril Cathala, un de mes collègues expert. Il est bien entendu nécessaire que le choix de nos actions se fasse en concertation avec la direction technique. C'est un peu le même système qui a permis à Google de faire germer Gmail et Google Maps.

Ce dernier point est vraiment la partie que j'apprécie le plus, cela permet aussi de rencontrer des gens et d'échanger, ce qui est toujours enrichissant. J'aimerais beaucoup pouvoir y passer plus de temps mais il faut

trouver un juste milieu, tout d'abord vu l'investissement non négligeable pour la société mais également pour ne pas perdre de vue la réalité du terrain.

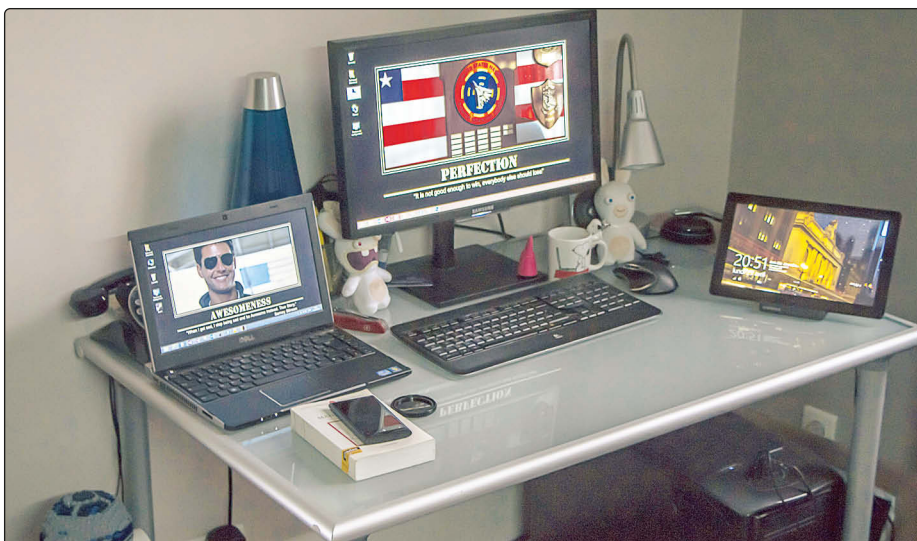
En tout cas, je compte bien m'éclater à ce poste pour un bon petit moment encore !

Quel est ton environnement de travail ?

Tout dépend où je suis ! Dans mon appartement, je possède un ordinateur fixe et une tablette Slate. So@t me fournit un ordinateur portable pour être mobile. Tout ce petit monde fonctionne sur Windows 8. Je possède également un Lumia 800 qui m'a été offert par Nokia pour ma conférence aux Tech'Days 2012, il a permis d'animer nos deux jeux multi-écrans « SoNuts » et « SoTank » sur notre stand pendant les trois jours. En clientèle, je dispose de la machine qui m'est fournie. Cependant, j'essaie en permanence d'installer ReSharper et LinqPad, deux petits utilitaires qui sont pour moi indispensables lorsque l'on travaille avec .Net.

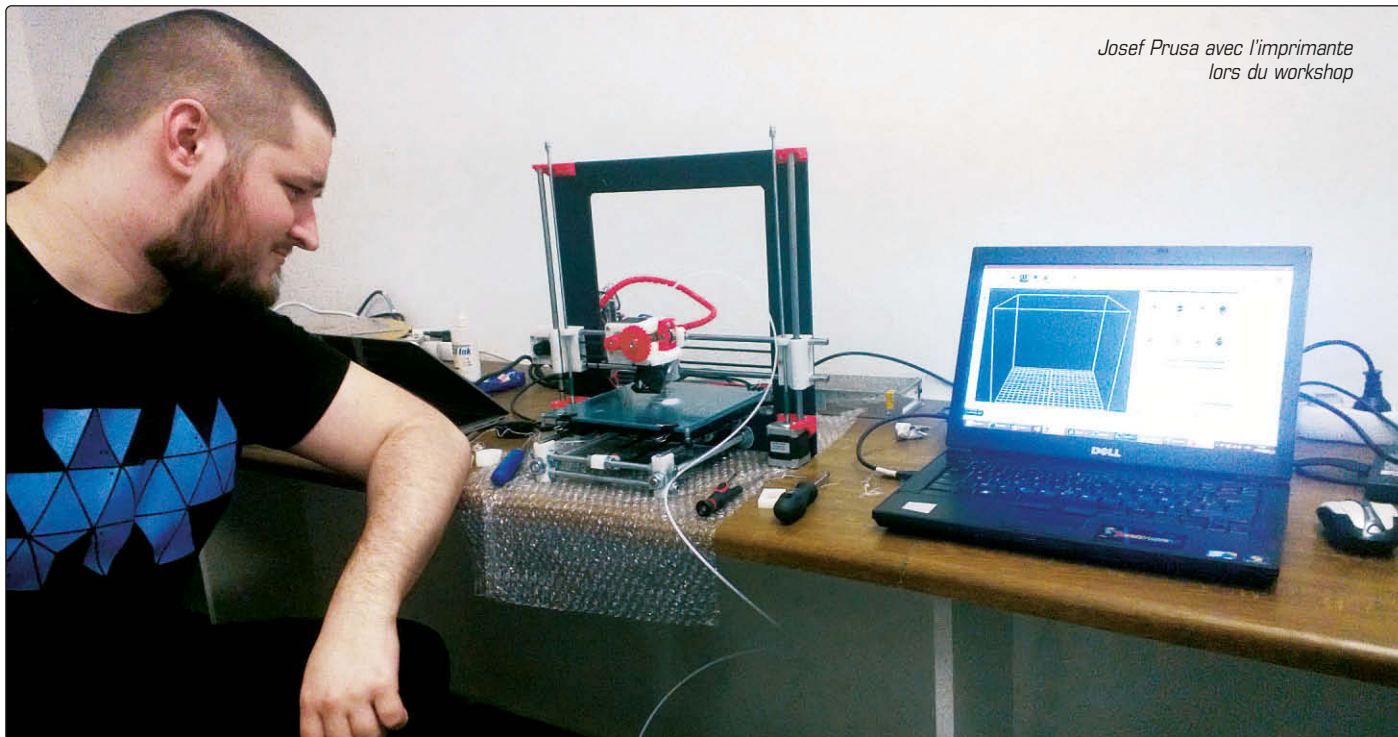
Des conseils aux étudiants et dévs qui nous lisent ?

Tout d'abord, je tiens à préciser que je suis fier d'être développeur. Pour moi, être développeur n'est pas une étape vers le management ou un poste de chef de projet comme on peut le voir trop souvent. C'est un poste à part entière, une vocation particulière qui nécessite certaines compétences, les postes de chef de projet en requérant d'autres. On peut tout à fait faire une carrière technique sur le long terme et réussir sa vie. Il faut aussi prendre conscience que tout le monde ne peut pas être développeur de jeux vidéo, que les postes les plus sympas nécessitent un investissement mais même dans des missions qui, en apparence, ne semblent pas géniales, il est toujours possible de trouver des côtés passionnants. #



Impressionné par l'impression 3D ?

L'impression 3D à le vent en poupe. Il faut dire qu'en ce moment c'est le buzz absolu. Ce n'est pas une nouveauté et cela existe depuis de nombreuses années, mais elle a surtout été réservée au monde industriel.



Josef Prusa avec l'imprimante lors du workshop

L'utilisation dans ce monde industriel est assez diverse : réalisation de noyaux complexes en fonderie, prototypage de pièces dans l'industrie automobile ou pour le design.

Une vidéo montre d'ailleurs comment ces impressions 3D ont été utilisées dans le design de la Surface de Microsoft (<http://aka.ms/y4kshy>).

Elles coûtent une fortune et ne sont pas accessibles aux geeks que nous sommes. Et c'est là qu'Adrian Bowyer a fait naître le projet RepRap, la première version d'une imprimante 3D simple à réaliser, peu coûteuse et utilisant des composants standards. Le tout disponible en open source, ce qui permet de construire son propre modèle. Depuis, de nombreux dérivés ont été créés et des entreprises se montent autour.

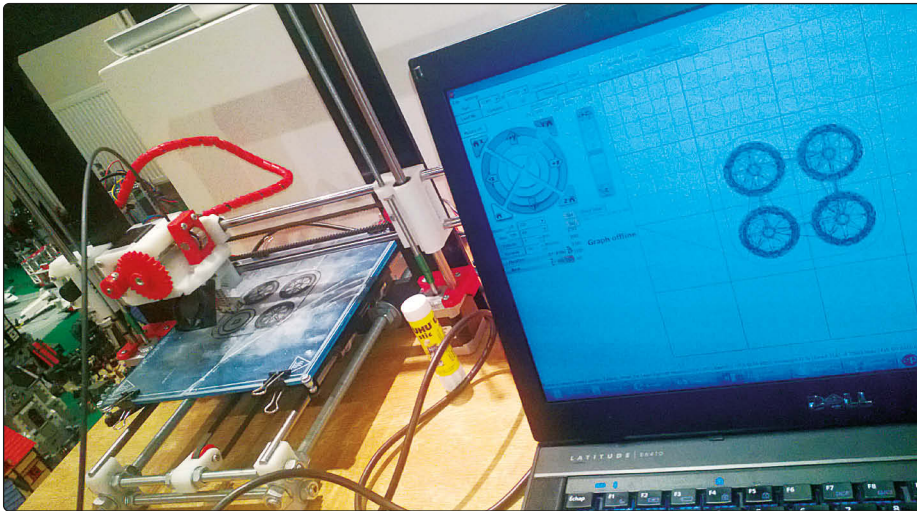
> J'en veux une ! Combien ça coûte ?

Un de mes collègues en république Tchèque, Stepan Bechynsky en possède une. Et évidemment, quand je l'ai vue avec toutes ses possibilités, en bon geek (<http://aka.ms/pid0d5>), je n'ai pu résister à m'en acheter une :-). Me voilà donc lancé à la recherche d'un moyen d'en acquérir une. Et là, plusieurs solutions s'offraient à moi. La première est de la construire soi-même en suivant les indications du wiki de RePrap.org. Et là, rapidement je me rends compte qu'il faut déjà avoir une imprimante 3D pour pouvoir en réaliser une ! Une bonne partie des pièces nécessaires s'impriment. Mince, encore une histoire d'œuf et de poule ! Deuxième solution, en acheter une « sur étagère », mais je me rends compte que je tombe dans le marché professionnel et qu'en réalité il n'existe pas vraiment d'offre « abordable » pour un particulier. Troisième solution, en construire une, mais avec l'aide d'un spécialiste. Et c'est la solution que j'ai choisie. En cherchant un peu je me suis rendu compte que Stepan était passé par Josef Prusa. Et que Josef est très connu dans la communauté de l'impression 3D pour avoir grandement amélioré bien des aspects des imprimantes 3D. Et voilà comment lors d'un voyage en République Tchèque je me retrouve dans un workshop

avec Josef pour monter mon imprimante. Il faut compter environ 1000€ pour une bonne imprimante de ce type, avec le montage, un minimum de consommable et la formation qui va avec. Soit une journée bien remplie.

> Comment ça marche ?

Il existe plusieurs techniques d'impression. Les imprimantes industrielles utilisent des procédés à base de poudre qui avec un liant vont se transformer en céramique. La plupart permettent aussi d'ajouter de la couleur. Les imprimantes 3D plus artisanales et simples fondent des plastiques (ABS, PLA et même nylon) et les déposent en fines couches superposées. En séchant, le plastique durcit et forme l'objet par strates. La qualité de l'objet final va donc dépendre de la précision des éléments mécaniques. Pour le déplacement de l'extrudeur qui fait fondre le plastique, l'imprimante dispose de 3 axes. Ils sont motorisés par des cerveaux-moteurs. Les axes X et Y forment le plan et l'axe Z le volume. Les couches sont donc déposées sur la surface et montées sur l'axe Z.



Vue générale de l'imprimante avec logiciel d'impression en parallèle

L'électronique se compose d'un Arduino Mega avec un shield qui permet le pilotage des moteurs, le chauffage de la plaque du bas, de l'extrudeur, l'alimentation en fil de plastique à fondre, les capteurs pour les positions mini et maxi, le ventilateur de refroidissement, etc. Le montage électronique est finalement assez simple et bien documenté sur le wiki (http://reprap.org/wiki/RAMPS_1.4). Quant à la partie logicielle dans l'Arduino, tout est également assez simple car il s'agit d'interpréter des commandes, soit envoyées par un ordinateur, soit lues sur une carte SD. Les logiciels les plus communs pour cela sont Repetier-host (<http://www.repetier.com/>), Pronterface (<https://github.com/kliment/Printrun>) ou encore ReplicatorG (<http://replicat.org/>). Tous ces logiciels sont open source et fonctionnent sur la plupart des environnements. J'ai une préférence pour Pronterface qui est simple, léger et très efficace.

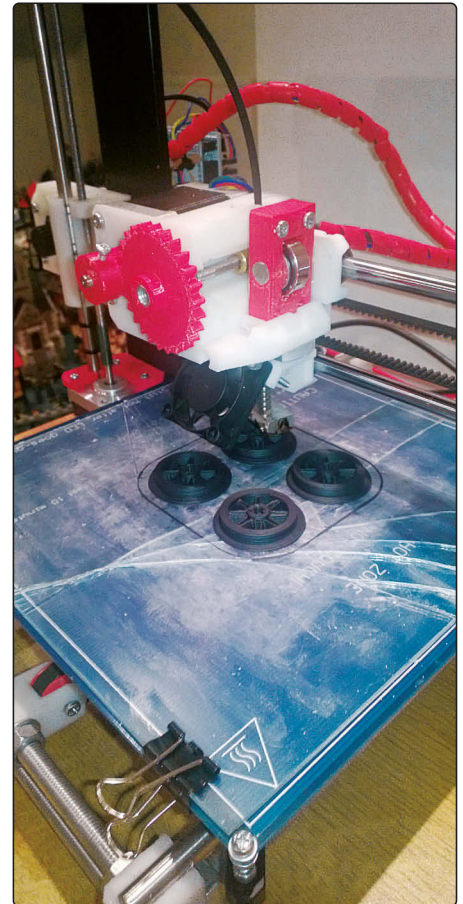
Le langage utilisé pour le pilotage est appelé G-Code. C'est un langage destiné à l'origine pour piloter les commandes numériques. Le G-code est un standard ISO (<http://aka.ms/xgkr20>). Pour les imprimantes 3D, toutes les commandes pour les imprimantes de type RepRap sont très bien documentées (<http://reprap.org/wiki/G-code>). Exemples de G-code :

```
G1 F600.00 E-0.50000
G1 X62.714 Y60.460 F1680.000 E0.01252
M104 S0
M140 S0
M84
```

Ces commandes permettent les déplacements, mais aussi de faire avancer le filament dans l'extrudeur, d'arrêter le chauffage



Roues de train Lego imprimées avec axe Lego normal



L'imprimante en train d'imprimer des roues de train Lego

Interview de Josef Prusa

Quand as-tu commencé à construire ta première imprimante 3D et pourquoi ?

J'ai commencé en 2009 et la raison était simple, j'ai juste pensé que c'était cool :) Rien de plus profond, en y repensant, ce n'était pas aussi populaire, j'ai lancé la mode je suppose.

Pourquoi utilises-tu le modèle RepRap ?

L'idéologie me plaît bien, je peux contribuer et je peux tout modifier quand j'en ai besoin. Si je trouve une nouvelle façon de faire quelque chose, je peux la mettre à jour très rapidement.

Tu mets ton nom partout sur ce que tu réalises, pourquoi ?

Ce nommage est une histoire intéressante, je n'ai pas vraiment appelé ce que je faisais « Prusa ». Mon nom a d'abord été ajouté en tant qu'adjectif dans les premiers designs. Mais la communauté a commencé à appeler les imprimantes

tout simplement « Prusa ». C'est amusant de voir maintenant les designs nommés Darwing, Mendel, Prusa, Huxley... :)

Tu as créé ta propre entreprise autour de l'impression 3D, as-tu beaucoup de demandes et est-ce que ça marche bien ?

C'est incroyable ! Principalement grâce à mes fans tout autour du globe. Chaque jour je fais des envois à travers le monde. C'est hallucinant. Ça marche vraiment bien, mais j'aimerais ajouter que je mets beaucoup d'énergie dans mon entreprise, sans doute plus que la plupart des autres entreprises en république Tchèque.

Si quelqu'un veut construire sa propre imprimante avec toi, comment peut-il faire ?

Il peut regarder <http://reprap.org> et s'il veut en construire une directement avec moi, alors il peut me contacter en envoyant un mail à info@prusaresearch.com

de la plaque du bas, de l'extrudeur, d'arrêter les moteurs.

> Du modèle 3D à l'impression

Vous l'aurez compris, il faut donc générer du G-Code pour imprimer. Et en remontant la chaîne logicielle il faut un logiciel permettant à partir d'un modèle d'objet en 3D de générer ce G-Code. Sachant que la tâche n'est pas facile à partir d'un modèle 3D car il faut pouvoir le découper en tranches, calculer les déplacements, la vitesse d'extrusion, éviter les géométries qui ne peuvent tenir debout et autres problèmes de ce genre... Pour cela, l'outil incontournable est Slic3r (<http://slic3r.org>). C'est l'outil le plus utilisé. Il est lui aussi open source.

En remontant encore, il faut donc des modèles 3D. Et là, vous avez le choix entre créer votre propre modèle en 3D, scanner un objet existant pour le transformer en modèle 3D ou récupérer un modèle 3D existant (gratuit ou payant).

Pour réaliser votre propre modèle en 3D, il faut utiliser un logiciel de CAD. J'utilise OpenSCAD (<http://www.openscad.org/>). Son principal intérêt est d'être comme un outil de développement. Les modèles sont créés par des commandes, comme du code pour ajouter des cubes, sphères, les déplacer, creu-

ser, ajouter de la matière. Il existe de très nombreux autres logiciels. Mais il faut cependant vérifier qu'ils supportent l'export au format stl. C'est le format utilisé en entrée dans slic3r et par d'autres logiciels de conversion en G-Code.

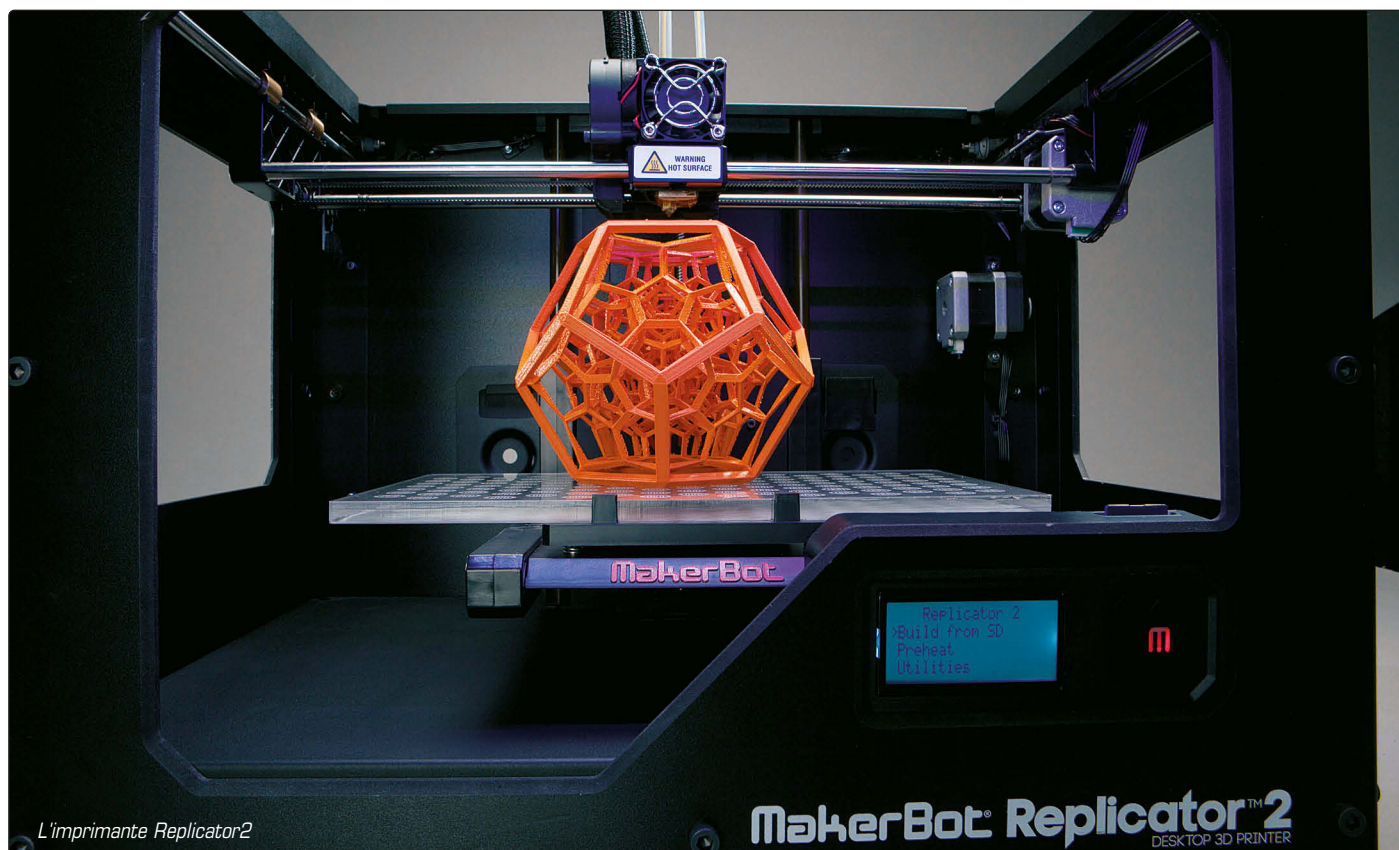
Scanner un objet en 3D avec Kinect ou d'autres outils est encore quelque chose qui n'est pas très répandu. Il existe des solutions professionnelles mais elles sont loin d'être abordables. Il commence à y avoir quelques implémentations plus accessibles mais je n'ai encore rien trouvé de très fonctionnel. A noter le très bon outil d'Autodesk : 123D Catch (<http://www.123dapp.com/catch>), qui à partir de photos reconstitue un modèle 3D d'un objet. Son utilisation est gratuite et donne des résultats plutôt bons sur de gros objets. Reste ensuite à rendre l'objet utilisable en ajoutant des morceaux manquants, polissage des surfaces, etc. Il reste donc toujours du travail, quelle que soit la méthode de numérisation pour obtenir un objet imprimable. C'est une étape qui peut être très longue. Créer des objets complexes en 3 dimensions prend donc beaucoup de temps ! La dernière option est donc de profiter de modèles déjà existants pour réduire ce temps de création. Il existe des modèles payants et gratuits. Le site Thingiverse (<http://www.thingiverse.com/>) est une des

meilleures sources pour découvrir tous types d'objets en 3D. On y trouve vraiment de tout !

> Prise en main et premiers tests

Me voilà donc prêt avec tout ce qu'il faut : l'imprimante est montée, les logiciels installés, tout est paramétré. La première étape consiste à calibrer l'imprimante, rendre le plateau d'impression bien plat, et positionner au mieux le capteur de l'axe vertical (Z). C'est le plus important car si l'axe est un peu trop haut, l'objet ne tiendra pas, s'il est trop bas, le plastique ne sortira pas correctement. Cela se joue au dixième de millimètre ! Cette calibration doit être effectuée à chaque fois que l'imprimante est démontée ou transportée. Avec un peu d'habitude, cela ne prend que quelques minutes.

On commence par chauffer à 90°, dans le cas du plastique ABS, le socle. Il permettra une meilleure adhésion de la première couche et de pouvoir enlever l'objet imprimé facilement. Une plaque de verre est posée sur le socle chauffant. La plaque de verre doit être enduite de colle (type bâtonnet) puis on y dépose un mélange d'acétone avec du plastique ABS dissous. Ce mélange forme une espèce de lait. Il permettra à la première couche de bien tenir sur le verre. Cette première couche est cruciale et si elle ne



L'imprimante Replicator2

MakerBot® Replicator™ 2
DESKTOP 3D PRINTER

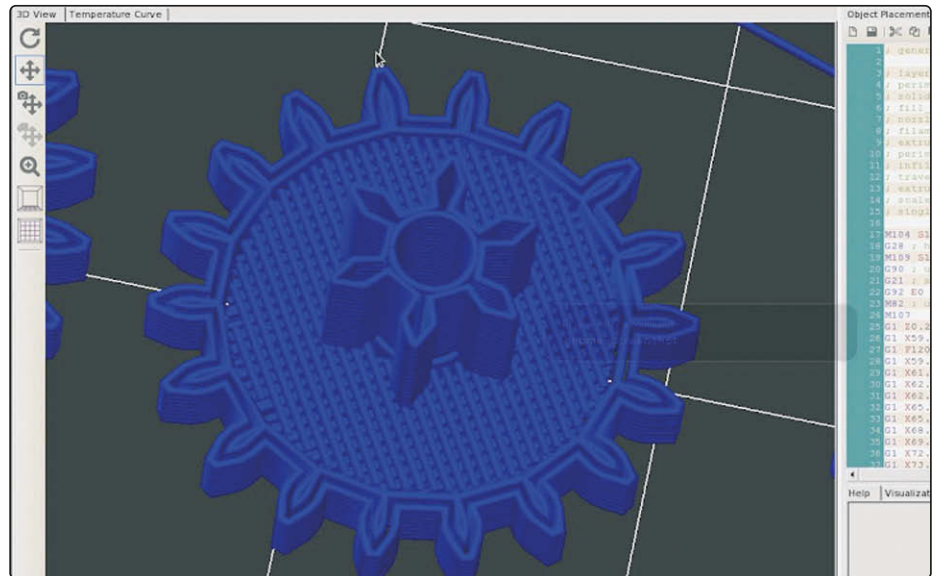


© Shapeways

Dita Von Teese avec une robe entièrement créée avec une imprimante 3D et des modèles de Shapeways.

tient pas bien, autant arrêter l'impression et recommencer. C'est la phase la plus importante. Une fois celle-ci lancée, en général tout se passe bien. Il faut quand même faire attention à plusieurs paramètres. Notamment si l'objet est petit, il faut ralentir l'impression pour laisser le temps au plastique de refroidir avant la couche suivante. Sinon le résultat crée une sorte d'écrasement et déforme l'objet. Aussi, il est souvent plus intéressant d'imprimer plusieurs fois le même petit objet, ce qui laisse ainsi le temps à chaque couche de bien refroidir avant la suivante.

Une fois l'impression terminée, avant que le socle ne soit refroidi, il faut retirer les objets qui se détachent assez facilement. Si la surface refroidit trop, en se rétractant et avec la simple colle et le mélange acétone d'origine, en décollant l'objet, des morceaux de verre peuvent être arrachés. La plaque de verre, si elle est bien entretenue peut servir



Slic3r

à de très nombreuses impressions. Et ce n'est qu'une simple plaque de verre, facile à remplacer en cas de problème.

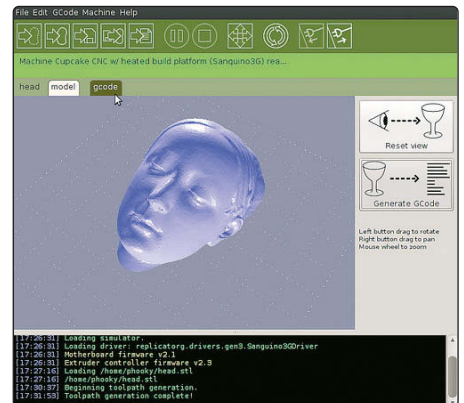
Côté outils nécessaires, il vous faudra une spatule pour décoller les objets, une autre pour gratter la plaque de verre, un flacon pour y mettre le mélange d'acétone, des tournevis et une clé anglaise.

La matière première plastique arrive en bobine. Les bobines sont disponibles dans différentes couleurs. Attention au choix du fabricant ! Il faut préférer des produits fabriqués en Europe car ils ne contiennent pas de matières toxiques et ne sentent pas mauvais à l'impression. Il faut compter environ 25€ la bobine d'un kilo. Pour pouvoir imprimer un objet en plusieurs couleurs, il faut disposer de plusieurs extrudeurs, un par couleur. Sur les RepRap, il est possible d'ajouter un second extrudeur. C'est supporté par le hardware et les différents logiciels. Mais en général, les impressions se font en une couleur. Les objets peuvent ensuite être peints avec de la peinture compatible suivant le plastique utilisé.

Pour assembler des objets entre eux, avec le plastique ABS, il est possible en utilisant le mélange acétone/ABS de coller les objets entre eux. Sinon, l'assemblage par vis peut également très bien fonctionner.

> Quels usages ?

La grande question est toujours : à quoi ça sert ? La plupart des personnes que je connais et qui ont une imprimante 3D l'utilisent vraiment. Cela va du remplacement d'une pièce cassée, à la création d'objets d'art, pour les passionnés de modélisme ou de robotique, c'est un must have. De mon



l'outil ReplicatorG

côté, je l'utilise principalement pour créer des pièces compatibles Lego qui n'existent pas (roues de train spéciales, pièces de montages pour des rails, goulottes pour des câbles). En tout cas, dès que l'on a une de ces imprimantes, de nombreuses portes s'ouvrent à la créativité et cela donne de nombreuses idées ! Pour l'instant, cela reste encore trop compliqué pour une utilisation grand public. Beaucoup de petits problèmes peuvent survenir, beaucoup de choses sont basées sur l'expérience, et les réglages ne sont pas automatiques. Il faut compter environ 1 mois d'apprentissage, d'impressions ratées. Nul doute que les techniques et les modèles vont quand même se démocratiser et qu'il va y avoir, comme dans le monde industriel, des imprimantes, à prix abordable. En attendant, pour notre plus grand plaisir, cela reste un produit pour les geeks :-)



Laurent Ellerbach
Audience Marketing Director
Microsoft EMEA
laurelle@microsoft.com

Les URL courtes

Le phénomène qui consiste à raccourcir les liens des pages web n'est pas nouveau. Cependant, il est de plus en plus utilisé, par conséquent stratégique et principalement boosté avec l'apparition des différents réseaux sociaux, comme Twitter. Maintenant de nombreux sites internet partagent leurs contenus à travers ce format, mais certaines règles doivent être respectées.

La volonté de vouloir raccourcir les URL (ou short URL) évite d'écrire du texte à rallonge dans la barre de votre navigateur. Grâce à ce procédé, vous évitez les erreurs de frappe et les problèmes de césures, des liens coupés à cause de la longueur. Par conséquent vous n'affichez pas dans le navigateur une page 404 pour signaler que la page n'existe pas. Or, les raccourcis des URL ont bénéficié d'un nouveau souffle grâce aux différents réseaux sociaux comme Twitter qui limite le contenu des messages à 140 caractères. Mais aussi des micro-bloggings en général, qui ont su exploiter cet avantage, et éviter ainsi la saisie de nom de domaines et sous-domaines à rallonge.

La décomposition d'un lien URL se présente de la même manière :

http://www.programmez.com/magazine.php?num_magazine=158

et le lien raccourci que nous voulons obtenir, se présente sous la forme suivante : <http://bit.ly/W1HxUh>

Bien entendu, les intérêts sont variés car vous pouvez utiliser ces URL courtes dans vos emails, les signatures électroniques, les sujets des canaux IRC, les flux de contenus pour les plateformes mobiles, etc.

> Construction

Cette volonté de construire ces URL courtes, correspond à un désir de gérer ces liens par soi-même. Mais vous pouvez parfaitement vous appuyer sur une plate-forme déjà existante au lieu de la créer. Quelle que soit la manière choisie, la construction d'URL courtes peut s'effectuer de la façon suivante :

- A travers un site déjà existant
- Fabriquer vos URL courtes

Le semi automatique

Le mode semi automatique s'effectue à travers un des nombreux sites web déjà existants. Les plus connus sont :

- TinyURL
- Bit.ly
- goo.gl (google)
- fb.me (Facebook)

Chaque site propose une certaine longueur de codage, pouvant aller de 5 à 15 caractères suivant la complexité du calcul. Par ailleurs, il existe une API (application), pour que vous puissiez créer des raccourcis à la volée et qu'ils pointent directement sur vos pages web. L'opération s'utilise à travers le langage PHP avec l'extension cURL, qui interrogera l'API de votre choix.

Par exemple, vous pouvez utiliser un script PHP qui va utiliser l'API de TinyURL, à partir d'une URL longue, pour en obtenir une courte.

```
<?php
$url_longue= "http://www.programmez.com/magazine.php?num_magazine
```

```
=158";

$ch = curl_init();
curl_setopt($ch,CURLOPT_URL,'http://tinyurl.com/api-create.php?url='.$url_longue);
curl_setopt($ch,CURLOPT_RETURNTRANSFER,1);
curl_setopt($ch,CURLOPT_CONNECTTIMEOUT,5);
$url_courte = curl_exec($ch);
curl_close($ch);

echo 'url courte : '.$url_courte;

?>
```

Le résultat obtenu est : url courte : <http://tinyurl.com/a9p3tsl>

Ce script s'explique comme ceci : Après avoir initialisé l'extension cURL, nous interrogeons l'API de Tinyurl pour qu'il nous retourne 1 seule réponse. Nous ajoutons que la réponse doit arriver dans les 5 secondes.

Fabriquer vos URL courtes

Avant de commencer à réaliser votre propre développement, vous devez réfléchir à un nom de domaine, de préférence court, mais il doit être aux moins de 3 caractères. Au niveau de l'extension, vous avez le choix car il existe de nombreux « point quelque chose » très courts, c'est-à-dire 2 caractères, et vous pouvez utiliser par exemple .am ou .im. Maintenant que vous possédez un nom de domaine pour les URL courtes, vous devez préparer la base de données et son code PHP associé.

La base de données

La base de données sert à stocker et à réaliser la correspondance entre l'adresse courte et l'adresse longue. Cette opération est nécessaire, car suivant le lien de provenance, l'url d'origine sera retrouvée. Pour cela, vous stockez les informations dans une table MySQL et elle se présente sous cette forme :

```
CREATE TABLE `urls` (
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `url_longue` VARCHAR( 1024 ) NOT NULL ,
  `url_courte` VARCHAR( 6 ) NOT NULL ,
  `date_creation` DATE NOT NULL ,
  UNIQUE ( `url_longue` , `url_courte` )
) ENGINE = InnoDB;
```

Vous pouvez bien entendu faire évoluer cette table en ajoutant un compteur, et une date d'accès.

Ainsi, avec ces informations supplémentaires, vous pouvez obtenir des statistiques journalières.

Calculer

Le calcul d'une URL courte permet de générer une chaîne automatique composée de chiffres et de lettres minuscules. L'algorithme que vous devez mettre en place, va permettre de réaliser une combinaison unique, et suivant le nombre de caractères à proposer, le nombre de combinaisons de liens possibles devient plus important. Pour calculer une URL courte, vous prenez les 10 chiffres et 26 lettres minuscules et vous appliquez la formule suivante : $(10+26)^n$. Le n correspond à la longueur du résultat. Par exemple, si vous souhaitez proposer 5 caractères, vous obtenez 60 millions de combinaisons possibles. Cependant, si le chiffre est de 4, le résultat de combinaison sera de 1,6 million.

Formulaire

Pour obtenir un nouveau lien court, il est nécessaire de proposer une interface aux utilisateurs pour déclencher une URL courte par rapport à une URL Longue. Le formulaire est très simple car vous avez seulement besoin d'un champ de saisie et d'un bouton Confirmer. Bien sur, on peut se dispenser de ce formulaire si le lien est généré à partir de la saisie d'un contenu.

Relation avec une base de données

Lors de la validation du formulaire, il est important de créer une action pour sauvegarder les données dans une base de données (par exemple MySQL). Pour effectuer l'opération, vous devez calculer l'URL courte pour ensuite générer la requête SQL, qui se présente sous la forme suivante :

```
INSERT IGNORE INTO `url` (url_longue, url_courte, date_creation)
VALUES ('$url_longue', '$url_courte', '$date')
```

Il s'agit d'ajouter dans la base de données l'url longue et la date de création. Le champ 'url_courte' est automatiquement rempli par l'intermédiaire du calcul mathématique.

Lecture

L'accès au contenu d'une page, par l'intermédiaire d'une URL courte s'effectue en interrogeant la table de la base de données. Il est nécessaire d'appliquer les filtres de sécurité (voir plus loin) et d'interroger la table, de la façon suivante :

```
SELECT url_longue FROM `url` WHERE url_courte = '$_GET[url_courte]';
```

Le résultat renverra au navigateur la page obtenue.

> Validation

Quelle que soit la méthode d'utilisation ou création de votre plateforme d'URL, il est important de vérifier si celle-ci est conforme. Cependant, la validation d'une URL courte peut aussi s'appliquer sur les liens externes. Pour vérifier et éviter les risques d'erreurs, vous devez effectuer un certain nombre de tests et installer certains outils dans votre ordinateur pour améliorer la sécurité.

Vérification

Pour vérifier si vos liens sont valides ou tester les URL des autres, vous pouvez utiliser le site 'Check Short URL' (<http://checkshorturl.com/>). Il est compatible avec l'ensemble des fournisseurs de réduction d'URL : bitly, Google... Il s'agit d'un site web qui va tester si le lien qui vous est proposé est bon et par la même occasion fournir certaines informations venant de ce lien :

- L'URL longue
 - Le temps de chargement de la page
 - Le titre de la page
 - La description de la page
 - Une prévisualisation de la page
- Nous allons réutiliser la première URL courte.

Sécurité

Quelle que soit la tendance du web, cela attire de nombreux aventuriers et par conséquent un certain nombre d'acteurs veulent tester la résistance de ce concept. Les URL courtes n'échappent pas à la règle, avec différents risques d'attaques dont tout le monde peut être victime. Les risques sont nombreux et très variés. Ils se traduisent par différents types d'attaques malicieuses, comme :

- Le Phishing ou vol d'informations pour usurper l'identité
- Les attaques Xss
- La recherche de failles dans les navigateurs
- Les SPAM
- Repère associé

L'ensemble de ces points est régulièrement traité dans le magazine. Toutefois, le repère associé, est quelque chose d'assez nouveau. Il s'agit d'associer une extension connue à une URL courte différente, par exemple un cas a été dévoilé en aout 2012, sur l'utilisation des URL courtes « .gov » et « .mil » et l'on a pu voir des liens de cette forme 1.usa.gov. Le but final est bien entendu de tabler sur le fait que l'internaute ne va pas vérifier ni contrôler l'URL car l'extension présentée est connue et reconnue. Ici il s'agit de l'extension du domaine du gouvernement des Etats-Unis.

> Solution

Venant de la plateforme

Il existe de nombreuses solutions payantes pour assurer la qualité du service mais aussi la possibilité d'associer une URL courte à une compagnie ou société, par exemple : nyti.ms associé au New-York Times.

Venant de l'internaute

La solution que chaque internaute peut effectuer facilement, consiste à utiliser les extensions de son navigateur pour se protéger, ou le lien de destination avant de cliquer. Vous trouverez au niveau des add-ons par exemple :

- Mozilla Firefox : Xpnd.it!
- Google chrome : View Thru
- Opéra : Shorten url

> Conclusion

L'utilisation des URL courtes est un phénomène qui remporte un fort succès. Cependant, il rencontre quelques limitations sur le court et le long terme. Tout d'abord, le service de redirection, que vous utilisez, peut devenir inaccessible ou arrêter son activité du jour au lendemain, ce qui provoquerait un lien qui n'est plus accessible. Par ailleurs, la qualité du service peut présenter des inconvénients pour vous, lorsque la redirection est associée à l'ouverture de pop-up de partenaires du service concerné. Toutefois, il est important de vérifier et de tester les différents services et surtout de se prémunir des liens douteux.

Christophe Villeneuve

Consultant pour Hello-Design, auteur du livre « PHP & MySQL-MySQLi-PDO, Construisez votre application », aux Éditions ENI. Membre des Teams DrupalFR, AFUP, LeMug.fr, PHPTV.

Choisir son PaaS Java

Nous sommes en l'an 2442, je me promène dans un parc pendant une belle journée ensoleillée avec les toutes dernières lunettes G.Glass sur les yeux et là, j'ai une révélation, je veux établir la carte du bonheur en France en relevant avec le réseau des « G.Glass » le nombre de sourires en temps réel et rendre cette carte disponible au grand public.

Rien de plus facile, je dispose d'un SDK très avancé qui réagit à la voix (on est en 2442 toujours) auquel je commande l'instruction du calcul souhaité, je commande aussi une interface web et indique aussi l'instruction pour afficher le résultat avec un smiley sur la carte. Pour finir, je commande (toujours par la voix) de déployer l'application web sur mon PaaS préféré. Je sais ce que vous allez dire, encore un qui a oublié les tests unitaires...

Disposer d'un environnement complet, accessible à travers internet, permettant à tout programmeur de développer et déployer en presque un clic son application web et la rendre accessible instantanément à des millions d'utilisateurs, ça a toujours été mon rêve ! Cet environnement a un nom, PaaS pour Plate-forme As A Service. Mais au fait, qu'est-ce qu'un PaaS ? Un PaaS est une *plateforme de développement* hébergée dans le nuage informatique et qui permet aux développeurs de construire une application web sans installation d'outils spécifiques sur leur poste de travail et leur permet le déploiement de l'application web sans compétences d'administrations systèmes spécifiques.

Mais pourquoi délaisser le bon vieux serveur physique (dédié ou virtuel) et nos bon vieux outils de développement et aller vers une nouvelle solution. Une chose est sûre, l'avenir est au PaaS mais, auquel ? Pour quels services ? Quels sont les critères à prendre en compte pour choisir le « bon » PaaS ?

Afin d'amener des éléments de réponse à nos interrogations nous allons tout d'abord, définir l'ADN d'un PaaS, les services qui, selon moi, doivent impérativement faire partie du package d'un PaaS, puis nous passerons à un comparatif des principales solutions PaaS du marché tout en nous concentrant sur ceux qui proposent Java comme plateforme et langage. A la fin de l'article nous nous intéresserons à Google Cloud Platform pour voir ce que cette dernière propose.

LES 7 CRITÈRES CAPITAUX

Même si de plus en plus d'applications web sont utilisées sur des PaaS, ces derniers sont actuellement considérés comme des technologies jeunes et non encore stabilisées, preuve en est, la différence existant entre les divers acteurs du marché.

Chacun des fournisseurs PaaS propose une multitude de services autour de sa solution, même si ces dernières sont d'ordre expérimental ; l'idée est de regarder et d'analyser le comportement des développeurs face aux services proposés et privilégier ce qui pourrait être le plus innovant et distinctif dans un marché assez concurrentiel mais non encore saturé.

Mais tout PaaS doit avoir un ensemble de services essentiels qui me paraissent vitaux et capitaux. J'en dénombre 7 que je vais vous présenter. Nous établirons ensuite une échelle nous permettant de leur donner une note de criticité/utilité que nous exploiterons durant le comparatif entre les PaaS.

1. Langages de programmation et framework

Voici le dilemme : est-il opportun pour un fournisseur PaaS d'offrir une large palette de langages et de frameworks afin d'attirer le plus possible de programmeurs sur sa plateforme ou plutôt de se concentrer sur un langage et un framework afin d'en avoir la maîtrise totale.

Faudrait-il proposer des langages standards ou inventer son propre langage (comme ce que fait Salesforce sur Force.com ou Google avec Go ou Dart).

En tant que développeur Java essentiellement, le PaaS que j'utiliserai doit impérativement me permettre de développer et déployer des applications web Java, et aussi d'embarquer des frameworks venant du monde de l'Open Source (Spring, Hibernate, etc.). et toujours en tant que développeur Java, je souhaite aussi avoir la possibilité de

développer mes applications avec un des langages de programmation destinés à la JVM (Scala, Groovy, Ruby).

En définitive, il n'y a pas de règles précises, mais l'histoire nous enseigne que les plateformes ouvertes sont plus à même de faire adhérer des communautés très actives et créatives, ce qui en général pousse le fournisseur à exceller en matière de stabilisation et performance du PaaS proposé.

2. Bases de données

C'est la partie la plus sensible dans un PaaS. Toute l'intelligence métier se trouve en général dans une base de données. Les fournisseurs PaaS proposent une large palette de choix, chacune pour un besoin bien précis. Cependant le monde des bases de données n'est plus l'apanage d'Oracle ou d'IBM car un mouvement nommé NoSQL, qui a émergé depuis quelques années, commence à grignoter des parts de marché de plus en plus importantes et répond à des problématiques que les RDBM classiques ne permettent pas nativement, comme les Blobs ou encore les temps de réponses optimisés.

Néanmoins, le monde SQL est sur la place depuis bien des années et tellement enraciné dans le monde des entreprises qu'imaginer que ce dernier puisse disparaître au profit du NoSQL n'est probablement pas pour demain...

Pour résumer, je pense qu'à terme, un mix des deux mondes devra être envisagé, ce qui est d'ailleurs proposé par plusieurs fournisseurs de PaaS, et qui permettra de pouvoir intégrer les bases en production et d'entamer les futurs développements sur du NoSQL pour le *front* et du SQL pour reprendre des principes métiers spécifiques.

Il faut impérativement vérifier que le fournisseur PaaS permet la localisation des données (Europe, USA...) et de quels moyens on dispose pour récupérer les données sans surcoûts.

Reste le fait d'exposer les données dites sen-

sibles des entreprises sans pour autant les déplacer physiquement sur le PaaS. Plusieurs solutions existent et notamment l'établissement d'un tunnel sécurisé entre le PaaS et les données.

3. Disponibilité

C'est le contrat qui lie le développeur et le fournisseur et qui porte le nom de SLA (Service Level Agreement). Pour une boutique en ligne par exemple, pouvoir vendre ses articles 24h/24h 7j/7, ce qui correspond en général à un SLA de 99,99% (le SLA varie selon le fournisseur et le service, NDLR). L'application web devra donc être accessible tout le temps et doit aussi supporter la charge (nombre de requêtes, nombre de sessions, ...) pour cela, une étude approfondie des contrats proposés par les fournisseurs PaaS est à prendre très au sérieux car cela engage les deux parties et doit désigner clairement les responsables en cas d'incidents.

4. Sécurité

Un thème tout aussi critique et sensible, la sécurité, car il s'agit de l'accès même aux données et aux services de l'application web. Un fournisseur PaaS doit donc être en mesure de protéger l'ensemble des applications web déployées ainsi que les données liées. Plus encore, si ce dernier propose l'hébergement du code source ou les binaires propriétaires de l'application, il est dans l'obligation de protéger ses éléments applicatifs et dans le cas externe (la perte d'un ou de plusieurs data centres) la restauration des données dans leur état avant l'incident. La sécurité concerne aussi la non-divulgaration et la protection contre le piratage de l'ensemble des composants de l'environnement de développement, la protection contre des attaques du genre DoS ou autres. Un volet tout aussi important est l'authentification applicative. Plusieurs choix de protocoles existent, le plus utilisé, OAuth, dont la dernière version 2.0 ajoute plus de restrictions pour le client.

Note de la rédaction : vous pouvez vérifier que toutes les connexions se réalisent en https / ssl. Les connexions distantes peuvent se faire en protocole RDP ou en SSH.

5. Options et Services

Chaque fournisseur PaaS voudrait se démarquer de ses concurrents et propose des services et options afin d'attirer plus de développeurs sur sa plateforme. Cela va de

proposer des langages de programmations différents jusqu'à offrir une usine de développement (comme ce qui se fait chez nos amis Cloudbees).

6. Transparence, communication et support

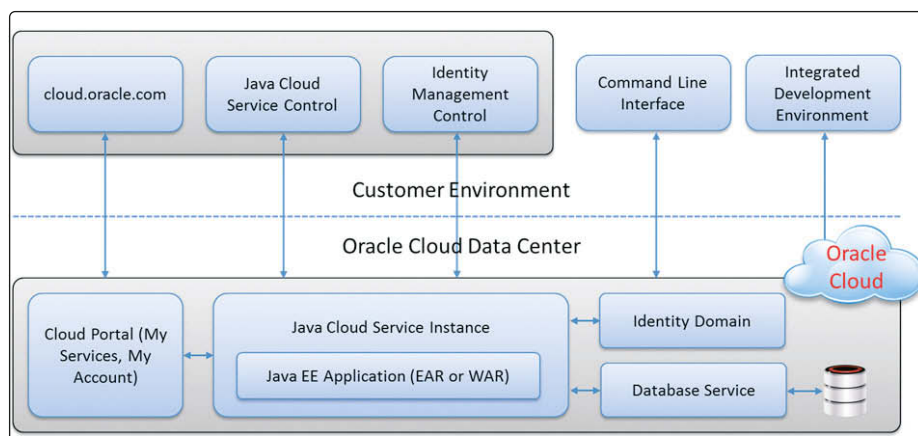
Selon votre abonnement le support est un élément vital sans quoi le moindre blocage deviendrait un cauchemar. Suivant le type d'abonnement pris chez votre fournisseur, le support peut varier entre communautaire et 24/24 et 7/7, ou même une intervention sur votre application par votre fournisseur PaaS. La communication et la transparence en cas de panne ou incident de votre fournisseur doit être la plus claire possible (communiqués publics ou spécifiques selon le cas). Reste l'aspect du coût, ce dernier doit identifier clairement le coût mensuel ainsi que les coûts en cas de dépassement ou à la consommation à la demande, un peu comme ce qui se passe avec votre opérateur mobile sauf que les unités de mesures diffèrent d'un fournisseur à l'autre, mais en règle générale, j'ai constaté que les coûts sont globalement bien expliqués et que les quotas gratuits sont largement suffisants pour pouvoir bénéficier d'un véritable environnement de tests.

7. Développement et déploiement

Publier son application d'un simple clic, pouvoir émuler la plateforme en local sur le poste de développement sont des éléments à ne pas négliger et qui s'avèrent être des alliés de poids pour être productif et réactif. La majorité des PaaS propose des plugins pour les IDE les plus populaires, Eclipse ou IntelliJIDEA, des plugins Maven existent aussi qui permettent l'intégration des déploiements dans les usines de développements et d'automatiser ainsi cette phase.

D'autres PaaS permettent tout simplement l'exécution sur le poste de développement du même environnement du PaaS, comme ce qui se fait avec OpenShift ou VMWare Cloud Foundry, et qui permet tout simplement de disposer de son propre PaaS privé.

Dans la plupart des cas, l'équipe dispose de trois environnements, un environnement de développement, un environnement de staging et enfin, l'environnement de production. On pourrait donc, et dans le cas de l'utilisation de Maven, utiliser autant de profils Maven que d'environnements configurés, le tout sur un Jenkins qui s'occupera du déploiement continu sur chacun des environnements.



Le cloud PaaS Java d'Oracle et ses tarifs

Java S1	Java S2	Java S4
\$249 / Month	\$499 / Month	\$1,499 / Month
Buy Now	Buy Now	Buy Now
1 Oracle WebLogic Server ¹	2 Oracle WebLogic Servers ¹	4 Oracle WebLogic Servers ¹
1.5 GB RAM for Java Heap ²	3 GB RAM for Java Heap ²	6 GB RAM for Java Heap ²
5 GB File Storage ³	10 GB File Storage ³	25 GB File Storage ³
50 GB Data Transfer ⁴	250 GB Data Transfer ⁴	500 GB Data Transfer ⁴

L'ÉCOSYSTÈME JAVA ET LE CLOUD

Java représente une part non négligeable du Cloud, de par sa notoriété non discutable en ce qui concerne le développement d'applications web (et aussi par le patrimoine applicatif, NDLR). Cependant, certains fournisseurs PaaS n'incluent pas Java et sa plateforme associée de façon homogène et il n'existe pas un pattern qui déterminerait comment cela doit se faire.

Prenons l'exemple de OpenJDK. L'implémentation open source de Java reste absente pour le moment dans la plus part des PaaS que j'ai pu étudier et la raison selon moi vient du fait que cette dernière ne s'est pas encore démocratisée dans la communauté des développeurs. Néanmoins, la version 7 de Java est, quant à elle, proposée par l'ensemble des PaaS. Un autre volet tout aussi important et critique est celui de la plateforme JEE. Très peu de PaaS proposent cette plateforme en version JEE 6. On peut citer Cloudbees avec GlassFish ou encore OpenShift via JBoss AS. Oracle propose aussi JEE via Oracle Cloud ou encore IBM Cloud.

LES SOLUTIONS ACTUELLES DU MARCHÉ

Il existe une multitude de solutions PaaS sur le marché, chacune d'elles offrant des services qui peuvent varier d'un fournisseur à l'autre. Dans ce qui suit, je vais vous présenter un échantillon non exhaustif de quelques-uns des fournisseurs PaaS qui, selon moi, proposent des services différents et apportent donc une plus-value intéressante à prendre en considération.

> Heroku

Avec un nom qui ressemble plutôt à un titre de BD manga, Heroku propose une multitude de services (addons) et une approche innovante, le tout hébergé sur des machines chez Amazon. Au début, Heroku proposait essentiellement du Ruby on Rails, une grande partie des tâches devant être exécutées en ligne de commande, ce qui fait d'Heroku un PaaS efficace mais qui comporte un effort d'apprentissage élevé. Heroku propose Java, Python et plus récemment NodeJS. Heroku cependant appartient depuis 2010 à Salesforce et héberge un peu moins de 200 000 applications.

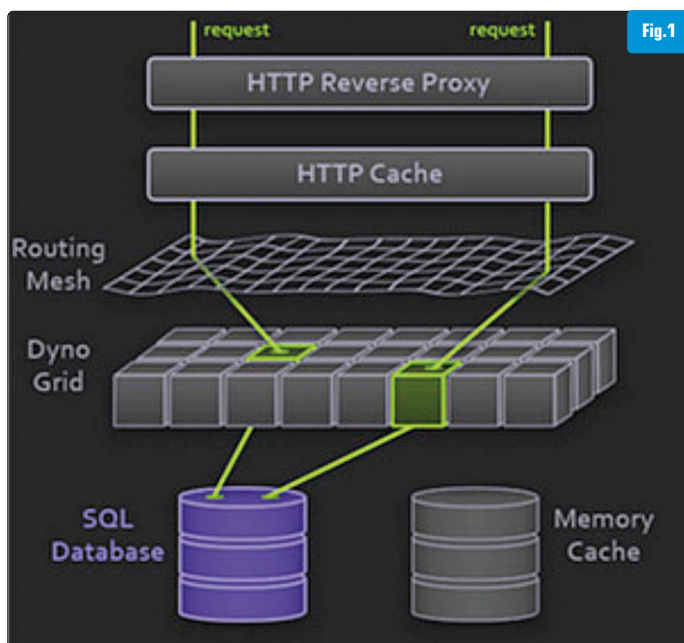


Fig.1

L'unité de mesure des processus est le Dyno, exemple d'une instance Cedar Stack :

- 35 GB Ram
- 13 EC2 unités de calculs
- dans les 120 instances Dynos [Fig.1].

> Windows Azure (Microsoft)

Bien qu'ayant une longue expérience du service informatique dans les entreprises, j'avoue avoir été bluffé lors des premières présentations d'Azure, notamment par le grand choix que ce PaaS/IaaS propose. Azure répond parfaitement aux critères cités ci-dessus et complète la liste en offrant beaucoup plus, par exemple les redondances Géo localisées ou locales, une gestion des rôles des utilisateurs efficaces, la possibilité de l'utilisation side-by-side de plusieurs versions du SDK, l'intégration avec Visual Studio, etc. Il y a plusieurs façons de déployer des applications web Java sur la plateforme Azure, le site dédié à la plateforme regorge d'informations très utiles et de *tutos* très détaillés permettant à tout développeur de bien commencer ses développements. Une des façons de faire pour déployer une application web Java passe par la création d'une machine virtuelle sur laquelle on installera le JDK de notre choix puis un conteneur Servlet (Tomcat, jetty, etc.), puis configurer les EndPoints (les accès vers/depuis l'extérieur pour la VM) et pour finir ouvrir les ports nécessaires du firewall, le tout via un processus étape par étape.

Lien : <http://www.windowsazure.com/en-us/develop/java/>

Et <https://github.com/Microsoft/WindowsAzureToolkitForEclipseWithJava>

> OpenShift (RedHat) [Fig.2]

OpenShift est la solution PaaS de chez RedHat, ce dernier propose un large choix de langages (Java, NodeJS, PHP, Ruby ...), il propose aussi Jenkins, Maven, un plugin Eclipse, Appcelerator. Plusieurs types de bases de données (MySQL, MongoDB ... etc.)

Mais ce qui rend OpenShift différent est le fait que les composants de la plateforme Openshift sont open source et donc exploitables et permettent à toute personne de créer son propre PaaS qui pourrait ensuite être hébergé sur des VM ou en local sur la machine de développement. Il faut noter que la même chose est proposée avec Cloud Foundry depuis quelque temps déjà, mais les composants de la plateforme ne sont pas open source.

> Google Cloud Platform

Le PaaS de Google est un environnement totalement "managé" par Google. Le développeur n'a aucun moyen d'installer ou d'utiliser autre chose que ce que Google propose et déploie. Une telle façon

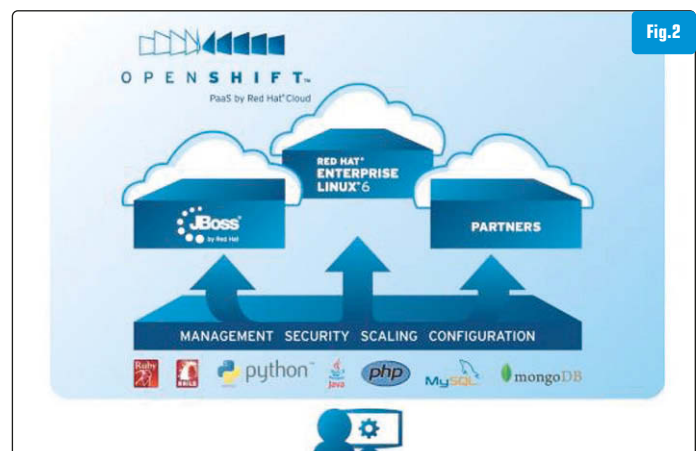


Fig.2

de faire comporte ses avantages et ses "inconvenients". Un aspect qui pourrait être considéré comme un inconvénient par certains est le fait de ne pas pouvoir disposer des versions spécifiques de telle ou telle partie de la plateforme, ce qui réduirait le facteur "tunner" du développeur/administrateur. C'est exactement ce qu'un fournisseur comme Google essaie de réduire afin que le développeur puisse se concentrer sur ce qu'il fait de mieux : développer des applications performantes et laisser la tâche du choix des versions des OS à Google qui utilise la même plateforme pour fournir des services mondialement sollicités par des millions d'utilisateurs tels que Youtube et Gmail. Donc, exécuter son application web sur le PaaS de Google revient à l'exécuter sur les mêmes data centres que Gmail ou YouTube. Voici les principales qualités qui donnent au PaaS de Google de sérieux atouts face à ses concurrents.

1. Google App Engine (GAE)

GAE est un environnement de développement et d'exécution d'applications web. Un plugin Eclipse permet très facilement de développer l'application et de la déployer en quelques clics quelle que soit sa complexité. GAE offre aussi de généreux quotas gratuits qui, bien gérés, sont largement suffisants pour exploiter la plateforme pour des applications gourmandes en ressources ou en nombre de requêtes. Voici un aperçu des prix et quotas de la plateforme :

Resource	Free Default Limit		Billing Enabled Default Limit	
	Daily Limit	Maximum Rate	Daily Limit	Maximum Rate
Channel API Calls	657,000 calls	3,000 calls/minute	91,995,495 calls	32,000 calls/minute
Channels Created	100 channels	6 creations/minute	Based on your budget	60 creations/minute
Channels Hours Requested	200 hours	12 hours requested/minute	Based on your budget	120 hours requested/minute
Channel Data Sent	Up to the Outgoing Bandwidth quota	22 MB/minute	2 GB	740 MB/minute

GAE fournit une base de données NoSQL, l'exécution des tâches en asynchrone, un Dashboard permettant le contrôle de l'état de l'application, la consultation des Logs, la gestion de la sécurité, etc.

2. Google Cloud Endpoints

Il s'agit d'une nouveauté de la version 1.7.5, qui consiste en une fonction expérimentale permettant aux applications Android et iOS l'accès aux services Google App Engine [Fig.3].

3. Google Compute Engine (IaaS)

Des machines virtuelles Linux hébergées sur l'infrastructure de Google, permettant l'exécution de calculs lourds et qui peuvent "scaler" vers l'utilisation de plusieurs milliers de processeurs selon la complexité des traitements des instances.

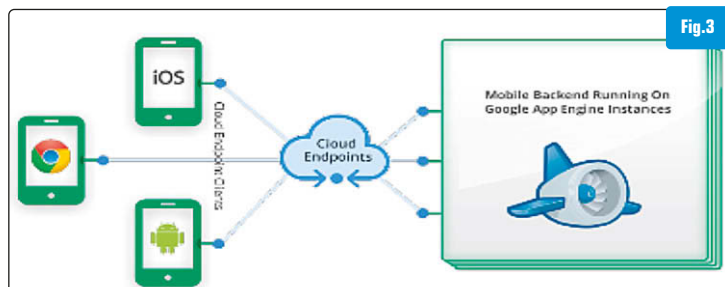


Fig.3

Configuration	Virtual Cores	Memory (GB)	GCE16 vCPU 16.0GB	Local Disk (GB)	Price/hour	\$/GCE16hour
n1-standard-1-d	1	3.75	2.75	420	\$0.145	\$0.053
n1-standard-2-d	2	7.5	5.5	870	\$0.29	\$0.053
n1-standard-4-d	4	15	11	1770	\$0.58	\$0.053
n1-standard-8-d	8	30	22	2 x 1770	\$1.16	\$0.053

4. Google Cloud Storage

Permet le stockage et la gestion de volumes presque infinis de données directement sur la plateforme de Google.

Storage Pricing (per GB per month)

Monthly Usage	Standard Storage	Durable Reduced Availability Storage
First 0-1 TB	\$0.085	\$0.063
Next 9TB	\$0.076	\$0.054
Next 90TB	\$0.067	\$0.049
Next 400TB	\$0.063	\$0.045
Next 4500TB	\$0.054	\$0.042

5. Google BigQuery

Permet le requêtage de gros volumes de données se trouvant dans le Cloud (pas nécessairement chez Google) pour des fins d'analyse en temps réel, le tout en quelques secondes quel que soit le volume de la requête

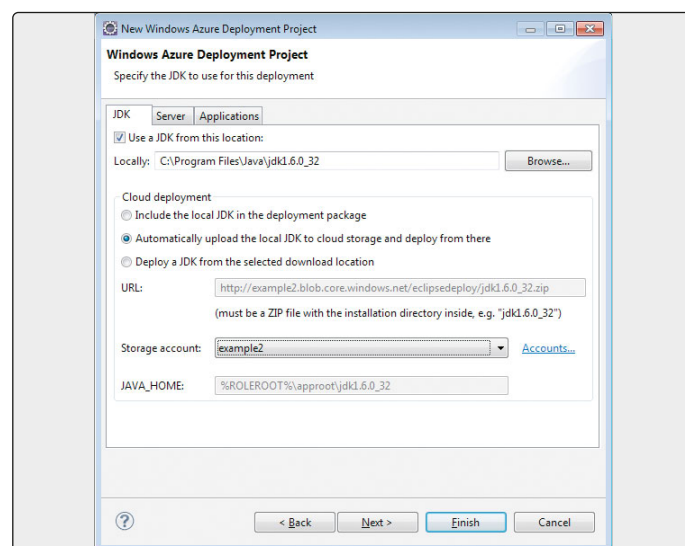
6. Google Cloud SQL

Un autre service totalement "managé" par Google et qui donne accès à une ou plusieurs instances MySQL, ce qui permet au développeur de porter sa totale concentration sur le côté métier et non plus sur l'optimisation de la base de données.

Pour avoir plus d'information concernant la plateforme Google Cloud je vous invite à aller faire un tour sur le site dédié qui permet, entre autres, de s'enregistrer gratuitement et de commencer instantanément à développer et déployer une application web profitant des services mis à disposition. Pour conclure, développer et déployer des applications web n'a jamais été aussi simple et à la portée de tous les développeurs. Du Java au Ruby en passant par Grails, Scala ou Python ou même Javascript, tous les langages majeurs ainsi que leur bibliothèques sont désormais disponibles dans le Cloud afin de nous permettre d'innover et de développer encore plus de services mondialement utiles.

Abderrazak Bouadma

Tête dans les nuages et les pieds sur terre - Architecte chez SFEIR @mbtdoor



Les outils indispensables du développeur

Si les bons outils ne font pas nécessairement les bons développeurs, ils peuvent par contre les aider à gagner en efficacité. C'est l'objectif de cet article qui présente une liste d'outils qui viennent enrichir efficacement le couteau suisse du développeur !

La liste d'outils présentée ici ne se veut pas exhaustive, ni d'ailleurs objective : le choix d'un outil plutôt qu'un autre ne se faisant pas dans l'absolu mais dans un contexte particulier, et pour une personne en particulier. Nous essayons néanmoins de proposer une palette d'outils suffisamment large pour couvrir des besoins divers au cours de la vie d'une application, le développeur étant souvent amené à revêtir différentes casquettes (architecte, administrateur système, designer, ...). Les outils présentés se veulent, dans la mesure du possible, légers et complémentaires d'un IDE (souvent plus lourd).

> L'édition de code source

Pour cette tâche, les IDE modernes sont tout indiqués car ils intègrent bien plus que le simple éditeur de code (gestion de versions, transfert de fichiers, refactoring, debug...). Pourtant leur poids n'en fait pas un outil pratique pour l'édition rapide ou pour un simple coup d'oeil sur un fichier. Voici deux outils qui permettent d'épauler efficacement un IDE pour des cas d'utilisation bien précis.

Notepad++ [Fig.1]

La concurrence fait rage sur ce secteur mais ce petit éditeur de texte sous licence GPL sait tirer son épingle du jeu. Il est léger, gère la coloration syntaxique de nombreux formats de fichiers, et propose beaucoup de fonctionnalités complémentaires qui peuvent par ailleurs être étendues par l'ajout de plug-ins. De plus, son interface est entièrement

configurable par l'utilisateur. Difficile de lui trouver des points faibles !

Alternatives : dans ce domaine, il y en a pour tous les goûts. Notepad2 (Windows, Licence BSD) et SciTE (Linux et Windows) utilisent le même moteur que Notepad++ (SciTEntilla). Sous Linux, les différents environnements graphiques proposent des éditeurs complets à l'image de Gedit (également disponible pour Windows et OS X) sous Gnome. Au croisement du simple éditeur de texte et de l'IDE très léger, on peut citer Geany (Windows, OS X, Linux, sous licence GPL et basé sur SciTEntilla) et PSPad (Windows uniquement, Freeware). Pour les irréductibles Linuxiens de la première heure, nous ne pouvons pas ne pas évoquer les incontournables Vi et Emacs (nous ne nous risquons pas à tenter de donner une préférence ici !). Sous OS X une très bonne alternative est TextWrangler, ou encore Smultron, qui s'intègre bien à l'interface d'OS X.

Codiad [Fig.2]

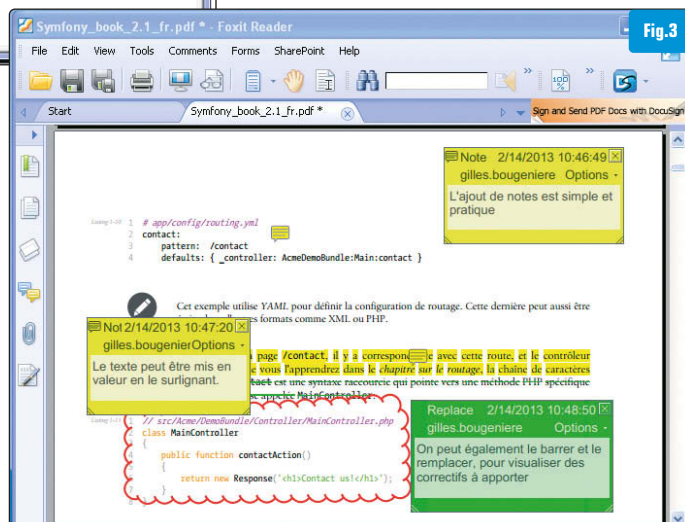
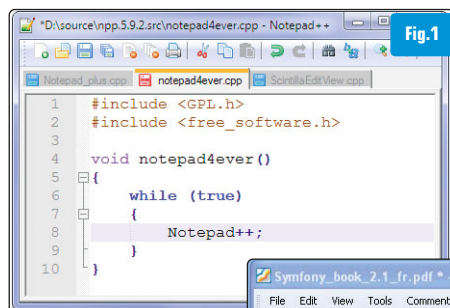
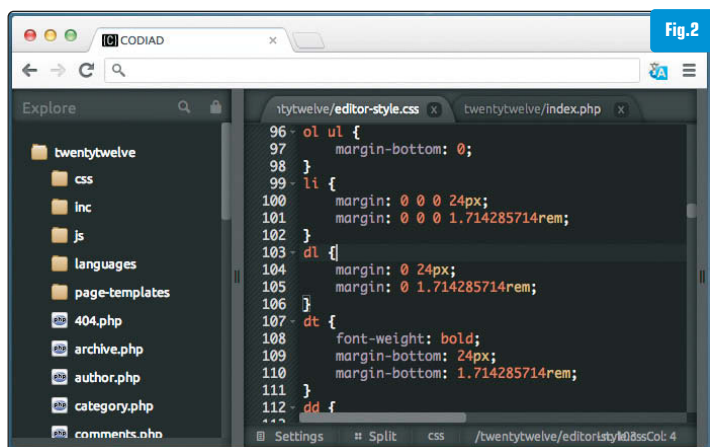
Il s'agit d'un éditeur de code source sous forme d'application web sous licence MIT. Il est donc accessible depuis n'importe quel

support connecté, comme une simple tablette munie d'un clavier Bluetooth. Très simple à utiliser, Codiad gère la coloration syntaxique, l'auto-complétion, plusieurs utilisateurs et plusieurs projets. Il lui manque cependant des fonctionnalités de collaboration un peu plus poussées que l'on s'attend pourtant à trouver sur ce type d'outils en ligne.

Alternatives : Il existe de nombreux éditeurs de code source en ligne en mode SAAS (Software As A Service), regroupant plus de fonctionnalités, notamment sur l'aspect collaboratif. Citons par exemple Cloud9 ou Codenvy. Mais on sort ici du cadre du petit outil simple et pratique pour se rapprocher du véritable IDE. Par ailleurs, l'avantage de Codiad est bien de pouvoir maîtriser l'hébergement de ses propres fichiers, et accessoirement de pouvoir modifier directement les fichiers du serveur voulu, sans avoir recours à un logiciel de transfert de fichiers (attention cependant à ce cas d'utilisation qui requiert de prendre des précautions particulières vis à vis de la sécurité).

> Conception, documentation et prises de notes

Durant les différentes phases de vie d'un projet, les développeurs sont amenés à manipuler de la documentation, à utiliser différents outils de conception, et à gérer une certaine quantité de notes prises tout au long du projet. Voici quelques outils qui peuvent s'avérer utiles en ce sens.



Foxit Reader [Fig.3]

Disponible sous Windows ou Linux (mais pas mis à jour depuis 2009 pour cet OS), Foxit Reader est un lecteur PDF un peu plus léger qu'Adobe Acrobat Reader et qui permet l'annotation des documents PDF. Malheureusement, les dernières évolutions de Foxit Reader n'ont pas joué en faveur de sa légèreté. De plus, il faut être attentif lors de l'installation pour éviter la pollution du système par la barre d'outils incluse dans l'installateur.

Alternatives : Sous OS X, il n'est pas nécessaire d'installer d'outil supplémentaire car Aperçu, disponible par défaut et donc parfaitement intégré à l'OS, permet de consulter et annoter les PDF. Il est cependant moins riche en fonctionnalités que Foxit et ses fonctionnalités ne sont pas mises en avant. C'est un peu le même cas de figure sous Linux où le lecteur de PDF fourni par défaut avec Gnome, Evince, permet l'annotation des documents PDF, mais de façon peu accessible dans l'interface. Pour accéder plus simplement aux annotations de PDF, on peut alors installer Xournal, donc le but premier est justement la prise de notes sur les documents PDF. Sous Windows, il existe plus léger que Foxit Reader, avec Sumatra PDF, mais ce dernier se contente du strict nécessaire en ne permettant que la simple consultation des PDF.

Microsoft Office Viewers

Pour lire ou imprimer mais rien de plus des documents de Microsoft Office sans faire l'acquisition d'une licence, on peut se contenter des "viewers" mis à disposition gratuitement par Microsoft. Ils sont téléchargeables séparément pour Word, Excel et Power Point.

Alternatives : On peut également bénéficier

de quelques fonctionnalités d'édition avec des outils libres (licence GPL) tels que Gnumeric et AbiWord, plus légers que leurs homologues Excel et Word (Microsoft Office), ou que Writer et Calc (Open Office / LibreOffice), mais aussi beaucoup moins complets... et c'est sans parler des problèmes de compatibilité !

Tomboy [Fig.4]

Avec Tomboy, outil multiplateforme (Mono) disponible sous licence LGPL, on organise ses notes comme on le ferait avec un wiki (wikipedia, docuwiki, ...). Les différentes notes peuvent donc être liées les unes aux autres de façon très simple. L'avantage par rapport à un bloc-notes classique est de pouvoir ordonner efficacement ses prises de notes et de passer moins de temps à les rechercher par la suite (qu'il est énervant de ne pas retrouver l'unique note qui nous intéresse dans cette pile de papiers... !). En termes de mise en page, c'est très sommaire, mais l'essentiel est là. Par ailleurs, les fonctionnalités de Tomboy peuvent s'étendre à l'aide de plugins. On regrette tout de même de ne pas pouvoir ajouter d'image / audio / vidéo à une note.

Alternatives : Dans l'esprit Wiki de Tomboy, on retrouve Zim (multiplateforme, Python), un outil un peu plus complet que Tomboy (gestion des listes de tâches et des images), mais aussi moins facile d'accès. Il existe également Keepnote, aux fonctionnalités complètes mais à l'interface plus imposante et moins directe. Ces deux alternatives sont distribuées sous licence GPL.

Balsamiq Mockup [Fig.5]

Pour la réalisation de maquettes, il y a ceux qui utilisent exclusivement le crayon et le

papier. Pour les autres il y a Balsamiq Mockup (multiplateforme, AIR). Avec cet outil, la réalisation de maquettes est rapide et simple d'approche. Par contre, pour profiter de toutes les fonctionnalités, il faut s'acquiescer d'une licence. Si cet achat est tout à fait envisageable pour des projets volumineux, la question se pose par contre pour un développeur freelance ou un projet de faible envergure.

Alternatives : Il existe une alternative libre (licence GPL), nommée Pencil Project. Il lui manque cependant cette touche particulière qui rend Balsamiq Mockup si simple et agréable à utiliser. Par contre Pencil Project ne se cantonne pas au maquettage d'applications et propose des formes très variées permettant de réaliser des diagrammes en tout genre. Par ailleurs, Pencil Project peut s'installer en tant que simple extension Firefox.

> Débogage et tests

En règle générale, les outils de debug les plus pratiques à utiliser dans le cadre du développement sont ceux de l'IDE retenu. Dans le cas du développement Web cependant, des outils complémentaires peuvent se révéler d'une grande aide. Il est par ailleurs nécessaire de tester l'application web sur les principaux navigateurs du marché, il est donc important d'avoir une panoplie des dernières versions des navigateurs les plus courants, voire des anciennes versions, notamment pour IE qui a longtemps été le plus répandu... et le moins respectueux des standards pour ne pas simplifier les choses.

Firebug [Fig.6]

Firebug (licence BSD) est le premier et le plus connu des outils de débogage intégrés au navigateur. Il s'installe sous forme d'une

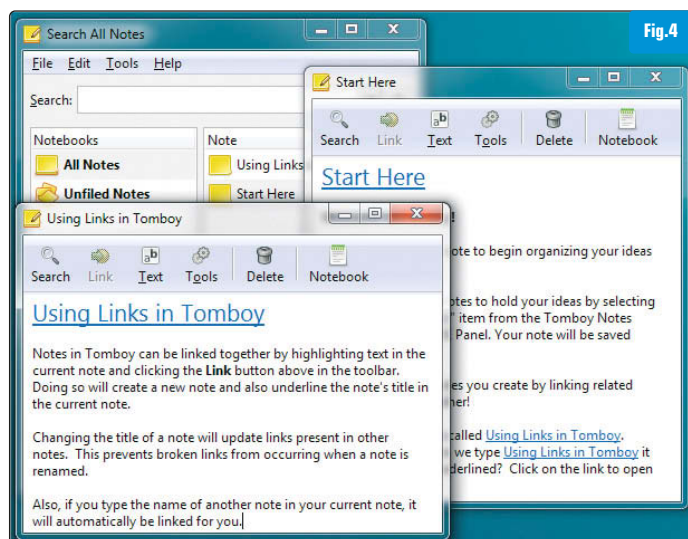


Fig.4

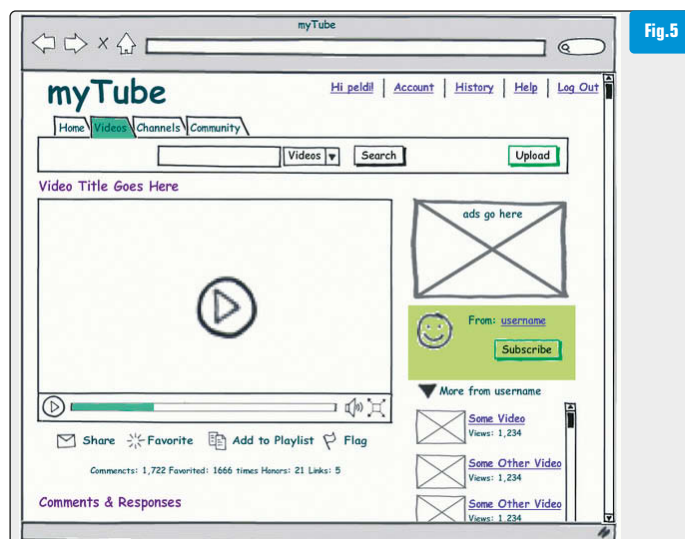


Fig.5

extension de Firefox. Ses principales fonctionnalités sont

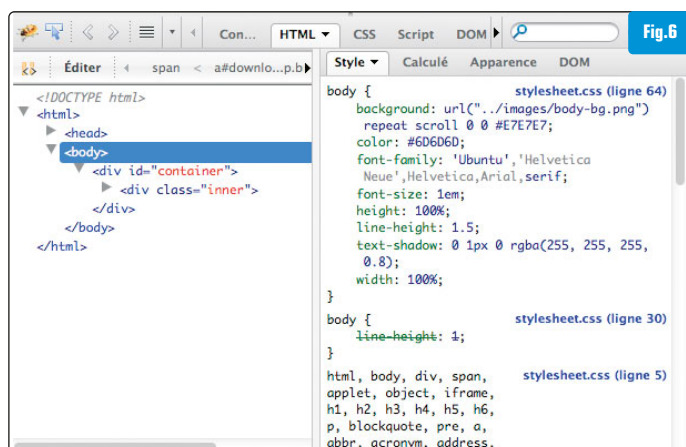
- l'affichage du code HTML associé à un élément de la page avec possibilité de le modifier à la volée
- la visualisation et la modification à la volée des règles CSS associées à un élément sélectionné
- le débogage des scripts Javascript de la page

Alternatives : Les navigateurs récents sont maintenant souvent fournis avec des outils de développeurs intégrés reprenant les mêmes fonctionnalités que leur précurseur Firebug, parfois de façon encore plus complète à l'instar des outils du développeur intégrés à Google Chrome qui permettent, entre autres, un profilage plus poussé, une gestion du javascript plus complète, ... Petite particularité pour IE : des outils de développeurs sont intégrés depuis la version 8 du navigateur, mais pour les retrouver sous IE7 il faut installer IE Developer Toolbar.

IETester [Fig.7]

IETester est un outil permettant aux développeurs de tester leurs applications avec le moteur de rendu ainsi que le moteur Javascript des anciennes versions d'IE sur des versions de Windows qui ne permettent plus de les installer. Malheureusement, IETester n'offre pas un rendu 100% identique au moteur d'origine exécuté sous XP. Il est donc nécessaire de tester l'application en dernier lieu sur une version originale d'Internet Explorer, sur une machine équipée d'XP ou via une machine virtuelle.

Alternatives : Sous Windows, il y a quelques outils avec les mêmes fonctionnalités et aux limitations équivalentes, parmi lesquels MultipleIE et IECollection. Il n'existe pas de réelle alternative à IETester sous OS X ni Linux. Le plus simple dans ce cas est d'avoir recours à la virtualisation.



VirtualBox [Fig.8]

VirtualBox est un logiciel de virtualisation OpenSource (licence GPL) et multiplateforme qui permet de virtualiser les OS les plus répandus. Son utilisation est simple et ses options de configuration complètes.

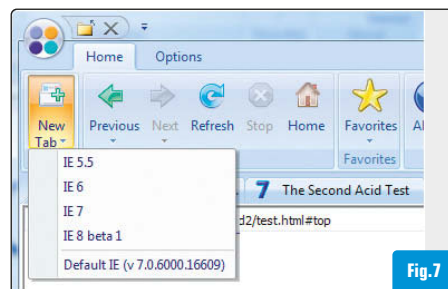
Alternatives : Si la machine à émuler est un Windows XP et si la machine hôte est sous Windows 7 pro, Microsoft propose gratuitement un mode de compatibilité qui n'est autre qu'un Windows XP émulé (par Virtual PC). Par ailleurs, il existe différents outils de virtualisation tels que Virtual PC (Microsoft) ou encore VMware avec lesquels il faut bien évidemment posséder une licence valide du système.

> Le déploiement et la maintenance

Pour intervenir directement sur un serveur, y déployer un correctif, y copier des fichiers, consulter les logs... voici une petite sélection d'applications très utiles.

Filezilla [Fig.9]

L'échange de fichiers avec le serveur distant peut se faire de plusieurs façons, la plus répandue étant certainement grâce au protocole FTP, ou mieux, un équivalent sécurisé. Filezilla permet un échange simple de fichiers avec un serveur FTP, SFTP ou FTPS. Rapide, complet, libre (sous licence GPL) et multiplateforme, que demander de plus ? Peut-être une interface un peu moins « fouillie », ce qui se ressent surtout sur les



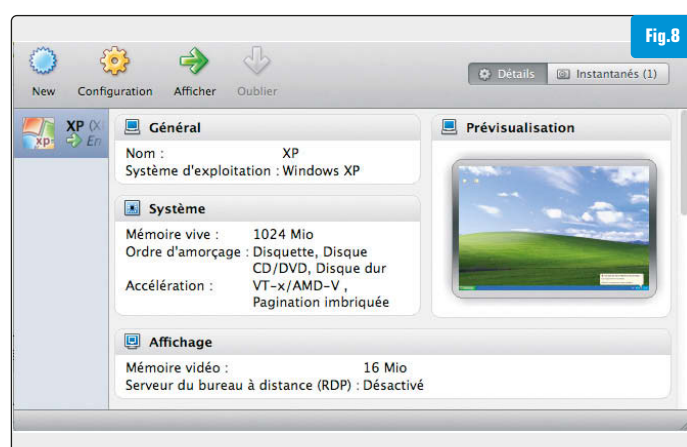
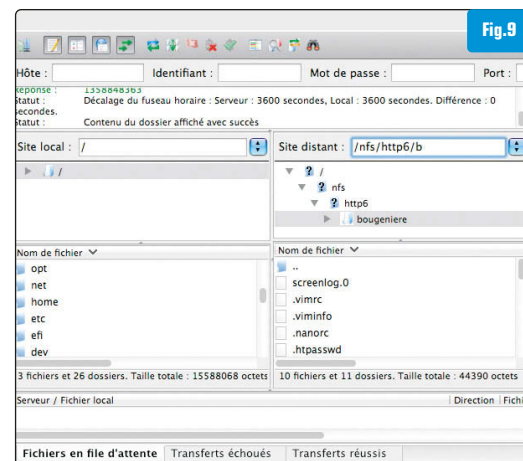
petits écrans, situation heureusement de moins en moins fréquente.

Alternatives : Les alternatives sont nombreuses pour ce type d'outils. S'il fallait citer un autre client, ce serait certainement Cyber Duck (Windows, OS X) sous licence GPL, pour son interface épurée qui sort un peu du moule. Notons que les versions récentes des différents OS proposent des clients intégrés qui permettent de gérer les connexions FTP (ou autres connexions réseaux) comme s'il s'agissait de répertoires locaux.

Putty [Fig.10]

Lorsque l'on travaille sous OS X ou Linux, l'accès en SSH à un serveur Linux ne pose pas de problème particulier, un client SSH en ligne de commandes étant disponible par défaut. Sous Windows, il faut en passer par l'installation d'un client. Un des plus répandus, grâce à sa gratuité (licence MIT) et à ses options de configuration, est Putty.

Alternatives : Sous Linux, si on cherche plus complet que la console par défaut, Remmina (licence GPL) propose de centraliser au sein d'une même interface tous les clients d'accès à distance dont vous avez besoin, dont SSH (mais aussi VNC, Terminal Services, ...)



GNU Screen [Fig.11]

GNU Screen est ce que l'on appelle un multiplexeur de terminal. Ce programme sous licence GPL s'installe sur un serveur Linux et permet de lancer plusieurs terminaux "virtuels" au sein d'un unique terminal (et donc, le cas échéant, d'une unique connexion SSH lors d'un accès distant). Il permet de nommer les différents terminaux virtuels, d'afficher différents terminaux en même temps (avec séparation horizontale), mais surtout, le processus de GNU Screen peut être détaché du terminal dans lequel il a été ouvert d'une simple combinaison de touches, les scripts en cours continuant leur exécution normalement. Le processus peut ensuite être ré-attaché à un autre terminal à tout moment. Ceci simplifie donc le lancement de scripts sur des serveurs distants, en éliminant tout risque d'interruption due à la connexion SSH.

Alternatives : Le concurrent principal de GNU Screen est Tmux qui offre les mêmes fonctionnalités mais est, quant à lui, distribué sous licence BSD et représente d'une certaine façon l'avenir. En effet, GNU Screen n'est plus maintenu, et de nouvelles fonctionnalités ont très peu de chances de voir le jour.

UltraVNC [Fig.12]

Lorsque la ligne de commande n'est pas suffisante, on peut utiliser le protocole VNC pour obtenir le bureau graphique d'une machine à distance. UltraVNC est un client ainsi qu'un serveur VNC pour Windows. Il embarque par ailleurs des fonctionnalités complémentaires comme le transfert de fichiers entre le client et le serveur.

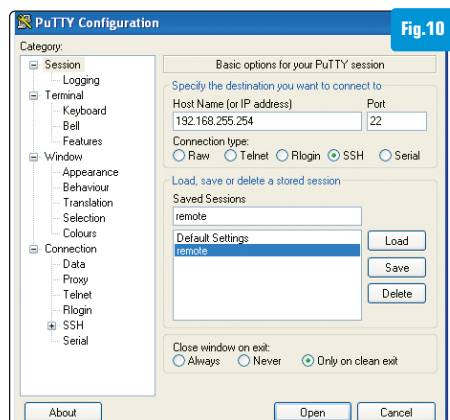


Fig.10

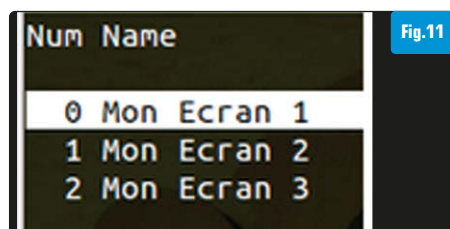


Fig.11

Alternatives : Il existe sous Windows de nombreuses implémentations de VNC répondant peu ou prou aux mêmes besoins qu'UltraVNC. Parmi les plus répandues, on retrouve TightVNC ou RealVNC. OS X et de nombreuses distributions Linux intègrent par défaut un client et un serveur VNC. Par ailleurs, pour se connecter à un serveur Windows à distance, on peut utiliser un client compatible avec le protocole d'accès distant de Microsoft : Terminal services. Le client est intégré par défaut sous Windows, Microsoft en propose un officiel pour Mac, et Remmina remplit très bien ce rôle sous Linux. Enfin, il existe des outils en ligne, comme TeamViewer ou LogMeIn. L'avantage de ces deux outils est la mise en place simplifiée : le client et le serveur communiquent via un tiers distant (un serveur de LogMeIn ou de TeamViewer) et se comportent donc tous deux comme simple client du même service. Ceci permet de s'affranchir de la configuration des pare-feu et des services NAT sur le réseau. Il faut bien sûr dans ce cas accepter que les échanges passent par un tiers sur lequel on n'a pas la main, aussi sécurisé soit-il.

> Divers

Dans cette section sont rassemblés pêle-mêle quelques logiciels aux fonctionnalités très hétéroclites.

PrtScr [Fig.13]

Un petit croquis étant bien souvent plus efficace qu'un long discours, de la documentation ou des emails sont souvent avantageusement accompagnés de captures d'écran explicatives. PrtScr permet de compléter efficace-

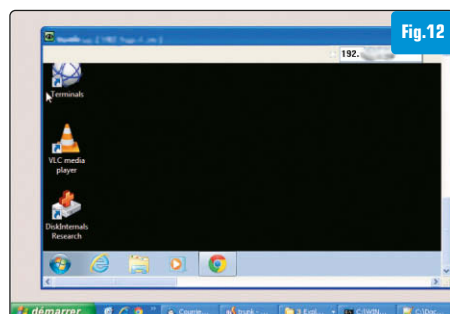


Fig.12

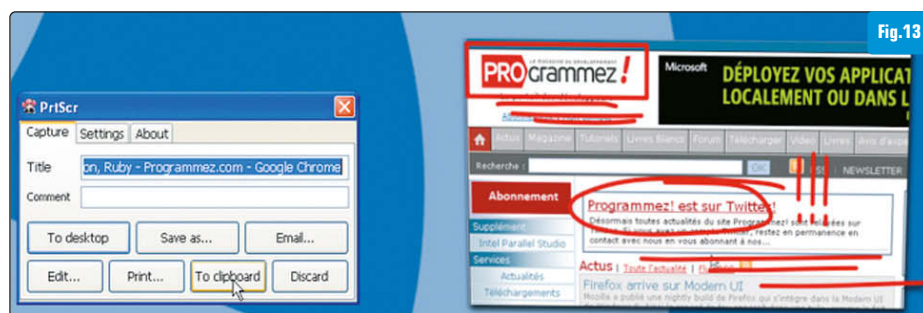


Fig.13

ment la capture d'écran disponible par défaut sous Windows. Une fois la touche Impr Ecran enfoncée, l'image peut directement être annotée et recadrée. Le résultat peut alors être copié dans le presse papier, enregistré, envoyé par mail ou encore ouvert directement dans un éditeur d'images. On regrette cependant de ne pas pouvoir choisir la couleur des annotations ou simplement ajouter du texte saisi au clavier.

Alternatives : Sous Windows, il existe d'autres outils gratuits plus complets, comme Greenshot (licence GPL) ou FastStone Capture (gratuit jusqu'en version 5.3, payant ensuite). Pour des annotations poussées ils sont plus complets que PrtScr, mais moins directs (il faut ouvrir la capture dans l'éditeur intégré pour l'annoter). Sous OS X, il existe Skitch, disponible gratuitement. Il est moins léger et direct, mais est très complet. Sous Linux, citons par exemple Shutter qui s'intègre très bien à Gnome.

Tortoise SVN [Fig.14]

Tortoise SVN est un client SVN sous licence GPL qui s'intègre à l'explorateur de fichiers de Windows et permet de gérer les dépôts depuis le menu contextuel. Des icônes spécifiques permettent de visualiser en un clin d'oeil l'état du répertoire par rapport au dépôt.

Alternative : C'est à notre connaissance l'outil le plus avancé dans son genre, il n'existe pas de véritable alternative pour Windows, OS X ou Linux s'intégrant aussi pleinement au navigateur de fichiers.

HeidiSQL [Fig.15]

HeidiSQL est une alternative légère à MySQL Workbench pour Windows distribuée sous licence GPL. Il permet également de remplacer avantageusement phpMyAdmin car il n'est pas soumis aux contraintes des applications Web. Cerise sur le gâteau, HeidiSQL est compatible avec SQL Server de Microsoft, il est donc aussi une alternative légère à SQL Server Management Studio. Il ne dispose en revanche d'aucune fonctionnalité de conception, comme peuvent en présenter

Mysql Wordbench et SQL Server Management Studio.

Alternatives : Dans le même esprit de légèreté, il existe Sequel Pro pour OS X (licence MIT), ou encore Toad for MySQL pour Windows. Une alternative pour Linux : DBeaver (multiplateforme, Java).

WinMerge [Fig.16]

Souvent utilisé de pair avec un outil de gestion de versions, l'outil de comparaison / fusion qu'est WinMerge (licence GPL) peut également être utilisé seul pour comparer des fichiers, voire même des répertoires, et les fusionner.

Alternatives : Meld (licence GPL également) sous OS X, Linux ou Windows permet de réaliser les mêmes tâches.

Reflector [Fig.17]

En .Net ou Java Lorsque les sources d'une librairie ne sont pas disponibles, on peut utiliser des outils dits de décompilation pour accéder à ses entrailles. Pour .Net, la référence dans le domaine est très certainement Reflector, mais qui, malheureusement pour les petites bourses, est devenu payant avec les dernières versions.

Alternatives : Avec les mêmes fonctionnalités, mais gratuit, il existe DotPeek proposé par le prolifique JetBrains (Éditeur connu pour son IDE Java, IntelliJ). Mocha est un équivalent pour le monde Java.

Quelques applications WEB

Il existe également une quantité grandissante d'outils disponibles en ligne, pour répondre à des besoins très variés. Sans pour autant les détailler ici, voici une petite liste qui pourrait être utile aux développeurs :

- <http://jsfiddle.net/>, <http://jsbin.com/>, <http://cssdesk.com/>, <http://codepen.io/>, <http://tinkerbin.com/>, <http://liveweave.com/> ... pour le prototypage rapide et le partage de bouts de code html / CSS / javascript
- <http://www.generatedata.com> pour la génération de données (pour remplir des bases de données de tests par exemple)
- <http://mergely.com/>, outil de fusion en ligne
- <http://spritepad.wearekiss.com/>, outil de création de sprite CSS
- <https://cacao.com>, <https://moqups.com/>, <http://mocknow.com/>, outils de création de diagramme et mockup

- <http://www.colorblender.com/>, <http://colorscheme-designer.com/>, outil d'aide à la génération du code couleur d'un site web

> Conclusion

Nous pourrions continuer la liste de ces outils indéfiniment, ceci étant, vous en avez à présent une sélection assez large qui devrait permettre à tout développeur de pouvoir enrichir son environnement de développement. Les applications mobiles (pour tablettes notamment) n'ont pas été présentées ici, car le marché est encore majoritairement destiné aux loisirs (jeux, multimédia). Mais de plus en plus souvent, les tablettes sont utilisées dans le milieu professionnel. On devrait donc y retrouver ce type d'outils en plus grand nombre, et surtout de façon plus adaptée à l'usage tactile. Les applications web et les offres de Cloud ont également leur carte à jouer pour conquérir ce type de support.



Gilles Bougenière
Consultant Osaxis

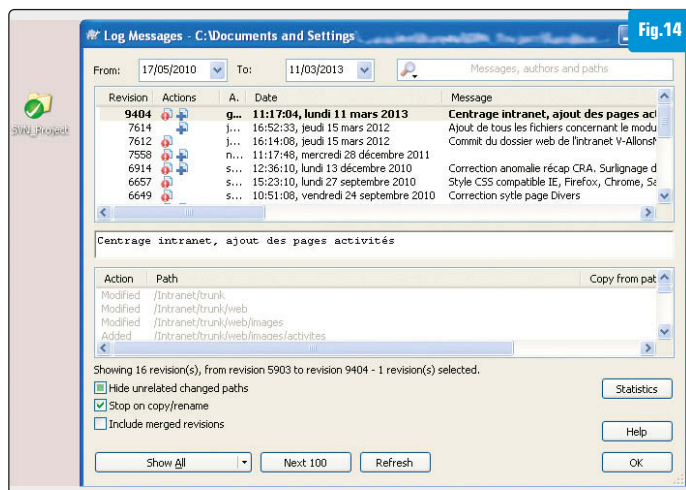


Fig.14

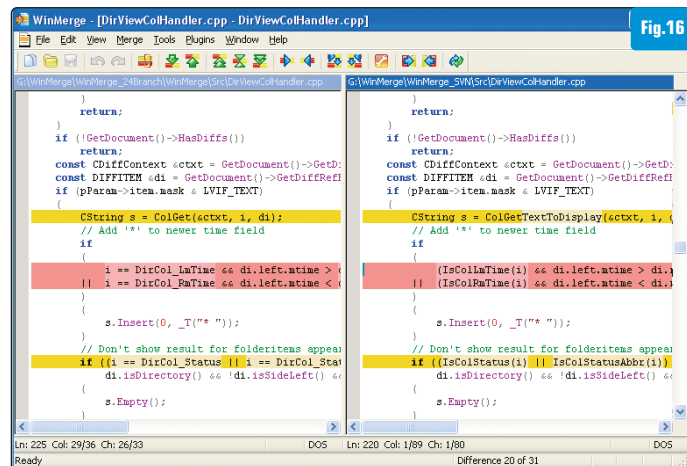


Fig.16

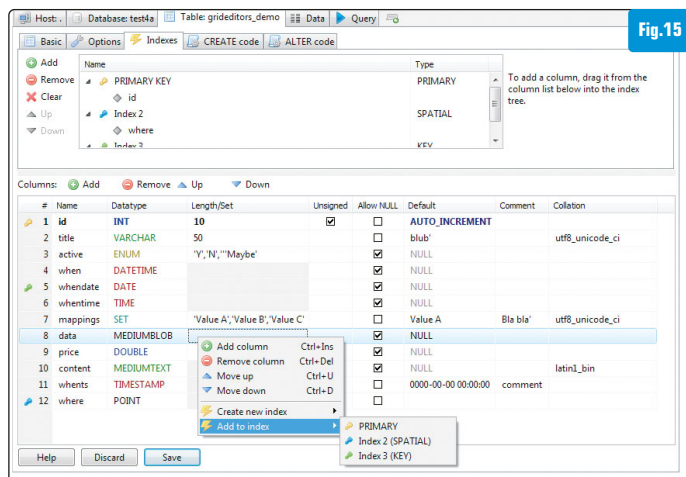


Fig.15

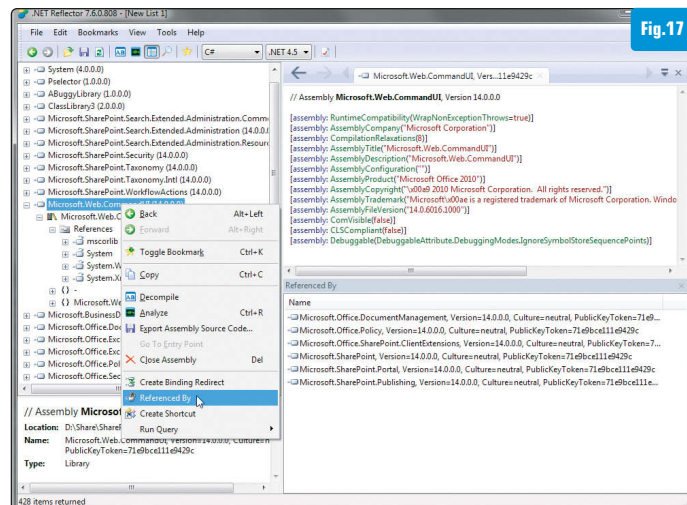


Fig.17

OFFREZ-VOUS

UN ABONNEMENT jusqu'à -50%

Code, gestion de projets, développement web, mobilité, Programmez ! est à la fois votre outil pratique, des articles de code par les meilleurs experts et votre veille technologique.

1 Abonnement 1 an au magazine

49 €* (au lieu de 65,45 €, prix au numéro)

2 Abonnement Intégral : 1 an au magazine + Archives Internet et PDF 59 €*

3 Abonnement 2 ans au magazine

78 €* (au lieu de 130,90 €, prix au numéro)

4 Abonnement intégral 2 ans au magazine + Archives Internet et PDF 88 €*

5 Abonnement Etudiant 1 an au magazine 39 €* + Archives Internet et PDF 49 €*



(*) Tarifs France métropolitaine

Toutes les offres en ligne : www.programmez.com

Abonnez-vous à partir de 3,80 € seulement par mois

Oui, je m'abonne

à retourner avec votre règlement à

Programmez, 17 route des Boulangers 78926 Yvelines cedex 9

- ☐ **Abonnement 1 an au magazine** : 49 €* (au lieu de 65,45 €, prix au numéro)
- ☐ **Abonnement Intégral : 1 an au magazine + archives Internet et PDF** : 59 €* (au lieu de 65,45 €, prix au numéro)
- ☐ **Abonnement 2 ans au magazine** : 78 €* (au lieu de 130,90 €, prix au numéro)
- ☐ **Abonnement intégral 2 ans au magazine + archives Internet et PDF** : 88 €*
- ☐ **Abonnement Etudiant : 1 an au magazine** : 39 €* (joindre copie carte étudiant)
- ☐ **Abonnement Etudiant : 1 an au magazine + archives Internet** : 49 €* (joindre copie carte étudiant)

(*) Tarifs France métropolitaine

☐ M. ☐ Mme ☐ Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

Tél : _____

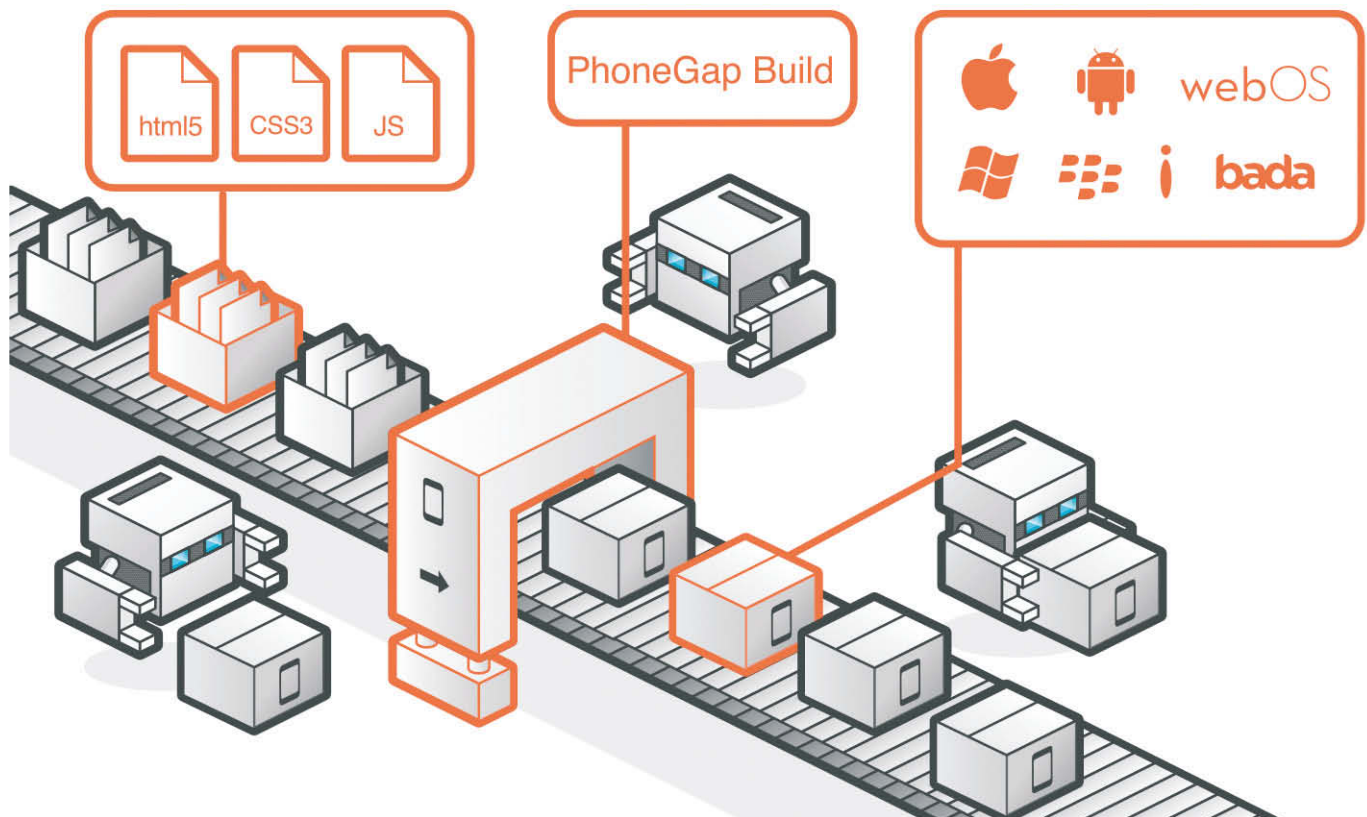
E-mail : _____ @ _____

☐ Je joins mon règlement par chèque à l'ordre de Programmez !

☐ Je souhaite régler à réception de facture

(Attention, e-mail indispensable pour les archives sur internet)

Développer pour les tablettes !



Le marché de la tablette tactile est en plein boom et certains utilisateurs remplacent même leur ordinateur par des tablettes, particulièrement pour les Netbooks. A tel point que l'on parle même de l'Après-PC. Disons que pour un grand nombre d'utilisateurs, la tablette est un terminal complémentaire à son PC / Mac. Un constat cependant : le marché du PC s'écroule (- 7,6 % prévu en 2013, Gartner), mais les tablettes s'envolent ! (+70 %). En 2013, plus de 197 millions de tablettes devraient être vendues (Gartner), les analystes d'IDC parlent de 191 millions.

Si aujourd'hui les iPad dominent toujours le marché, les tablettes Android (du fait de la multiplication des offres) se vendront davantage, mais aucun constructeur Android ne peut encore rivaliser avec l'iPad (seul modèle iOS) d'Apple qui s'est encore vendu à 19,4 millions d'unités en 3 mois... Et l'AppStore d'Apple reste la boutique d'applications la plus rentable. Bref, iOS + Android, c'est comme sur le smartphone, 90 % du marché des tablettes !

Bon, que reste-t-il derrière ? A peine 10 %, bref, des miettes. La playbook de BlackBerry étant hors-jeu, les tablettes Windows 8 (rajoutons Windows RT) existent réellement depuis 6 mois, notamment avec les modèles Surface et Surface Pro. Pour les premiers mois de 2013, les tablettes Windows 8 se sont écoulées à environ 3 millions d'exemplaires... IDC est d'ailleurs prudent sur les perspectives des tablettes Windows. Il estime qu'ils pèseront -10 % en 2017 (dont - 3 % pour Windows RT). Il faut dire que le couple Android / iOS sera toujours à 90 % !

Dans ce dossier, nous allons bien entendu évoquer le développement spécifique pour chaque environnement mais aussi aborder les aspects économiques, la fragmentation du marché et nous demander si l'approche native est plus pertinente que l'approche HTML 5 / CSS / JavaScript.

Bon tactile !

François Tonic



HTML vs Code natif vs Hybride

L'utilisation des smartphones et des tablettes se concrétise, ce n'est plus un secret pour nous les développeurs. Cela dit, certaines questions voient le jour et forcément des choix s'imposent. L'un des premiers étant architectural : dans quel langage allons-nous développer notre application ?

Faut-il se poser d'autres questions, notamment en ce qui concerne les performances, l'interface et l'expérience à apporter aux utilisateurs finaux ? S'ensuit la nécessité d'accéder aux API, les coûts de développement ainsi que la réutilisation de notre code. J'ai eu la chance de participer – au cours d'un stage ou à titre personnel – au développement de plusieurs applications sur ces terminaux. Parmi ces projets, certains m'ont demandé (ou à mon équipe) de répondre à cette problématique.

QUELQUES RAPPELS

> WebApps

Les WebApps mobiles sont des sites mobiles accessibles depuis un navigateur Internet installé dans le téléphone. Elles n'apparaissent pas dans les kiosques d'applications mais ont le mérite d'être utilisables dans tous les OS mobiles.

> Applications hybrides

Ce sont simplement des WebApps intégrées dans une application native pour pouvoir apparaître dans les kiosques et qui peuvent interagir avec le système hôte.

> Applications natives

Les applications natives sont développées dans un langage de programmation plus ou moins spécifique à la plateforme. Objective-C pour iOS, Java pour Android, C++/Silverlight C#/VB.net, etc. pour Windows 8 / Windows Phone 8.

LES QUESTIONS À SE POSER

Décider entre développer une application native ou une application « cross-platform » HTML est un choix technique essentiel qui aura des répercussions tout au long du développement du projet.

> Performances

Une application lente peut-être une raison de son abandon par l'utilisateur. A moins qu'elle soit vraiment de première nécessité, clin d'œil à l'application « Mes comptes » de BNP

PARIBAS sur Windows Phone en tout cas. Développer en HTML comprend un risque majeur pour les performances de l'application qui varient en fonction de la connectivité, du navigateur, de sa version et de son moteur de rendu. Sans oublier tous les éléments médias : téléchargement, stockage, ... Ces éléments engendrent une perte de performances.

> Évènements OnTouch !

Un point essentiel, bien souvent oublié, c'est que dans une App mobile, il n'est pas question d'utiliser *onclick*, *onmouseover* et *onmouseout* !

Pour gérer des événements équivalents dans les navigateurs mobiles il faut utiliser *ontouchstart*, *ontouchend*, *ontouchmove*. Pour ceux qui ne le savent pas, *onclick* correspond à *ontouchstart* + *ontouchend*. Inutile de vous dire que le temps de réponse est 5 fois supérieur... 500 vs. 110 millièmes de secondes...

> Stockage

Dans le lot des nouveautés de HTML5, le Web Storage permet de stocker des données dans une liste clé/valeur jusqu'à 10 Mo ! Si votre application en a l'utilité, n'hésitez pas à vous en servir pour sauvegarder des données personnelles de l'utilisateur, par exemple. Pour en savoir plus, <http://de-bray-jerome.developpez.com/articles/comprendre-le-storage-en-html5/>.

NB : Tous les navigateurs Internet n'implémentent peut-être pas encore cette fonction.

Dans une application mobile native, les éditeurs des OS nous ont fort heureusement ouvert les portes pour l'enregistrement de données via les SDK : les classes *IOStorage* pour iOS, *Isolate Storage* pour Windows 8 et Windows Phone et les différentes méthodes Android nous permettent d'y accéder et de pérenniser des données.

> Moteur de rendu

Les performances d'une WebApp dépendent du moteur de rendu du navigateur fourni

avec l'OS de votre smartphone ou de votre tablette, en l'occurrence Safari pour iOS, Internet Explorer pour Windows Phone et/ou d'outils comme Phone Gap.

> Interface et Expérience Utilisateur

La quasi-totalité des marques affichent leur identité par un jeu de couleurs, de graphismes et autres signatures distinctes pour se démarquer ou ressembler à une autre. L'expérience utilisateur est censée mettre à l'aise l'utilisateur, qui s'est habitué à sa tablette ou à son smartphone et retrouve les mêmes gestuelles dans votre application ainsi que dans les autres applications qu'il s'est procurées.

> Design + WebApp

Si vous décidez de développer une WebApp, vous offrez le même affichage pour toutes les plateformes, à quelques détails près. Vous n'offrez donc pas l'expérience attendue par l'utilisateur final ni l'interface voulue par la philosophie du système d'exploitation mobile (gestuelles, taille de polices, quantité d'informations, disposition des éléments, ...).

> Design + Plateforme dédiée

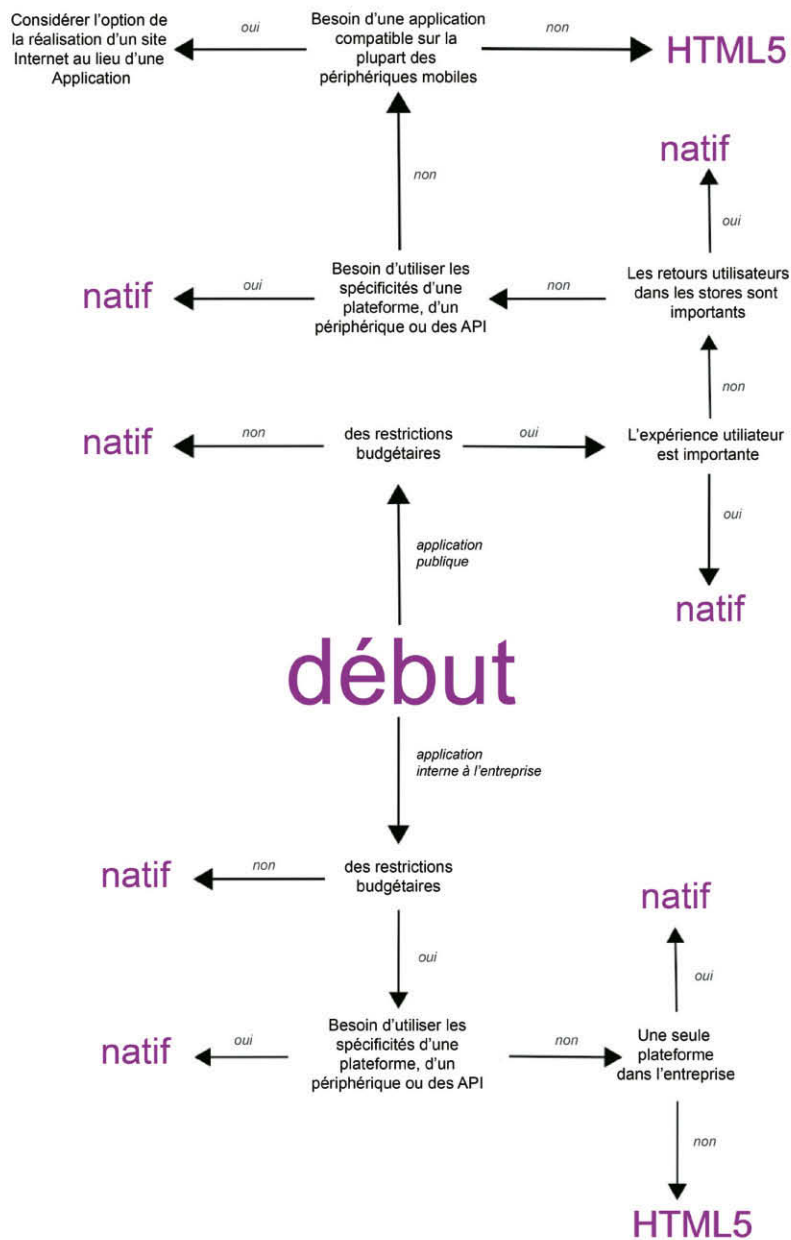
Dans le cadre d'une application développée avec un langage natif, l'interface respecte dans la plupart des cas les guidelines fournies par les éditeurs des systèmes, Modern UI pour Windows 8, skeuomorphisme pour iOS par exemple.

En suivant les guidelines, vos utilisateurs se retrouveront aisément dans l'application : la navigation sera similaire aux autres applications (y compris celles du système) de même pour les éventuels menus de l'application, les informations seront plus ou moins hiérarchisées ; bref, votre application appréhendera la philosophie de la plateforme !

> L'accès aux dernières fonctionnalités des périphériques et/ou des API

Sur la plupart des plateformes, environ 3/4 des fonctionnalités des API sont accessibles

Choisir entre une application native ou HTML5



par les WebApps : géolocalisation, stockage, réseau, ... Pour combler le quart manquant (contacts, géolocalisation complète, moteurs de rendu 3D avancé, ...), il existe différentes solutions : injecter du JavaScript, créer une couche pour communiquer entre la WebApp et le navigateur qui exécute celle-ci, utiliser l'existant : Phone Gap et sans doute bien d'autres solutions !

> Coûts de développement

Développement plus rapide et multiplateforme oblige, les WebApps et hybrides rempor-

tent haut la main la manche des coûts. Ayant besoin de passer du temps pour chaque plateforme et parfois ayant besoin de connaissances plus techniques, les développeurs d'applications natives peuvent coûter plus cher, en nombre au moins.

> Réutilisation et pérennité de code

Versionning, commentaires ordonnés, code structuré, les mêmes conseils s'appliquent pour le développement d'applications HTML et natives.

Cela dit, en plus des IDE très complets qu'offrent les éditeurs (Visual Studio, Xcode, ...), ces derniers fournissent à leurs développeurs des outils plus ou moins poussés et intégrés pour permettre de collaborer sur un projet, de gérer aisément les différentes versions, etc.

> Publication dans les kiosques

Si vous avez une WebApp, uniquement accessible sur Internet, il faudra inviter l'utilisateur à taper son URL directement dans le navigateur ou à vous trouver via son moteur de recherche préféré.

Pour que l'application soit publiée dans les kiosques dédiés aux différentes plateformes, il faut les soumettre en donnant un maximum d'informations et attendre leur éventuelle validation. Les utilisateurs seront alors libres de télécharger votre chef d'œuvre sur leurs petits bijoux.

LES PROJETS AUXQUELS J'AI PRIS PART

Développeur passionné par les nouvelles technologies, j'ai été amené à développer divers projets notamment sur smartphone et tablettes. Et pour chaque projet, il fut nécessaire de se poser ces quelques questions.

> Les Apps astro-privée.com, pour Circulys

Site Internet de rencontres dont les calculs sont basés sur l'astrologie, des schémas d'application WinRT et Windows Phone ont vu le jour dans le but d'une visibilité élargie sur différentes plateformes. Le site étant développé en PHP, il était tentant de s'amuser avec du HTML 5, des Media Queries pour adapter au tactile et de l'intégrer dans un projet d'application Windows 8 HTML et dans un Web Browser Windows Phone.

Mais ergonomiquement, ce n'était pas très « Modern UI friendly »... Or moi-même et la plupart des utilisateurs considèrent l'expérience utilisateur comme un point essentiel d'une application. N'ayant rien d'existant, mon choix s'est porté sur le développement d'une application native respectant les guidelines Microsoft en ce qui concerne le design. À ce moment-là, j'ai même pu avoir une plateforme commune Windows Phone 7, Windows Phone 8 et WinRT. Il suffisait juste de récupérer les données dans un projet et de les afficher correctement dans les 3 projets correspondant aux interfaces utilisateur des plateformes Microsoft.



> GaySpot, pour Circulys

Afficher des points, des lieux de rencontre pour être un peu plus explicite, sur une carte Windows Phone 7 & 8. C'était le défi. Sur Windows Phone 7, la carte Google fonctionne sur la plateforme mobile Microsoft. Suite à quelques difficultés techniques, mon choix de garder la page Internet existante et de l'intégrer dans une application s'est renforcée et a été adoptée. L'expérience utilisateur est similaire à la version Internet de l'application, propre à l'identité de GaySpot.

Pour Windows Phone 8, les possibilités offertes par les nouvelles API était bien trop intéressantes pour les laisser de côté. Entre autres, les cartes offline, les tuiles, etc. Ergonomiquement parlant, l'application est full Modern UI et est agréable à utiliser.

> Moulin de la forge au Hackathon de décembre 2012

En décembre 2012, 100 développeurs de la communauté Windows Phone/Windows 8 ont été invités à se retrouver, invités par Microsoft. Dans l'un des concours, deux développeurs passionnés et moi-même avons relevé le défi de développer rapidement une application qui pouvait être utilisée par tous les visiteurs du moulin de la forge. Ayant tous les trois des compétences en Silverlight C# et l'application étant à la destination des possesseurs de Windows Phone, notre choix s'est vite porté sur du natif.

> Gossip

Puisqu'il s'agit d'un projet personnel avec un collaborateur, c'est entièrement à moi de faire les choix techniques. Mais nous voulons la porter sur les différentes plateformes existantes pour que les possesseurs de périphériques Apple, Google, Microsoft et BlackBerry puissent utiliser l'application.

Petit bémol : on n'a pas les compétences techniques sur iOS. Budget restreint, on a fait le choix de faire du natif sur Windows Phone et WinRT, puis des portages sur Android et iOS à l'aide de Xamarin. Le développement continue sa route et l'application sera disponible dès que les développements sur toutes les plateformes seront terminés ! Pour ceux qui ne connaissent pas, Xamarin est en même temps une extension Visual Studio et un IDE à part entière qui permet de développer des applications iOS et Android en C# tout en ayant les mêmes performances qu'une application développée avec les outils recommandés par Apple et Google, soit XCode et Eclipse + ses SDK.

> Une application interne

La direction d'un grand acteur du sanitaire dans une région voisine de celle où j'habite veut pouvoir consulter des informations internes régulièrement et aisément sur ses tablettes et autres téléphones personnels. Le parc de ces périphériques étant très divers (iOS, Android, WinRt, Windows Phone, ...) et le budget étant limité, c'est en HTML5 que l'application a été codée, reprenant les résultats envoyés par un serveur externe sur une page Internet n'affichant que ce qui est demandé : chiffres d'affaires, bénéfices, triés par famille de produits, fabricant, etc. Merci PhoneGap !

PhoneGap ? Pour ceux qui ne connaissent pas, le projet permet de développer rapidement et facilement des applications en utilisant HTML, CSS et JavaScript et de les rendre compatibles sur des tablettes Windows 8, iPad et autres tablettes Android.

> Strasbourg+, pour les DevKings 2013

Qui n'a jamais cherché une application d'une ville dans laquelle on prévoit de passer quelques jours, semaines ?

Où chercher des informations telles que des horaires de bus, de tram, l'emplacement d'une médiathèque ou d'un office de tourisme ? La ville de Strasbourg s'agrandit peu à peu, les moyens de transports évoluent rapidement et la façon d'informer les strasbourgeois aussi (NFC, Flash Code aux arrêts de bus et aux lieux importants de la ville).

Sur Android, une application moins complète (simples horaires et itinéraires) a vu le jour il y a quelque temps, pareil sur iPhone.

Il ne manque plus que l'environnement Windows ! Les heureux possesseurs d'un Windows Phone ou d'une tablette Windows 8 pourront bientôt avoir tous ces services en une seule application !

Code partagé, maintenance aisée, destinée

uniquement aux possesseurs de Windows, le choix pour du natif s'est imposé naturellement.

> Catalogue de formations, pour Happly

Chez Happly, SSII et centre de formation basé à Strasbourg et Mulhouse, on m'a demandé de porter une application Windows 8 existante sur Windows Phone 7 et 8. L'application pour Windows 8 avait été développée en natif C#. Pour continuer dans cette lignée, l'application Windows Phone a repris un maximum de code, en ajoutant des touches « Windows Phone friendly », le partage d'une session ou d'une formation, l'envoi de mails, l'épinglage dans l'écran d'accueil, etc.

CONCLUSION

C'est en fonction des facteurs comme le budget, l'expérience utilisateur, les API nécessaires, etc. que le choix final se fait quant au développement d'une App native ou d'une WebApp.

En prenant un peu de recul, en me basant sur la théorie et ma jeune expérience, le schéma de la page 38 peut être utilisé comme guide pour faire le bon choix. (Source originale : <http://dontcodetired.com/blog/post/The-Architects-Guide-to-Choosing-Between-HTML5-or-Native-Mobile-Apps.aspx>).

Quitte à développer, compiler pour chaque plateforme et avoir un projet par plateforme dans une solution, des outils de développement plus ou moins coûteux.



Andrés Talavera
Développeur passionné
Microsoft
Microsoft Student
Partner - Communautés
Microsoft
about.me/im.cresus

Liens annexes

<http://www.brianmadden.com/blogs/brianmadden/archive/2012/04/16/how-to-argue-html-vs-native-touch-vs-keyboard-and-tablet-vs-pc-who-will-win-amp-quot-desktops-versus-tablets-amp-quot.aspx>
<http://www.localapps.co/mobile-application-development-html-5-vs-native/>
<http://www.scriptol.com/mobile/native-vs-html5.php>
<http://venturebeat.com/2011/07/08/hybrid-mobile-apps-take-off-as-html5-vs-native-debate-continues/>
<http://venturebeat.com/2011/04/07/how-html5-will-kill-the-native-app/>
<http://dontcodetired.com/blog/post/The-Architects-Guide-to-Choosing-Between-HTML5-or-Native-Mobile-Apps.aspx>
<http://www.html5rocks.com/en/mobile/native-debate/>

Développer des applications Android pour tablettes

Android a acquis sa domination en profitant du formidable essor des smartphones. Dans un marché des appareils connectés en perpétuelle évolution, la tendance est désormais aux écrans larges voir extra-larges et les tablettes s'annoncent comme le prochain segment de marché à conquérir. Pour faire face à ces enjeux, l'OS a connu de profondes mutations depuis sa création pour aboutir fin 2011 à une version 4.0 permettant aux développeurs d'exploiter au mieux le potentiel de ces appareils. Tour d'horizon des points essentiels à prendre en compte pour développer des applications Android pour tablettes.

Le marché des smartphones a connu une croissance exponentielle ces dernières années et aujourd'hui, la tendance est de proposer des écrans toujours plus grands. Dans ce contexte, le marché des tablettes est en phase ascendante et son avenir s'annonce radieux. Si selon certaines études sur la plateforme Android, la proportion de tablettes reste assez faible, la situation est diamétralement opposée en France où ces dernières constituent près de 25% du parc d'appareils. Face à un tel marché, les sociétés et développeurs ne peuvent se contenter d'applications fonctionnant parfaitement sur smartphones mais proposant un affichage passable sur tablettes. Les développeurs se doivent de proposer une expérience utilisateur de qualité sur tablettes ou à minima de spécifier qu'ils ne souhaitent pas cibler cette gamme d'appareils. Ceci dans le but d'éviter des retours désastreux d'utilisateurs qui nuisent grandement à la réputation d'une application sur le Google Play Store.

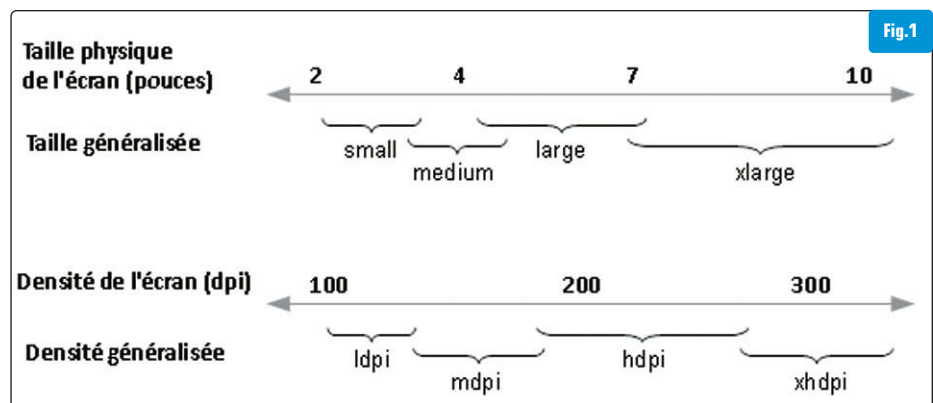
SUPPORT DES ÉCRANS EXTRA-LARGES

La première particularité des tablettes provient de leurs écrans extra-larges en regard de ceux équipant les smartphones, bien que pour ces derniers aussi la tendance soit de proposer des écrans toujours plus grands. Dans ce contexte, il est nécessaire de mettre en place un support multi-écrans au sein des applications. Pour simplifier cette gestion, Android propose un certain nombre de concepts et d'abstractions au centre desquels on retrouve la taille des écrans et leur densité. Exprimée en dpi, la densité d'un écran correspond à la quantité de pixels qu'une zone physique d'un écran peut contenir. Au sein du système, ces paramètres sont regroupés dans les ensembles généralisés suivants :

- Taille des écrans : small, normal, large, xlarge
- Densités : ldpi (faible densité), mdpi (moyenne), hdpi (haute), xhdpi (extra)

Le schéma de la [Fig.1] montre la correspondance que le système réalise au runtime entre ces données généralisées et les écrans physiques des appareils.

En outre, ces tailles d'écrans généralisées ont une résolution minimum associée, elle-aussi exprimée en dp, toujours dans un souci



Correspondance données généralisées et physiques pour les écrans

d'indépendance vis-à-vis des données physiques des écrans. On a ainsi les résolutions minimum suivantes :

- Écrans xlarge au moins 960dp x 720dp
- Écrans large au moins 640dp x 480dp
- Écrans normal au moins 470dp x 320dp
- Écrans small au moins 426dp x 320dp

Ce système d'abstraction prend tout son sens en comparant par exemple des appareils tels que le Nexus 4 et le Nexus 7 de Google. Dotés respectivement d'écrans de 4.7 et 7 pouces, ces 2 appareils ont à peu de choses près la même résolution en pixels aux alentours des 1280 x 800. Néanmoins, leurs écrans sont largement différents ! En intégrant la notion de densité détaillée précédemment et en pensant en dips plutôt qu'en pixels, il est plus aisé de conceptualiser la différence entre ces écrans puisque l'on obtient alors des résolutions qui n'ont plus rien à voir. Ainsi, celle du Nexus 4 est de 640 x 384 dp alors que celle du Nexus 7 est de 960 x 600 dp.

Pour les tablettes, la cible visée se situe au sein des catégories large et xlarge. Les tablettes 7 pouces telles que la Nexus 7 étant généralement en large alors que celles de taille supérieure se rapprochant des 10 pouces telles que la Nexus 10 ou la Samsung Galaxy Tab 2 seront généralement en xlarge.

LAYOUTS OPTIMISÉS

Afin de cibler au mieux les tablettes en proposant des layouts optimisés pour celles-ci, Android propose 2 méthodes. La première consiste à restreindre le spectre des périphériques pour lesquels

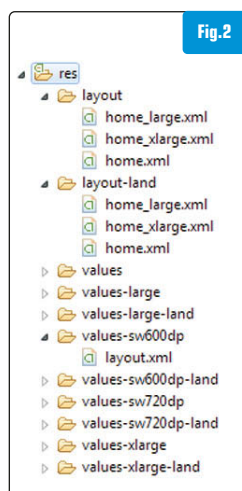


une application sera disponible en déclarant explicitement au sein de son manifest les tailles d'écran supportées. On pourra ainsi spécifier qu'une application sera uniquement accessible pour des écrans extra larges de la sorte :

```
<support-screens android:smallScreens="false" android:normalScreens="false" android:largeScreens="false" android:xlargeScreens="true" />
```

Cette solution s'avère limitée et il est préférable de créer des layouts spécifiques pour chaque type d'écran en laissant au système le soin de choisir le layout adapté suivant l'appareil où l'application est installée. Pour choisir la ressource adaptée au contexte, Android propose de suffixer les dossiers de ressources avec un certain nombre de qualificatifs prédéfinis. Avant la version 3.2 d'Android, le système proposait 4 classes de qualificatifs pour spécifier les caractéristiques de taille, de densité, d'orientation ou de ratio. A partir d'Android 3.2 et donc pour les versions actuelles tournant sous Jelly Bean, 3 nouveaux qualificatifs de taille d'écran sont mis à disposition des développeurs.

Parmi ces nouveaux qualificatifs, on note sw<N>dp permettant de spécifier une largeur d'écran minimale pour qu'un layout soit considéré par l'OS. Près de 45% des appareils Android étant encore en version inférieure à la 3.2, il est nécessaire de définir des ressources alternatives basées sur les qualificatifs large et xlarge mais également sur le qualificatif de largeur minimale. Pour éviter de dupliquer des fichiers layouts, le système propose un système d'alias très puissant donnant la possibilité de définir dans un seul répertoire layout l'ensemble des layouts utilisés pour une activité donnée et de pointer ensuite vers le bon layout dans les fichiers de ressources adressés par l'OS au runtime.



Hiérarchie de ressources pour support multi-écrans

Ainsi, dans le cas d'une application souhaitant servir des layouts optimisés pour smartphones et tablettes, la hiérarchie de fichiers layouts pourra être telle que présentée à la [Fig.2].

Les tablettes étant majoritairement utilisées en mode landscape, on remarque l'importance de définir des layouts spécifiques pour cette orientation. Les différents layouts utilisés sont de fait centralisés au sein des dossiers layout et layout-land. Les autres dossiers de ressources sont suffixés par des qualificatifs. Ainsi, pour les tablettes de 7 pouces en version supérieure à la 3.2 le fichier de ressources layout.xml est considéré puisque placé au sein du dossier values-sw600dp. Au sein du fichier layout, on utilise par exemple l'alias suivant :

```
<resources>
  <item name="home" type="layout">@layout/home_large</item>
</resources>
```

Une fois la hiérarchie de ressources mise en place, il est bon de rappeler quelques bonnes pratiques à garder à l'esprit lors de la création du contenu de ces layouts. Les bonnes pratiques se résument à

4 points clés :

1. Utiliser wrap_content, fill_parent ou des unités en dp pour les dimensions des layouts
2. Ne pas utiliser de valeurs en pixels dans le code applicatif
3. Ne pas utiliser l'AbsoluteLayout
4. Utiliser au maximum les ressources alternatives pour les drawables

Le respect de ces points permettant déjà de réaliser des applications offrant une expérience utilisateur optimisée au plus grand nombre d'appareils.

IHM DYNAMIQUES

Apparus avec Honeycomb, les fragments sont désormais pleinement accessibles depuis Android 4 et peuvent également être mis à profit sur d'anciennes moutures de la plateforme via la bibliothèque de support. Dans le même esprit que ce que le Responsive Web Design permet pour les IHM des sites Web, les fragments vont permettre de construire des IHM dynamiques s'adaptant au contexte d'affichage. En pratique, une même base de code permettra de tirer profit de l'ensemble de l'espace d'affichage disponible quel que soit l'appareil. Un fragment représente un comportement ou une portion d'interface utilisateur se trouvant au sein d'une activité ayant pour but d'aider à organiser les composants graphiques de manière réutilisable entre activités. De fait, un fragment pourra être réutilisé sur différents layouts.

En prenant l'exemple d'une application météo fournissant des cartes météo par pays, on pourra considérer que cette fonctionnalité se divise en 2 activités distinctes sur Smartphones : une première listant l'ensemble des pays pour lesquels une carte météo est disponible [Fig.3] et une seconde affichant la carte météo du pays sélectionné sur la liste précédente [Fig.4].

Sur des écrans de 4 pouces, le découpage en 2 activités distinctes se justifie parfaitement du fait de la taille de l'écran mais sur des tablettes de 7 pouces et plus, on se retrouve avec une part d'espace d'affichage inutilisée importante. Ceci nuisant fortement à l'expérience utilisateur, il est préférable de découper l'IHM en fragments pour proposer l'affichage au sein d'un même écran de la liste des pays

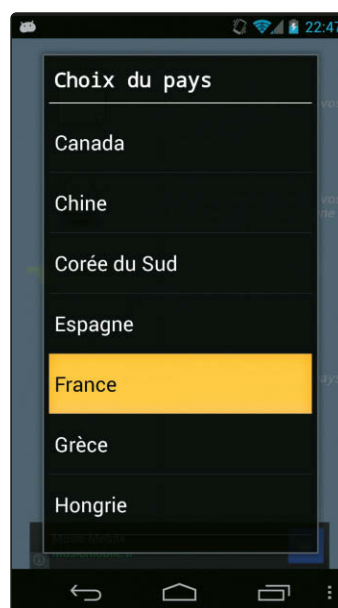


Fig.3 Activité listant les pays

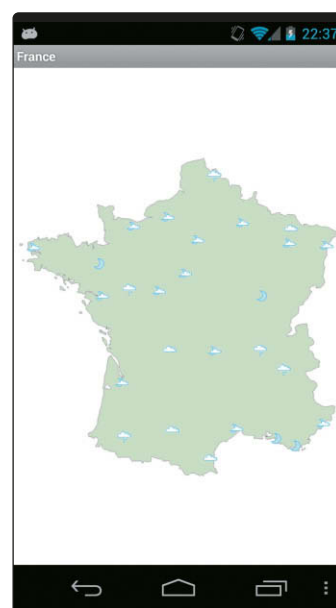


Fig.4 Activité affichant la carte météo du pays

ainsi que de la carte météo associée pour les tablettes. Le gros du travail consiste à créer les fragments `CountriesFragment` et `MapFragment` permettant respectivement d'afficher la liste des pays ainsi qu'une carte météo donnée. Au sein des layouts, on met en place un `map_meteo_layout` classique pour l'affichage normal, c'est-à-dire au sein du dossier layout :

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent">
    <fragment class="com.app.fastmeteo.fragments.CountriesFragment"
        android:id="@+id/countries"
        android:layout_width="match_parent" android:layout_height="match_parent" />
</FrameLayout>
```

En revanche, au sein du dossier `layout-sw600dp`, on définit ce même layout comme suit :

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent" android:layout_height="match_parent">
    <fragment class="com.app.fastmeteo.fragments.CountriesFragment"
        android:id="@+id/countries" android:layout_weight="30"
        android:layout_width="0px" android:layout_height="match_parent" />
    <FrameLayout android:id="@+id/map" android:layout_weight="70"
        android:layout_width="0px" android:layout_height="match_parent"
        android:background="?android:attr/detailsElementBackground" />
</LinearLayout>
```

On note ici l'ajout d'un `FrameLayout` servant à accueillir le fragment `MapFragment`. La mécanique de gestion des fragments se réalise au sein du `CountriesFragment`. Ce dernier ajoutant le `MapFragment` à l'IHM suivant le contexte d'affichage ou bien basculant vers la classe `MapActivity` lors d'un clic sur un pays de la liste. Présenté à la figure 5, le résultat montre clairement l'intérêt et la puissance des fragments dans la construction d'IHM dynamiques sous Android afin de répondre aux problématiques d'affichage multi-écrans [Fig.5].



Fig.5 Ecran partage pour tablettes

UTILISATION DE L'ACTION BAR

Apparue avec la version 3.0, l'Action Bar est un composant qui remplace la traditionnelle barre de titre située en haut de l'écran des applications. Élément central des nouveaux guides de conception IHM mis en avant par Google, l'Action Bar offre une meilleure interaction entre l'application et ses différentes fonctionnalités. En fluidifiant et facilitant la navigation entre celles-ci, elle est un remplaçant de qualité du menu options utilisé sur les vieilles versions d'Android. Par défaut, la barre d'applications inclut le logo de l'application sur son côté gauche, le titre de l'activité et éventuellement sur son côté droit, une liste proposant les items définis au sein du menu options de l'activité. Certains d'entre eux pouvant être directement intégrés au sein de l'Action Bar via l'attribut `android:showAsAction` positionné à "ifRoom" lors de la déclaration de l'item au sein du fichier XML du menu. La méthode callback appelée lors du clic sur un de ces items demeurant toujours `onOptionsItemSelected()` de la classe `Activity`. D'autre part, la barre d'actions peut également être employée sur tablettes pour afficher une élégante barre de navigation entre onglets pointant chacun vers un fragment d'une activité par exemple. Ce cas d'usage s'avère particulièrement adapté dans le cas où les utilisateurs doivent fréquemment switcher entre différents écrans. Néanmoins, un problème de compatibilité avec les versions inférieures à la 3.0 se pose puisque Google ne fournit pas de bibliothèque de support pour ces dernières. Pour combler ce manque, la bibliothèque `ActionBarSherlock` est la solution idéale. Disponible à l'adresse suivante : <http://actionbarsherlock.com/>, cette bibliothèque permet d'utiliser l'Action Bar quelle que soit la version d'Android et ce de manière totalement transparente pour le développeur. La bibliothèque choisit automatiquement d'utiliser l'Action Bar native si la version de l'OS le permet et une implémentation équivalente sur les versions inférieures.

En considérant une Action Bar utilisée pour afficher une barre de navigation entre onglets, le code de l'activité a l'allure suivante :

```
public class HomeActivity extends SherlockActivity implements
ActionBar.TabListener {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.home);
        getSupportActionBar().setNavigationMode(ActionBar.NAVIGATION
_MODE_TABS);
        // création des onglets
        ActionBar.Tab tab = getSupportActionBar().newTab();
        tab.setText(R.string.firstTab);
        tab.setTabListener(this);
        getSupportActionBar().addTab(tab);
        tab = getSupportActionBar().newTab();
        tab.setText(R.string.secondTab);
        tab.setTabListener(this);
        getSupportActionBar().addTab(tab);
    }

    @Override
    public void onTabReselected(Tab tab, FragmentTransaction
transaction) {
    }
}
```




```
@Override
public void onTabSelected(Tab tab, FragmentTransaction transaction) {
    if (getString(R.string.firstTab).equals(tab.getText())) {
        // ...
    } else {
        // ...
    }
}

@Override
public void onTabUnselected(Tab tab, FragmentTransaction transaction) {
}

public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getSupportMenuInflater();
    inflater.inflate(R.menu.myMenu, menu);
    return super.onCreateOptionsMenu(menu);
}
}
```

- android.hardware.telephony
- android.hardware.camera
- android.hardware.camera.front

Enfin, si une application nécessite le GPS pour fonctionner une idée intéressante peut être de proposer une version dégradée sur les appareils ne possédant pas de capteur GPS en laissant la possibilité aux utilisateurs de définir eux-mêmes leurs positions. Pour ce faire, l'application doit faire une vérification à l'exécution de la présence d'une fonctionnalité hardware avant de choisir de basculer éventuellement en mode dégradé le cas échéant.

DIVERS

Au niveau des ressources de type drawables, le support multi-écrans nécessite de fournir au système des images suivant le ratio 3/4/6/8 entre les 4 catégories généralisées de densités d'écrans. Ainsi, si une image est de taille 48x48 en mdpi qui est la référence, il faut alors fournir une image équivalente en 96*96 dans le dossier res/drawable-xhdpi. Ce principe similaire à celui employé pour les layouts permettant d'obtenir des images adaptées au contexte d'affichage. Il en va de même pour les polices de caractère pour lesquelles il est bon de définir des tailles spécifiques pour les tablettes de 7 ou de 10 pouces en utilisant les styles et les alias qui sont comme pour les layouts disponibles sur Android. Enfin, les Widgets doivent tirer parti du plus grand espace disponible sur les tablettes en étant redimensionnables à 420dp de largeur ou plus dans le but de couvrir 5 lignes de l'écran d'accueil au minimum.

CONCLUSION

Le développement d'applications sous Android ne peut se faire désormais sans prendre en compte les tablettes sous peine de se priver d'un grand nombre d'utilisateurs et d'un marché à fort potentiel de croissance. La solution idéale étant de fournir une application offrant un support multi-écrans. Pour ce faire, l'OS propose un grand nombre d'outils facilitant le travail du développeur qui ont été présentés au sein de cet article. Les fragments, la barre d'actions ainsi que le puissant système de gestion des ressources alternatives permettant de servir des layouts dédiés constituent des outils précieux pour les développeurs. Pour tester ces différentes configurations, l'émulateur Android fourni en standard s'avère suffisant puisque permettant de simuler des tablettes de toute sorte.

Sylvain Saurel – Ingénieur d'Etudes Java / JEE
sylvain.saurel@gmail.com

Enfin, cette barre d'actions se révèle également idéale pour afficher des vues plus interactives comme une zone de recherche par exemple. Pour ce, on rajoute dans la balise de déclaration de l'item l'attribut android:actionLayout pointant vers le layout utilisé par cette vue action et on associe les listeners adéquats à la création des items. La possibilité de rendre l'Action Bar contextuelle est également intéressante et permet d'améliorer grandement l'expérience utilisateur sous Android et ce quel que soit le terminal ciblé.

PRISE EN COMPTE DU HARDWARE

Les tablettes proposent un ensemble de fonctionnalités matérielles légèrement différentes des Smartphones. Les capteurs, la caméra ou la téléphonie sont parmi les principales fonctionnalités pouvant différer. Ainsi, beaucoup de tablettes proposent seulement un accès internet via Wi-Fi. Afin d'être sûr de rendre une application disponible à l'ensemble des utilisateurs Android, il est important de vérifier qu'une application n'a pas des exigences spécifiques qui sont la plupart du temps absentes des tablettes. De fait, il faut éviter d'inclure des éléments uses-features au sein du manifest désignant des fonctionnalités hardware absentes des tablettes généralement. De même, certaines permissions impliquent certaines exigences matérielles. Il faudra donc veiller à ne pas les déclarer au sein d'éléments permission du manifest. Principalement, cela concerne les fonctionnalités suivantes :

L'information permanente

- L'actu de Programmez.com : le fil d'info quotidien
- La newsletter hebdo : la synthèse des informations indispensables.

Abonnez-vous, c'est gratuit !

www.programmez.com

The screenshot shows the homepage of Programmez.com. At the top, there's a navigation bar with links like 'Accueil', 'Actualités', 'Téléchargements', 'Livres', 'Requêtes', 'Bibliothèque', 'Blogs', 'Partenaires', 'Contact', and 'Recherche'. Below the navigation bar, there are several featured articles and advertisements. On the left, there's a section for 'Abonnement' (Subscription) with details about the newsletter. In the center, there are articles about 'Concours Azure/Raytheon', 'Nouveaux flashs sur Programmez!', 'Le Groupe KPI, SSI spécialiste sur SAP', 'SUPERMON for JAVA', and 'Tribune Libre'. On the right, there are advertisements for 'Microsoft Windows Azure', 'IRIScan Book 2', and 'Programmez.com n°149'. The bottom of the page features a 'UN BUG + VISEUX' section and a 'Contact' link.

Je débute sur Android et iOS

Les tablettes sont définitivement entrées dans notre vie quotidienne ! On le voit au succès de l'iPad et des diverses tablettes sous Android, comme la Nexus 7 (certains voient déjà leurs ventes dépasser celles des PC de bureau). Dans l'entreprise aussi, l'adoption des tablettes devient une réalité. Mais alors, que doit-on savoir lorsque l'on souhaite développer une application sur tablette ou porter son application mobile existante sur tablette ?

La tablette n'est ni un smartphone plus large, ni un PC portable aux dimensions réduites. Elle est les deux à la fois : elle a su se faire une place à part entière dans le monde des périphériques connectés que nous côtoyons aujourd'hui. Ainsi, même si mobiles et tablettes partagent le même système d'exploitation (principalement iOS et Android), proposer exactement la même application pour les deux formats est très rarement adapté.

En premier lieu, les résolutions et les tailles d'écrans ne sont pas les mêmes : il est possible d'afficher beaucoup plus d'éléments et de fonctionnalités sur une tablette. Pour un même cas d'utilisation, il sera donc possible, sur tablette par exemple, de diminuer le nombre d'écrans par rapport à une application mobile. Les usages mêmes sont très différents : le mobile s'utilise abondamment dans un cadre itinérant (dans les transports par exemple) avec une connexion type 3G (il faut alors prendre en compte le chargement des pages ou des données et la perte de connexion potentielle). À l'opposé, la tablette est utilisée dans un cadre domestique ou professionnel, souvent à proximité d'un réseau WIFI. L'usage du mobile est principalement personnel, tandis que la tablette a une utilisation plus partagée : l'écran peut être vu par plusieurs personnes et permet plus facilement de partager la donnée visualisée. La tablette invite à l'interaction sociale. Enfin, les applications mobiles préfèrent rester consistantes dans leur IHM vu le peu de place disponible, alors que dans le cadre des tablettes, la capacité à innover et inventer de nouvelles interfaces - plus "sexy", plus naturelles - permet de surprendre l'utilisateur.

La tablette est rapidement devenu un format à prendre en compte lors de la réalisation d'une application mobile. Cet article va vous

éclairer sur les aspects ergonomiques à prendre en compte et les développements spécifiques que vous devrez mettre en oeuvre aussi bien sur iOS que sur Android.

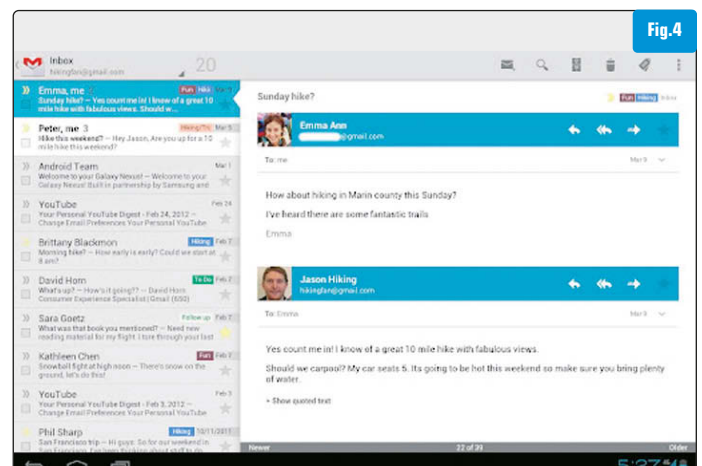
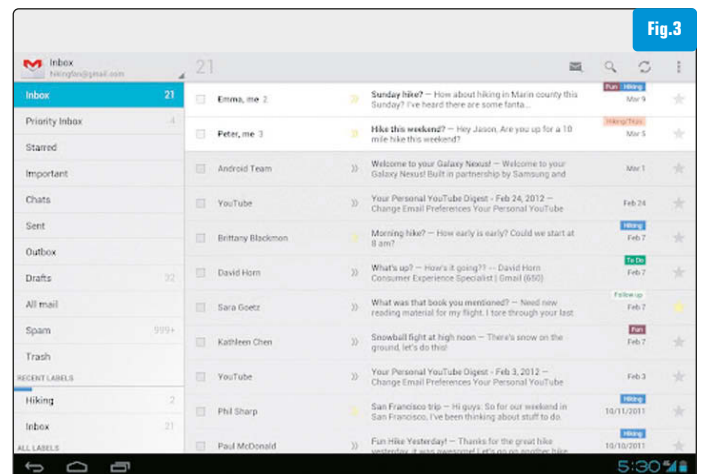
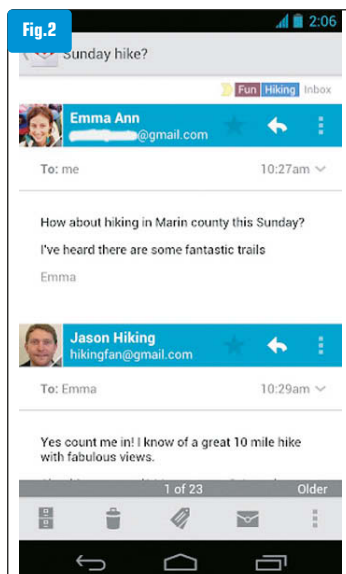
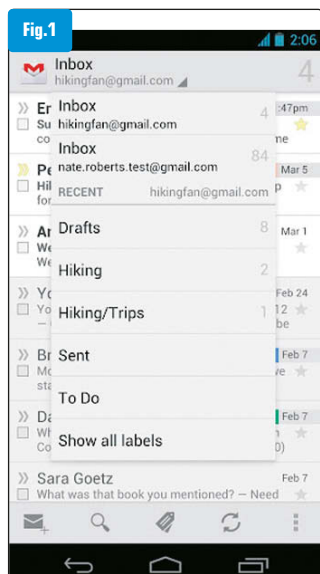
ADAPTER L'EXPÉRIENCE UTILISATEUR À LA TABLETTE

L'écran d'une tablette est plus grand que celui d'un téléphone, il est donc possible de placer plus d'éléments. Par exemple, l'application Gmail pour tablette affiche en permanence la liste des labels dans une colonne de gauche contrairement à l'application mobile où il faut aller les sélectionner par un menu en bas de l'application [Fig.1 et 2].

Les fonctionnalités dossiers, liste des mails et le corps du message sur mobile pour l'application Gmail [Fig.3 et 4].

Les mêmes fonctionnalités de l'application vues sur une tablette.

Il faut tout de même faire attention à ne pas surcharger l'IHM d'une tablette bien que celle-ci ait un plus grand écran.





D'autre part, il est possible de changer la façon d'interagir avec une tablette. Sur mobile, la façon dont l'utilisateur le tient classiquement fait que le placement d'une barre d'action en bas de l'écran est plus ergonomique car elle se retrouve plus proche du pouce. Au contraire sur une tablette, on peut sans problème mettre la barre d'action en haut de l'écran car on dispose plus souvent d'une main de libre qui pourra facilement aller cliquer sur ces éléments. Un autre exemple est le clavier virtuel SwiftKey qui propose une disposition différente sur tablette en mode paysage car les mains peuvent être alors disposées de chaque côté de la tablette pour accélérer la saisie. Si vous voulez comprendre n'hésitez pas à prendre un mobile ou une tablette et essayez!

Enfin, si on peut choisir de se limiter au mode portrait pour les applications mobiles, il faut que les applications tablettes supportent les deux types d'orientation en sachant que l'orientation paysage est la plus utilisée [Fig.5 et 6].

PRENDRE EN COMPTE TECHNIQUEMENT LES TABLETTES ET L'ORIENTATION SUR ANDROID

Avec la version Honeycomb d'Android est apparu un nouveau type de composant permettant de faciliter le développement sur tablette : le Fragment. Pour faire simple, il s'agit d'une portion réutilisable de l'interface utilisateur d'une Activity. Une activité peut contenir plusieurs fragments qu'il est possible d'ajouter ou retirer même si elle est en cours d'exécution. Cette modularité permet de construire une interface qui s'adaptera à l'espace disponible sur l'écran.

Le Fragment gère ses propres événements et a un cycle de vie dédié. Cependant, ce dernier est directement lié au cycle de vie de l'activité qui le contient. D'ailleurs, les callback méthodes d'un Fragment sont similaires à celles d'une Activity, on retrouve des méthodes comme onCreate(), onPause(), etc.

Maintenant, nous allons voir au travers d'un exemple comment mettre en place des fragments au sein d'une application qui aura un workflow comme suit : [Fig.7].

Tout d'abord nous allons créer le fragment qui affichera la liste d'éléments que l'utilisateur pourra sélectionner pour en afficher le détail. Pour cela, nous allons utiliser un dérivé de la classe Fragment : ListFragment.

ItemsListFragment

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/
```

```
res/android>
```

```
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">
    <TextView android:id="@+id/itemName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

Quand au fragment affichant le détail d'un item :

item_details_layout.xml

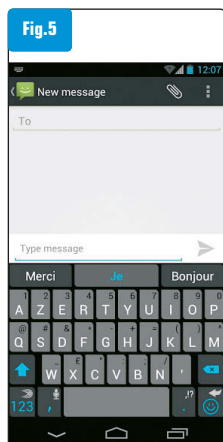
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/
res/android>
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">
    <TextView android:id="@+id/itemName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

ItemDetailsFragment

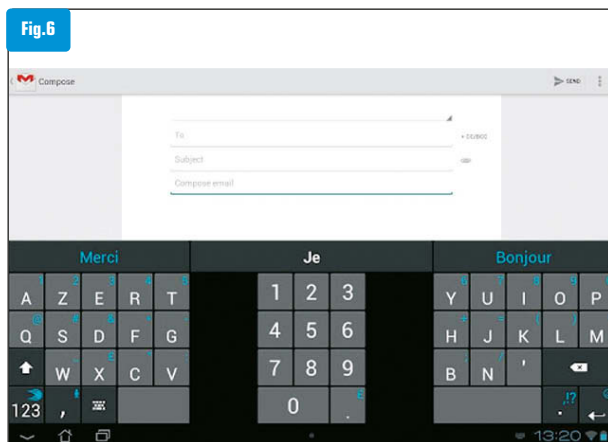
```
public class ItemDetailsFragment extends Fragment {

    private TextView itemNameTxt;

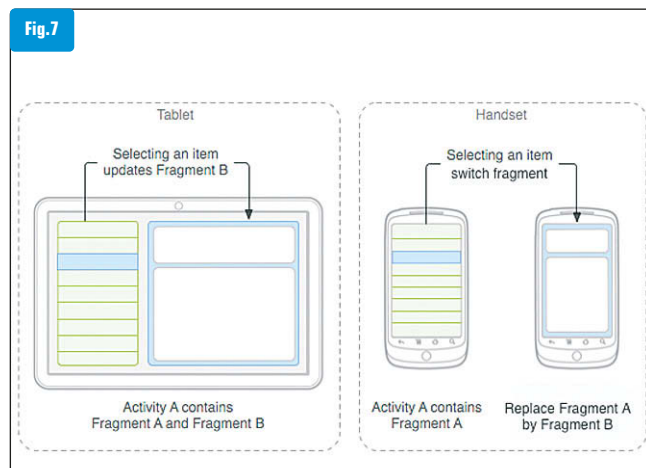
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.item_details_layout,
container, false);
        itemNameTxt = (TextView) view.findViewById(R.id.itemName);
        if(getArguments() != null){
            // On a reçu les données à afficher via les arguments du fragment
            updateItemDetails(getArguments());
        }
        return view;
    }
}
```



Le clavier en orientation portrait sur mobile



Le même clavier en orientation paysage sur tablette




```
public void updateItemDetails(Bundle selectedItem){
    // On récupère le nom de l'item et on l'affiche
    itemNameTxt.setText(selectedItem.getString("name"));
}
}
```

Maintenant que nous avons implémenté nos deux fragments, nous devons créer un layout adapté suivant le device. A partir de la version 3.2 d'Android il est possible de spécifier la largeur minimum que doit faire un écran pour afficher un layout donné.

Nous mettons celui de notre tablette dans le dossier res/layout-sw600dp (sw pour smallest width), ce qui signifie que seul un appareil ayant une capacité d'au moins 600dp en largeur (qu'importe l'orientation) comme le Nexus 7 pourra l'afficher.

Pour les tablettes pré-3.2 le dossier res/layout-x-large devra être privilégié car les ressources du dossier layout-sw600dp seront tout simplement ignorées.

En ce qui concerne le layout des téléphones, il sera composé d'un simple `FrameLayout` qui jouera le rôle de conteneur de fragments et sera placé dans le répertoire res/layout.

main.xml sur tablette

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/
android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <fragment
        class="fr.programmez.tablette.ItemsListFragment"
        android:id="@+id/items_list_frag"
        android:layout_height="match_parent"
        android:layout_width="0dp"
        android:layout_weight="1" />
    <fragment
        class="fr.programmez.tablette.ItemDetailsFragment"
        android:id="@+id/item_details_frag"
        android:layout_height="match_parent"
```

```
android:layout_width="0dp"
android:layout_weight="3" />
</LinearLayout>
```

main.xml sur téléphone

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/
android"
    android:id="@+id/fragment_container"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"/>
```

Il est temps de faire entrer en jeu notre activité qui a pour rôle de gérer nos deux fragments. Cette dernière a besoin de savoir sur quel type de device elle est lancée afin de déterminer quel mode d'affichage sera utilisé. Pour y parvenir nous créons une ressource de type boolean `multi_pane` qui vaut `false` par défaut sauf dans les dossiers res/values-sw600dp et res/values-x-large.

Dans le cas d'une application s'exécutant sur un téléphone, nous faisons appel au `FragmentManager` qui va nous permettre de remplacer le contenu du `FrameLayout` par le fragment souhaité.

Cela se passe sous forme d'une transaction au cours de laquelle nous précisons quel fragment nous souhaitons voir affiché. Il est possible d'historiser ces transactions (méthode `addToBackStack`) afin qu'un appui sur la touche back ait pour conséquence un retour à l'état précédent. Il est important de préciser que ce `FragmentManager` ne peut agir que sur des fragments ajoutés au runtime.

Pour finir dans le callback de sélection d'un item, nous remarquons que pour transmettre l'item sélectionné nous utilisons les arguments du fragment. Il s'agit d'un bundle que l'on peut passer à un fragment lors de sa création et ainsi le récupérer au moment où la vue sera créée.

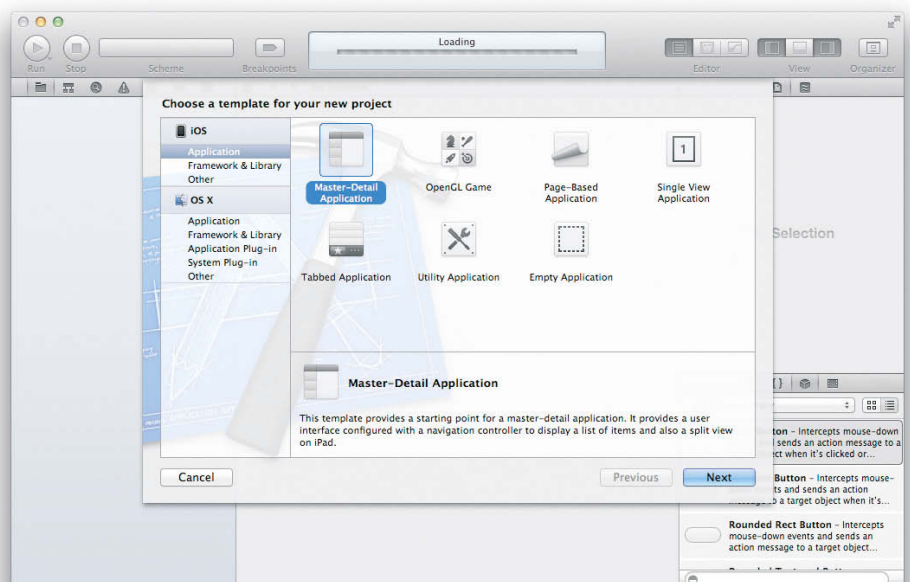
Lors d'une exécution sur tablette, pas besoin de transaction. Nous récupérons directement une référence vers le fragment déclaré dans le fichier xml grâce à la méthode `findFragmentByld` du `FragmentManager`.

De même, le traitement est simplifié dans le callback de sélection

Fig.8



Fig.9





d'un item car nous sommes sûrs que le fragment est créé et attaché à l'activité en cours. Il est alors possible d'appeler directement la méthode `updateItemDetails()` exposée par notre fragment.

SampleActivity

```
public class SampleActivity extends Activity implements ItemListFragment.Callback {

    /**
     * Référence vers le fragment affichant le détail (uniquement
     * utilisé pour les tablettes)
     */
    private ItemDetailsFragment itemDetailsFrag;

    /**
     * Sommes-nous sur une tablette ?
     */
    private boolean multiPane;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        multiPane = getResources().getBoolean(R.bool.multi_pane);
        if (multiPane) {
            // Le fragment est déclaré dans le xml, on récupère sa référence
            itemDetailsFrag = (ItemDetailsFragment) getFragmentManager()
                .findFragmentById(R.id.item_details_frag);
        } else {
            // On remplace le contenu du FrameLayout par le fragment
            // affichant la liste des items
            getFragmentManager().beginTransaction() //
                .replace(R.id.fragment_container, new ItemListFragment()) //
                .commit();
        }
    }

    @Override
    public void onItemSelected(Bundle selectedItem) {
        if (multiPane) {
            itemDetailsFrag.updateItemDetails(selectedItem);
        } else {
            ItemDetailsFragment fragment = new ItemDetailsFragment();
            // On met l'item sélectionné dans les arguments du Fragment
            // pour pouvoir le récupérer lors de sa création
            fragment.setArguments(selectedItem);
            // On remplace le fragment affichant la liste par celui du
            // détail de l'item
            getFragmentManager().beginTransaction() //
                .replace(R.id.fragment_container, fragment) //
                .addToBackStack(null) //
                .commit();
        }
    }
}
```

Nous obtenons le résultat suivant sur tablette (un fond blanc a été utilisé afin de distinguer les deux fragments) : [Fig.8](#)

Alors que sur téléphone, l'accès à l'ensemble des écrans se fait en 2 temps.

PRENDRE EN COMPTE TECHNIQUEMENT LES TABLETTES ET L'ORIENTATION SUR IOS

Depuis la sortie de l'iPad, et avec iOS 3.2, Apple a introduit le `UISplitViewController`, un composant natif permettant de gérer automatiquement l'affichage des vues sur l'écran d'une tablette.

Le `UISplitViewController` consiste donc en un conteneur hébergeant les composants visuels de deux `ViewControllers`.

La vue du `UIViewController` de gauche, plus petit, a une taille de 320 points et est toujours affichée en mode paysage mais cachée automatiquement en mode portrait.

Dans ce dernier cas, on peut l'afficher en utilisant un bouton placé sur la barre de navigation ou (à partir d'iOS 5.1) en glissant le doigt de gauche à droite avec un geste de swipe. La vue de droite du `UIViewController` est par conséquent affichée en plein écran quand l'iPad se trouve en mode portrait.

Le `UISplitViewController` est très souvent utilisé pour mettre en oeuvre une interface de type master-détail qui consiste à afficher, dans la vue de gauche (le master), une liste d'éléments et, dans la vue de droite (le détail), les détails de l'élément sélectionné dans le master. La taille du master n'est par ailleurs pas une coïncidence : il s'agit en fait de la largeur en points de l'écran de l'iPhone. Ce choix permet aux développeurs de réutiliser complètement l'interface utilisateur conçue pour afficher une liste sur le téléphone.

Nous allons expliquer, avec un exemple simple, comment mettre en oeuvre notre logique master-détail sur iPhone et iPad.

La méthode la plus simple pour utiliser le composant `UISplitViewController` est de créer un nouveau projet avec XCode et d'utiliser le gabarit "Master-Detail Application". Avec cette opération on obtiendra deux `Controllers` (un `MasterViewController` et un `DetailViewController`) aussi bien que les Storyboards pour iPhone et iPad [Fig.9](#).

Comment faire dans le cas où nous avons déjà une application iPhone avec une `UITableView` et que nous voulons la rendre compatible avec iPad ?

Nous allons supposer, pour notre exemple, que nous avons déjà codé un `UITableViewController`, nommé "ItemTableViewController", contenant une liste d'éléments, et un `UIViewController` nommé "ItemDetailViewController" contenant le détail d'un élément.

ItemTableViewController.h

```
#import <UIKit/UIKit.h>

@interface ItemTableViewController : UITableViewController

@end
```

ItemTableViewController.m

```
#import «ItemTableViewController.h»
#import «ItemDetailViewController.h»

@interface ItemTableViewController () {
    NSArray *_items;
}

@end
```

```
@implementation ItemTableViewController

- (id)initWithStyle:(UITableViewStyle)style
{
    self = [super initWithStyle:style];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    _items = @[«Item 1», «Item 2», «Item 3», «Item 4»];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
}

#pragma mark - Table view data source

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:
(NSInteger)section
{
    return _items.count;
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:
(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = «Cell»;
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier
forIndexPath:indexPath];
    cell.textLabel.text = _items[indexPath.row];

    return cell;
}

#pragma mark - Table view delegate

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)
sender
{
    if ([[segue identifier] isEqualToString:«showItemDetail»]) {
```

```
        NSIndexPath *indexPath = [self.tableView indexPathForSelected
Row];
        NSString *item = _items[indexPath.row];
        [[segue destinationViewController] setItem:item];
    }
}
@end
```

ItemDetailViewController.h

```
#import <UIKit/UIKit.h>

@interface ItemDetailViewController : UIViewController

@property (weak, nonatomic) IBOutlet UILabel *itemInfo;

@property (strong, nonatomic) NSString *item;

@end
```

ItemDetailViewController.m

```
#import «ItemDetailViewController.h»

@interface ItemDetailViewController ()
@end

@implementation ItemDetailViewController

- (void)viewDidLoad
{
    [super viewDidLoad];

    [self showItemDetail];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
}

- (void)showItemDetail {
    if (self.item) {
        self.itemInfo.text = self.item;
    }
}

@end
```

Dans le cadre de l'interface master-détail, ItemTableViewController sera notre master et ItemDetailViewController notre détail.

Nous commençons par changer le type de dispositif ciblé par notre application : sélectionnons le fichier de projet, puis notre target, et dans le menu déroulant "Devices", activons "Universal". Plus bas, dans la section "iPad Deployment Info" changeons le nom du "Main Storyboard" et saisissons, par exemple, "MainStoryboard_iPad". Ce storyboard contiendra l'interface utilisateur de notre application universelle et, si nous ne l'avons pas encore générée, il ne nous reste qu'à en créer une nouvelle à partir du menu File -> New -> User Inter-



face. N'oublions pas de sélectionner, ensuite, la "Device Family" qui, pour notre application, sera "iPad".

À l'intérieur de notre nouveau Storyboard nous allons ajouter à partir de l'Object Library un UISplitViewController. Plusieurs scènes seront créées automatiquement, avec leurs segues. Nous allons ensuite sélectionner le Table View Controller, dans l'Identity Inspector à droite de l'écran, on changera la Class, en saisissant "ItemTableViewController". La même chose doit être faite pour le Detail View Controller et sa classe devra être "ItemDetailViewController". Nous pouvons ensuite personnaliser les cellules du tableau sans oublier de définir le même "Identifiant" utilisé sur iPhone.

Il ne reste plus qu'à adapter notre ItemTableViewController pour le rendre compatible avec iPad. On a tout d'abord besoin de référencer notre ItemDetailViewController dans le ItemTableViewController.

Dans l'extension du ItemTableViewController on ajoute donc la ligne suivante, permettant de créer une property qui référencera le détail :

```
@property (weak, nonatomic) ItemDetailViewController *detailViewController;
```

À l'intérieur de la méthode viewDidLoad on va ensuite ajouter la ligne suivante :

```
self.detailViewController = (ItemDetailViewController *)[self.splitViewController.viewControllers lastObject];
```

qui enregistre dans la propriété que nous avons créée l'instance courante du ItemDetailViewController. Dans le block "implementation", on ajoute ensuite la méthode suivante :

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath {
    NSIndexPath *indexPath = [self.tableView indexPathForSelectedRow];
    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad) {
        NSString *item = _items[indexPath.row];
        self.detailViewController.item = item;
    }
}
```

Cette méthode, issue de UITableViewDelegate, communique au delegate de la UITableView (qui n'est rien d'autre que notre instance du ItemDetailViewController) qu'une ligne du tableau a été sélectionnée. Avec le bloc (if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad) nous vérifions si le dispositif que l'utilisateur est en train d'utiliser est bien un iPad et, dans ce cas, on définit la propriété item du detailViewController.

À l'intérieur du ItemDetailViewController on ajoute la méthode suivante, permettant de forcer l'affichage des informations à chaque fois qu'on changera la propriété item :

```
- (void)setItem:(NSString *)item {
    _item = item;
    [self showItemDetail];
}
```

On peut d'ores et déjà essayer l'application sur iPad : la liste des items s'affiche seulement en paysage et la sélection d'un élément

change le contenu de la vue de détail.

Mais comment faire pour pouvoir accéder aux éléments de notre tableau même si on est en mode portrait ?

À partir de iOS 5.1, le UISplitViewController permet aux utilisateurs d'afficher la vue de gauche (notre master) avec un geste de swipe de gauche à droite sur l'écran. Afin d'activer cette fonctionnalité, on devra définir le delegate du UISplitViewController sur notre ItemDetailViewController.

Dans ItemDetailViewController.h il faudra modifier la ligne :

```
@interface ItemDetailViewController : UIViewController
```

en :

```
@interface ItemDetailViewController : UIViewController<UISplitViewControllerDelegate>
```

Et, finalement, dans la méthode viewDidLoad du ItemDetailViewController.m nous allons ajouter la ligne suivante :

```
self.splitViewController.delegate = self;
```

Si on voulait accéder aux éléments de notre tableau avec un bouton, nous pourrions utiliser les fonctionnalités du UISplitViewControllerDelegate qui permettent au splitViewController de notifier son delegate à chaque fois qu'on change le mode de présentation en mettant en oeuvre, dans ItemDetailViewController.m des méthodes du protocole UISplitViewControllerDelegate comme les suivantes :

```
- (void)splitViewController:(UISplitViewController *)splitViewController
willHideViewController:(UIViewController *)viewController
withBarButtonItem:(UIBarButtonItem *)barButtonItem
forPopoverController:(UIPopoverController *)popoverController

- (void)splitViewController:(UISplitViewController *)splitViewController
willShowViewController:(UIViewController *)viewController
invalidatingBarButtonItem:(UIBarButtonItem *)barButtonItem
```

CONCLUSION

Dans cet article, nous avons fait une introduction au développement sur tablette. Que l'on démarre de zéro ou que l'on souhaite adapter son application mobile existante, il est important de comprendre les paradigmes de navigation sur ce nouveau périphérique et également les usages attendus.

Lorsque vous vous lancez dans le développement d'une application mobile, nous vous conseillons de réfléchir au plus tôt à la déclinaison que votre application va prendre sur tablette : cela vous permettra de mutualiser une grande partie du code développé. En effet, même si l'IHM proposée est souvent différente, ou encore la navigation, la façon de récupérer les données ou encore de les stocker varie peu. Le développement d'une application tablette vous prendra généralement moins de 50% de l'effort consacré au développement de votre application mobile.

Thibaud **Cavin**, Responsable mobilité chez Xebia

Simone **Civetta**, Consultant chez Xebia

Yannick **Grenzinger**, Consultant chez Xebia

Thomas **Guérin**, Consultant chez Xebia.

Découverte de Windows 8

Lors de la première Build, en septembre 2011, Microsoft introduisait le nouveau Windows : Windows 8. Les milliers de développeurs présents découvraient un système marquant un véritable changement de cap pour Microsoft. Le géant s'adressait à nouveau au grand public.

LA CONCEPTION ET LE DESIGN

Quand on commence un projet d'interface pour tablette tactile il y a quelques points à prendre en compte et tous font partie de la phase de design. La phase de design se déroule toujours avant le début des développements et du graphisme.

Le premier point, c'est l'étude de l'usage pour établir la liste la plus pertinente possible de fonctionnalités à mettre dans l'application. Il faut toujours garder à l'esprit que plus n'est pas toujours mieux. Comme il est très bien dit dans les guidelines Modern UI une application doit avant tout être : simple, efficace et utile.

Le deuxième point est de concevoir une expérience utilisateur satisfaisante et ceci est possible uniquement grâce au design global de l'application. Lors de cette phase le designer dessine le storyboard complet en donnant à la fois le contenu, en expliquant les gestuelles, les interactions ainsi que le positionnement des différents éléments. Ce storyboard doit se faire avant le début des développements et du graphisme. Il peut ainsi servir de base de travail pour les développeurs.

Le troisième point, c'est l'aspect "usages". Quand utilise-t-on une tablette ? La majorité des utilisateurs utilise sa tablette en situation de semi-mobilité. Dans le canapé, dans la cuisine, en réunion et dans les transports lors de trajets assis. La situation dans laquelle votre utilisateur va utiliser votre application va être déterminante pour positionner les éléments dans l'interface. Pour une application Windows 8, il pourrait s'agir de déterminer si votre application doit pouvoir être utilisée en snap view par exemple.

Le quatrième point, c'est l'expérience continue au travers des différents device. Et si concevoir une application tablette n'était qu'une des briques de l'expérience plus globale ? Lorsqu'on commence à créer une application tablette mais qu'un site web et une application smartphone existent déjà, il

faut avant tout déterminer les raisons de cette situation d'interaction. L'idée est de comprendre quel est le but de l'application tablette. Si on prend par exemple une application pour une agence de voyage, la première étape serait « La découverte » parfaite sur tablette puis passer par le site web pour finaliser l'achat.

Grâce à ces quatre points lors de votre phase préalable de design, vous assurerez une expérience positive à vos utilisateurs.

LES CARACTÉRISTIQUES D'UNE APPLICATION WINDOWS 8

Une bonne application pour tablette sous Windows se caractérise au travers de différents aspects. En premier lieu, le nouveau langage de design « Modern UI » doit être maîtrisé. Ainsi la prise en compte des modes d'interactions tactiles et stylet seront facilités, il faudra cependant veiller à conserver une gestuelle cohérente tant au sein de l'application que par rapport au système lui-même. Impossible d'ignorer aujourd'hui à quel point les performances constituent à elles seules un pilier de l'expérience tactile. Enfin, la prise en charge des différentes densités de pixels, de l'orientation de l'écran et de la « vue snap » donnent à l'utilisateur la large souplesse d'usage qu'il peut attendre.

> Modern UI

Depuis l'arrivée de Windows Phone 7, Modern UI se veut digital, c'est-à-dire que l'on ne cherche plus à reproduire des matériaux réels (verre, bois, métal, papier...) ou des phénomènes physiques (Ombres, transparence...). Le contenu doit être au centre de la création, non sans rappeler ce qui se fait dans le monde de l'impression. Une bonne illustration de cette différence d'ADN se constate en comparant les applications de gestion des contacts d'OS X Lion [Fig.1] et Windows 8 [Fig.2].

L'une imite résolument le calepin que nous laissons à côté du téléphone, tandis que l'autre propose une vue plus abstraite lais-

sant une meilleure place au contenu.

Lors de la conception d'une application modern UI, il convient donc de laisser la meilleure place possible au contenu ou aux données de vos utilisateurs. Ainsi, L'un des piliers des « guidelines [1] » éditées par Microsoft recommande par exemple de ne pas laisser les différents contrôles et actions polluer l'espace visuel et donc de les regrouper au sein des « Application bars » qui peuvent être affichées et masquées au besoin par l'utilisateur.

Modern UI reste un cadre et n'interdit pas d'exprimer les valeurs et la charte d'une marque existante.

> Le tactile

Windows 8 apporte une solide base sur laquelle designers et développeurs peuvent se reposer lorsqu'ils conçoivent l'expérience tactile de l'application. Cela se traduit par un « langage tactile » défini ainsi : [Fig.3].

A cela, on ajoute quelques règles que certains diront « de bon sens » :

- Utilisez les doigts pour ce qu'ils savent faire. N'attendez pas d'eux une trop grande précision. Respectez une taille minimale pour les cibles d'interactions.
- Utilisez le tactile pour la navigation. Ne pas hésiter à faire de larges pages, car scroller au doigt est naturel. L'usage du semantic zoom peut enrichir le contenu en

(1) Windows Store UX guidelines : <http://msdn.microsoft.com/en-us/library/windows/apps/hh465424.aspx>

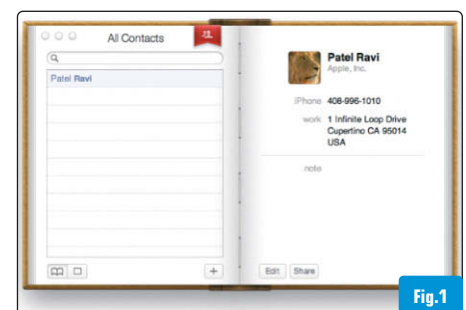


Fig.1

Address Book: OS X Lion



apportant une navigation rapide et/ou une vue différente des contenus

- Réagissez aux interactions. Chaque interaction doit en effet déclencher immédiatement quelque chose par exemple en pré-visualisant le résultat de l'action. Les contrôles comme les GridView et ListView implémentent directement des feedback cohérents pour le tap, le swipe et le drag and drop.
- Une interaction doit toujours pouvoir être annulée. Il est crucial lors de l'implémentation de gestuelles qu'elles soient réversibles afin que l'utilisateur soit en confiance et n'ait pas peur de ses mouvements.
- Autorisez autant que possible l'utilisateur à se servir d'autant de doigts qu'il le souhaite. Par exemple, si une zone est scrollable horizontalement, il est préférable qu'il puisse le faire aussi bien avec 1, 2 ou 3 doigts. En effet, les utilisateurs touchent souvent la surface avec plusieurs doigts sans même s'en rendre compte.
- Ne liez pas vos interactions au temps. Par exemple, le double tap est difficile à doser : un timing trop long générera des faux positif alors qu'un timing trop rapide sera difficile à exécuter pour certains. Autant se dispenser de cette gestuelle au profit d'une autre laissant plus de latitude à l'utilisateur.

> Fast & Fluid

Différentes études ont montré que l'expérience tactile demande une réactivité et une fluidité supérieures à celles exigées par des dispositifs de pointage tels que la souris. En effet, l'inconfort généré par un contenu qui ne se déplace pas de manière cohérente (saccade, décalage) par rapport aux doigts est beaucoup plus élevé que l'équivalent à la souris. Bien sûr, même à la souris, la performance et la fluidité sont importantes, mais avec le tactile elle a tendance à devenir primordiale, sans quoi vous risquez d'incommoder vos utilisateurs.

> Hétérogénéité des écrans et des usages

Une application tablette complète prendra en charge aussi bien le mode vertical, horizontal et si elle est pertinente, la vue « snap ». Ainsi vous prenez en compte la diversité de vos utilisateurs. Certains préféreront effectivement utiliser votre application en tenant la tablette verticalement et non horizontalement. D'autre part, Windows 8 apporte une prise en charge étendue des différentes densités de pixels. Afin d'apporter la meilleure qualité visuelle aux utilisateurs de votre application vous devez prendre en compte le comportement de Windows 8 face aux hautes densités de pixels offertes par certains équipements comme par exemple la Surface Pro. Tout comme en WPF, lorsque vous développez pour Windows 8, vous travaillez avec la notion de Device Independent Pixel (DIP). Lorsque vous donnez une taille de 20 à un élément il s'agit en réalité de 20 DIP et non 20 pixels physiques. Si votre écran est un écran présentant une très haute densité de pixels, la taille de l'élément sera augmentée à 180% c'est-à-dire qu'il fera 36 pixels physiques. A l'inverse, sur un écran basse densité, il fera simplement 20 pixels physiques [Fig.4]. En conséquence, vous devez observer les règles suivantes :

- Penser « responsive design », c'est-à-dire adapter le nombre d'éléments à l'écran naturellement en fonction de l'espace disponible
- Penser autant que possible vectoriel
- Les images bitmaps doivent, si possible, être disponibles en 3 résolutions : 100%, 140% et 180%

> Intégration au système

Les « charms » ou talismans dans la langue de Molière constituent l'un des grands piliers

de l'expérience Windows 8. Il s'agit en fait de la barre d'action que l'on fait apparaître à droite de l'écran tactile en effectuant un swipe depuis le bord. Elle permet de lancer un certain nombre d'actions de manière homogène pour l'ensemble des applications. Il s'agit de :

- La recherche
- Le partage
- Les périphériques (diffusion et impression)
- Les paramètres

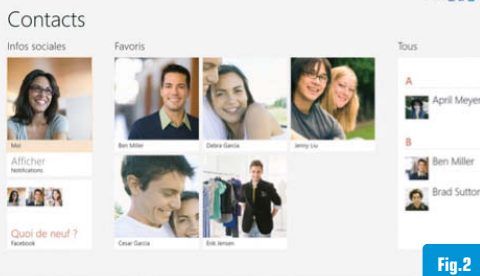
A cela vient s'ajouter la notion de contrat qui permet à votre application d'enrichir le système et les autres applications de son propre contenu. Parmi eux on notera :

- La recherche (suggestions, auto complétion, résultats)
- File Picker (l'utilisateur peut retrouver le contenu de votre application lorsqu'il utilise le « file picker »)
- Protocol (prise en charge d'uri spécifiques de type monapp://...)
- Share target : votre application peut alors recevoir le contenu partagé par une autre application

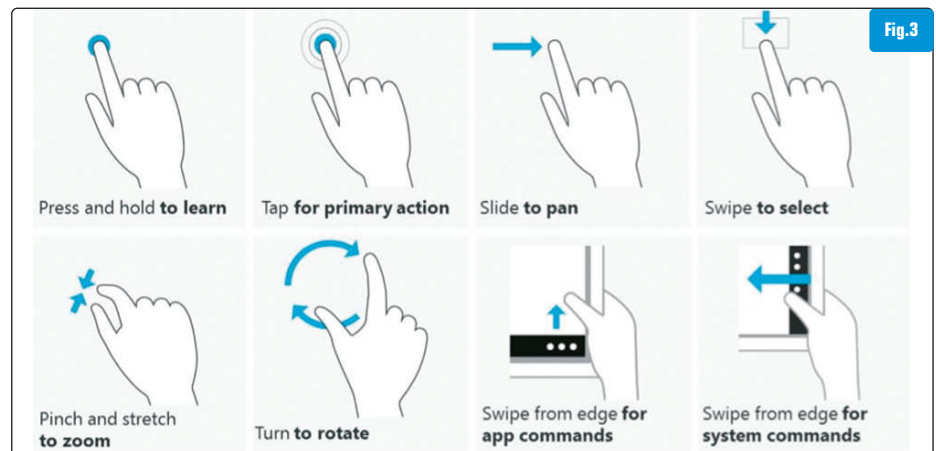
L'implémentation de ces contrats apporte de la valeur à votre application mais aussi une meilleure visibilité de votre application [Fig.5].

> Live tiles

Windows 8 introduit un nouvel écran appelé « Start screen ». A la manière de Windows Phone, il est constitué d'un panneau de tuiles dynamiques ou non. En général il est recommandé de mettre en place une tuile dynamique permettant à l'utilisateur de voir d'un seul coup d'œil les derniers contenus disponibles dans l'app, l'état de l'application ou encore le nombre d'éléments qui requièrent son attention. Il est important d'investir dans votre tuile car elle peut largement contribuer à faire revenir vos utilisateurs dans votre application.



Contacts: Windows 8



Langage tactile, la gesture

> Synchronisation des paramètres

Rien n'est plus désagréable que d'avoir à refaire ses paramétrages à la main sur chacun de ses périphériques. Windows 8 apporte une réponse technique à cette problématique au travers du Roaming. Chaque application dispose d'un petit espace de stockage synchronisé via le cloud, au sein duquel elle peut écrire les paramètres et de petits fichiers.

Il est en général attendu d'une bonne application qu'elle synchronise à minima sa configuration entre les différents appareils.

QU'EST-CE QUE WINRT

Maintenant que nous avons une vision plus claire du processus de conception d'une bonne application Windows 8 et des points qu'il faut absolument prendre en compte, regardons d'un peu plus près comment cela se passe sous le capot [Fig.6].

Avec Windows 8 vous pouvez bien sûr travailler avec les piles de développement classique (.Net, Silverlight, C++/MFC etc...). Cependant, cette approche est déconseillée car non seulement elle ne permet pas d'adresser les Windows RT (architecture processeur ARM) mais elle prive aussi des améliorations substantielles apportées par WinRT. WinRT est en fait un nouveau jeu d'API qui vient en remplacement de celles que fournissait Win32 jusqu'alors. Grâce à un mécanisme appelé « Projection », ces apis peuvent être consommées par des langages aussi différents que .net, C++/CX [2] et Javascript. Ainsi vous êtes libre, suivant vos compétences ou vos contraintes, de démarrer le développement en XAML/C++, Xaml/C# ou HTML/JS. Cette flexibilité ouvre le développement d'application Modern UI à l'ensemble des développeurs, qu'ils viennent du monde natif, .net ou Web.

WinRT offre un large nombre d'apis qui couvrent aussi bien la gestion du réseau, des données, l'UI, le modèle d'application ou l'impression. Ces apis présentent la particularité d'être pour la plupart d'entre elles asynchrones afin de ne pas bloquer l'interface utilisateur lorsqu'elles sont appelées. Les différents langages de développement apportent leur réponse pour la prise en charge de l'asynchronisme (async/await pour C# et promises pour Javascript) Certaines API classiques comme DirectX ou Media Foundation sont aussi autorisées et s'intègrent naturellement dans les applications Windows 8.

PAR OÙ COMMENCER

Il faut dans un premier temps disposer d'une machine de développement sous Windows 8. Vous devez par ailleurs créer un compte développeur [3] pour pouvoir distribuer vos applications sur le store. Ensuite, viennent s'ajouter un certain nombre d'outils dont il existe en général une version gratuite.

> Les outils

Visual Studio 2012

Visual Studio 2012, que l'on ne présente plus, vous permettra d'effectuer le développement et le débogage de vos applications. Une version express permet de se lancer à moindre coût puisqu'elle est gratuite. Cependant, les versions payantes apportent notamment des outils de profiling et de test très intéressants.

A noter qu'une mise à jour (Update 1) de Visual Studio est déjà disponible et qu'une seconde mise à jour est déjà là.

Blend

Blend est depuis l'avènement de WPF, l'outil optimal pour la réalisation des interfaces graphiques et des animations de vos applica-

tions. La dernière mouture de l'application supporte aussi bien le XAML que l'HTML.

Team Foundation Service

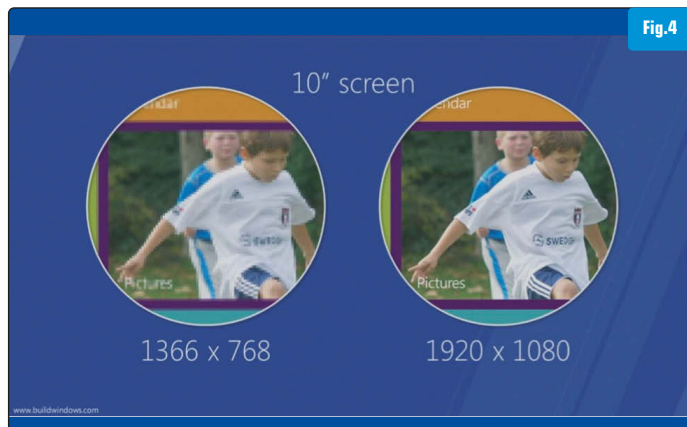
En général il vaut mieux se munir d'un bon contrôleur de source mais aussi d'un bon outil pour gérer son backlog et le suivi des développements lorsque l'on travaille en équipe. Team Foundation Service (gratuit pour les petites équipes), apporte toute la puissance de Team Foundation Server 2012 dans le Cloud.

Outils de maquettage

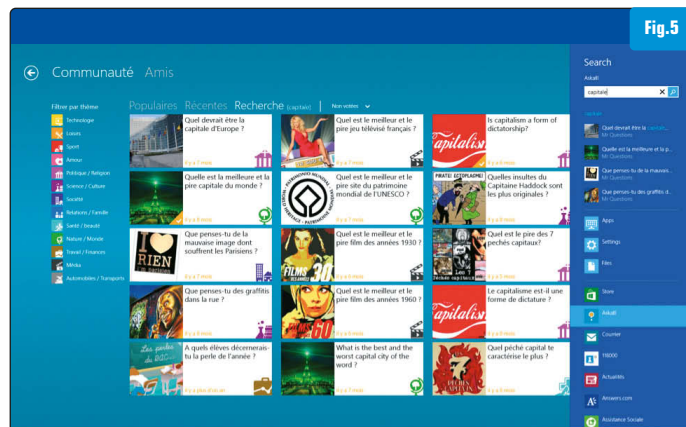
Afin de concevoir votre application vous pouvez bien sûr utiliser du papier, un crayon et une gomme (ça reste extrêmement efficace). Si le plomb vous effraie, vous pouvez tout aussi bien opter pour Balsamiq Mockups ou le nouvel outil de prototypage que Visual Studio a ajouté à Powerpoint.

> Les templates

Vous pouvez bien sûr démarrer le développement depuis un projet vide. Cependant, vous disposez aussi de différents templates qui proposent une base de développement, notamment avec différentes classes utilitaires mais aussi avec une base de navigation. Le template Grid application (Xaml) offre par exemple une base d'application proche de celle de l'application Windows Store à savoir une page d'accueil avec des éléments regroupés par catégorie, une page de catégorie avec un élément mis en avant et enfin une page de détail. Toutefois une partie de la richesse du template n'est pas directement visible. Il s'agit du contenu du dossier Common avec notamment les classes BindableBase (qui permettent d'implémenter facilement de manière maintenable INotifyPropertyChanged) ou encore LayoutAwarePage (simplifie la prise en char-



Comportement du scaling Windows 8 (Build 2011)



Intégration du contrat de recherche et prise en charge des suggestions au sein de l'application Askall



ge de la rotation de l'écran] ou Suspension-Manager (gère de manière transparente le tombstoning de l'application).

> Le manifest

Le manifest d'une application Windows 8 constitue le pilier d'une application Windows 8. Il s'agit du fichier Package.appxmanifest. C'est au sein de ce dernier que vous définissez les caractéristiques principales de votre application à savoir : son nom, ses logos, les apis qu'elle utilise (caméra, internet, etc..) et les contrats qu'elle expose.

L'onglet Capabilities est crucial. Un grand nombre d'API WinRT sont protégées. Si vous n'avez pas fait la déclaration de son usage, tous les appels échoueront avec l'erreur « Access Denied ». Cela permet au store d'indiquer avec certitude ce que l'application exploite à ses utilisateurs [Fig.7].

LES OUTILS COMPLÉMENTAIRES

> Contrôles tiers

Les éditeurs Telerik et DevExpress offrent déjà un large choix de contrôles spécialement conçus pour Windows 8. Ils peuvent offrir un gain de temps significatif lors du développement de certaines fonctionnalités. Du côté communautaire on trouvera Callisto, développé par Tim Heuer. Ce dernier offre quelques contrôles facilitant grandement le développement des applications Xaml.

> Bases de données locales

Il existe aujourd'hui deux moteurs de base de données utilisables simplement sur Windows 8. Le premier est bien connu et présente l'avantage d'être facilement portable il

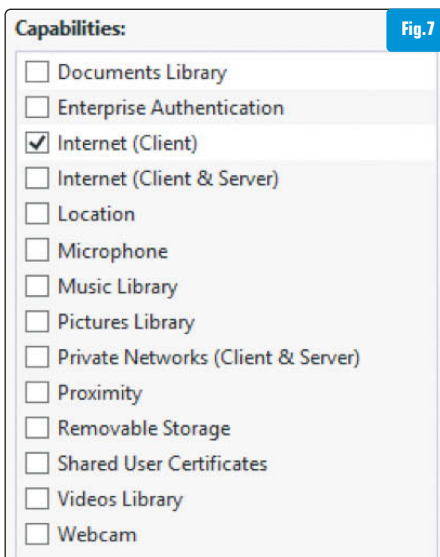
s'agit de SQLite. A noter qu'il induit une dépendance à Visual C++ et requiert donc une compilation spécifique pour chaque plateforme. L'alternative repose sur l'exploitation du moteur de base de données natif de Windows : ESENT. En C++/CX il peut être exploité directement. Le projet codeplex ManagedEsent permet de l'utiliser depuis du code .net

> Fiddler

Fiddler est un débogueur web qui agit comme un proxy et permet d'analyser avec une très grande efficacité les communications http/https de vos sites et applications. La dernière version de l'outil prend aussi en charge les applications Modern UI. Il s'agit d'un must have dans votre boîte à outils.

LE STORE

Le store Windows 8 se rapproche de ce que l'on connaît sur d'autres plateformes. La publication d'une application est un processus qui inclut des vérifications de sécurité et de qualité automatiques ainsi qu'une revue manuelle de l'application. Le processus de publication prend en général plusieurs jours. Microsoft publie les outils de tests automatiques qu'ils utilisent. Ils constituent le WACK (Windows App Certification Kit). Il est nécessaire de les exécuter avant toute publication et de corriger toutes les erreurs qui seraient remontées. Au cours de cette phase, votre application subira un certain nombre de tests fonctionnels afin de vérifier différents points tels que, sa raison d'être, le respect des guidelines, le comportement sur des réseaux instables etc... Testez donc bien vos applications. Une des raisons les plus courantes du rejet d'une application par les



Windows 8 Capabilities

équipes du store est l'absence de politique de confidentialité. Lorsque votre application accède à Internet, vous êtes tenu de fournir un tel document. Ce dernier doit être accessible depuis les paramètres de votre application mais vous devez aussi fournir un lien vers ces dernières lors de la publication de votre application sur le portail. Ce lien sera disponible depuis la page de description de votre application.

CONCLUSION

La plateforme Windows 8 a déjà une certaine maturité. Les outils sont riches et vont au-delà de ce que Microsoft lui-même fournit. Visual Studio suit un rythme de mises à jour soutenu (environ tous les 6 mois) afin d'offrir aux utilisateurs un environnement toujours meilleur. Le store connaît une croissance importante et on ne saurait trop rappeler l'importance de la qualité de la conception d'une application pour se distinguer dans la masse. Toutefois, de grands changements semblent encore à venir avec l'approche de Windows 8.1, qui semble marquer un nouveau pas dans la stratégie de Microsoft mais aussi vers l'uniformisation des outils de développement desktop, tablette et smartphone.

Cyprien Autexier

Consultant chez Wygwam et développeur chez Askall

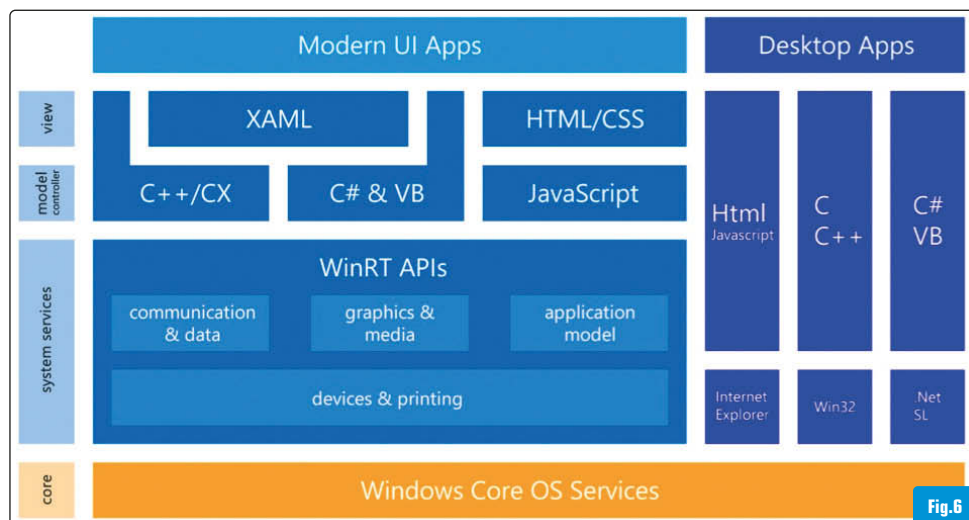


Fig.6

Piles de développements pour Windows 8

(2) C++/CX diffère de C++/CLI car il n'est pas managé (il ne repose pas sur la CLR).

(3) Création d'un compte développeur : <https://appdev.microsoft.com/StorePortals/en-us/Account/Signup/Start/>

Créer des applications hybrides avec PhoneGap



Aujourd'hui, les développeurs cherchent à réduire le temps de déploiement de leurs applications mobiles et ainsi à prendre un avantage sur leurs concurrents. L'objectif de cet article est de vous faire découvrir une approche alternative pour booster vos développements d'application sur tablette : l'approche Hybride. Ou comment tirer parti de la puissance de HTML5 et Javascript pour créer des applications natives.

La difficulté pour l'éditeur qui développe des applications mobiles est de devoir gérer une multitude de plateformes (différentes tailles, résolutions, densités de pixels, matériels, mémoire et performance) ainsi qu'une multitude d'OS (les plus connus étant Android, iOS, BlackBerry, Windows Phone). L'éditeur doit par ailleurs, pour des raisons financières, arbitrer entre deux alternatives : offrir un service aux fonctionnalités réduites au plus grand nombre d'utilisateurs ou permettre à une cible restreinte d'utilisateurs de vivre une expérience optimale. Face à ce dilemme, deux stratégies. Petit retour sur les avantages et inconvénients de chaque approche...

> Natif vs HTML Application native

L'application est développée dans un langage spécifique à une plateforme, Java pour Android, Objective-C pour iOS, etc.

Le code est compilé spécifiquement à la plateforme et tire parti du SDK propre au constructeur.

Avantages	Inconvénients
Interface et expérience utilisateur optimales.	Pas de mutualisation des développements sur les différentes plateformes.
Infrastructures de distribution et coûts de publicité réduits grâce aux Stores.	Développeurs avec compétences multiples (Java, Objective-C...) difficiles à trouver.
Visibilité accrue.	Coûts et temps de développement liés au nombre de plateformes ciblées.
	Modération des applications avant diffusion par le Store

Application Web

L'application est développée en HTML/Javascript/CSS pour toutes les plateformes (y compris Desktop) et s'exécute dans le navigateur web de la tablette.

Avantages	Inconvénients
Un seul langage : HTML.	Visibilité réduite (pas de Store).
Une cible d'utilisateurs la plus large possible.	Nécessité d'assurer la distribution.
Mutualisation et capitalisation des développements.	Interface et expérience utilisateur moins riches.
Mise en ligne immédiate de l'application	Pas de notification : c'est l'utilisateur qui sollicite votre marque ou votre produit en allant sur votre site.

> Hybride, une alternative séduisante

Une alternative est de mixer les deux solutions en proposant ce qu'on appelle une application hybride. Une application hybride c'est quoi ?

- Un cœur Web en HTML.
- Un conteneur d'application natif, mettant à disposition une WebView.

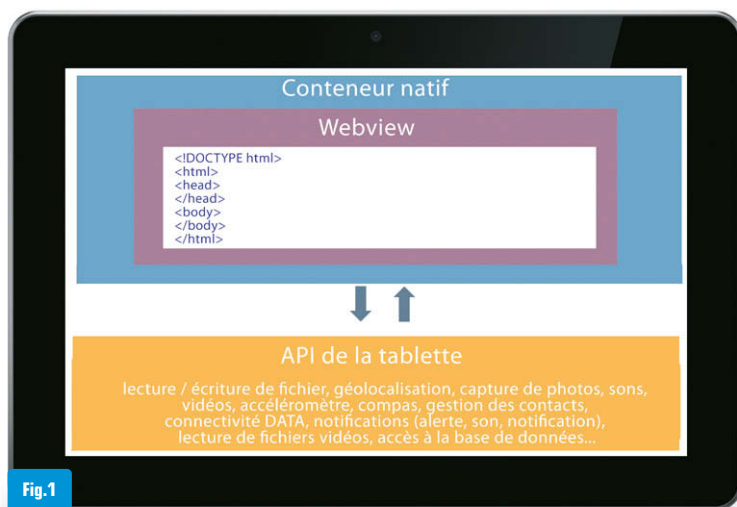


Fig.1

- Une API qui fait office de passerelle entre le cœur de l'application et les spécificités de la plateforme [Fig.1].

L'intérêt est de mettre à disposition une application HTML, et de la proposer à différents Stores sans coût supplémentaire de développement. On réduit le nombre de compétences nécessaires au développement puisque l'on a moins de spécificités techniques liées aux différents environnements. Il est par ailleurs possible de réutiliser le code HTML d'une application web existante ; attention cependant à ne pas négliger l'expérience utilisateur en lui proposant un simple site web dans une application.

PHONEGAP, C'EST QUOI ?

PhoneGap crée des applications hybrides : Android, iOS, BlackBerry, Symbian, WebOS, Windows Phone 7/8, Windows 8, Bada et Tizen. L'intérêt de PhoneGap est d'aller plus loin et de proposer un ensemble d'API permettant d'exploiter les fonctionnalités offertes par le matériel, ce qui n'est pas encore nativement disponible avec HTML5. L'API PhoneGap permet ainsi d'accéder à l'accéléromètre, la boussole, la géolocalisation, le stockage hors-ligne, la caméra, l'appareil photo, le micro, les notifications systèmes, les contacts, les informations et événements systèmes, les fichiers et splashscreen.

PhoneGap ou Cordova ?

Il est fréquent de confondre PhoneGap et Cordova. Pour la petite histoire, PhoneGap a été développé initialement par l'éditeur Nitobi, racheté par Adobe en 2011, qui décida par la même occasion de reverser PhoneGap à la fondation Apache. La plateforme devient Open Source, mais ne peut pas garder le même nom et devient



Apache Cordova (<http://cordova.apache.org/>) Adobe continue cependant à travailler sur PhoneGap qui devient donc une distribution de Cordova qui devrait intégrer ultérieurement des fonctionnalités et services complémentaires. Ne soyez donc pas surpris si vous voyez Cordova apparaître dans vos sources PhoneGap !

> Développement et Construction

PhoneGap ne dispose pas de son propre IDE. Il s'agit avant tout de sources au format HTML, Javascript. Le développeur peut conserver son IDE de développement Web habituel. Pour la construction de l'application, deux approches sont possibles :

- Depuis la console (pour Android, iOS, BlackBerry et Windows Phone), permet de construire rapidement l'application en ligne de commande.
- Depuis l'IDE (toutes plateformes), permet de tirer parti de l'interface graphique de développement pour construire l'application native. Pour ce faire, il faut télécharger PhoneGap (<http://phonegap.com/download/>) et suivre les étapes pour la création du projet. Quelques exemples d'environnements de base nécessaires à PhoneGap selon les plateformes :

• Android

Java, Ant, Eclipse Classic, Android SDK, plugin Eclipse ADT

• iOS

XCode 4.5, iOS 6 SDK

• Windows Phone 8

Visual Studio Express, Windows Phone SDK

> Créer votre projet

Dans cette partie nous allons nous concentrer sur la création d'un nouveau projet pour la plateforme Android. Il faut commencer par créer l'environnement de développement à base d'Eclipse Classic (<http://www.eclipse.org/downloads>), installer le SDK Android (<http://developer.android.com/sdk>) sous C:\Development\android-sdk-windows par exemple, ainsi que le plugin Eclipse ADT (<http://developer.android.com/tools/sdk/eclipse-adt.html>).

La variable d'environnement PATH doit contenir le chemin vers les tools et platform-tools du SDK :

```
C:\Development\android-sdk-windows\platform-tools
C:\Development\android-sdk-windows\tools
```

Pour la création de projet PhoneGap pour Android, se positionner dans le répertoire <PHONEGAP>\lib\android\bin (<PHONEGAP> correspond au répertoire d'installation de PhoneGap), puis lancer la commande suivante pour la création de projet :

```
create <project_path> <package_name> <project_name>
```

<project_path> chemin du nouveau projet PhoneGap

<package_name> nom de package de l'application. Par exemple com.masociete.MonApplication (pas d'espaces ni de tirets)

<project_name> nom du projet, par exemple MonApplication (pas d'espaces ni tirets)

■ **Note :** Le projet PhoneGap doit être créé hors du workspace Eclipse.

Lancer ensuite Eclipse pour importer le projet, File -> New à Other..., New... « Android Project from existing code ».

Sélectionner le répertoire <project_path> correspondant au projet PhoneGap créé précédemment (le projet doit être copié dans le workspace). Le projet est maintenant ouvert dans Eclipse et le

répertoire «assets/www» Contient les ressources Web nécessaires à l'application PhoneGap.

> Intégrer jQueryMobile

jQuery Mobile (<http://jquerymobile.com/>) est une librairie qui propose des composants optimisés pour mobiles et tablettes (effet de transition entre les pages, widget avancés de présentation ; Collapsible, Reflow table, NavBar...) particulièrement optimisée pour fonctionner avec PhoneGap. Voici un exemple de page :

```
<body>
  <div data-role="page" data-theme="a" class="page-map">
    <div data-role="header" data-position="inline">
      <h1>Démo PhoneGap!</h1>
    </div>
    <div data-role="content" data-theme="a">
      <div class="content-primary">
        <h2>Titre</h2>
        <p>test</p>
      </div>
    </div>
    <div data-role="footer" data-theme="a"></div>
  </div>
  <script type="text/javascript"
    src="jquery-1.9.1.min.js"></script>
  <script type="text/javascript"
    src="jquery.mobile.min.js"></script>
</body>
```

Grâce au ThemeRoller proposé par jQueryMobile (<http://jquerymobile.com/themeroller/>), il est simple de créer plusieurs thèmes pour une application PhoneGap. Dans les exemples de code ci-dessous, nous tirerons parti de jQuery pour simplifier la lecture du code.

Intégrer PhoneGap à un projet Web existant

Si vous souhaitez intégrer PhoneGap à un projet Web existant, il est important de vérifier la disponibilité des API afin d'éviter toute erreur Javascript au moment de l'exécution. En effet, les API PhoneGap sont spécifiques et ne peuvent donc pas s'exécuter dans un navigateur « classique ». Il est donc nécessaire de vérifier si les API sont disponibles au moment de l'exécution. Une bonne pratique est de tester la variable userAgent :

```
function isPhoneGap() {
  return navigator.userAgent.match(/(iPhone|iPod|iPad|Android|BlackBerry)/)
}
```

Pour que les API PhoneGap puissent être accessibles, il est nécessaire d'ajouter la librairie Javascript Cordova. Il est préférable de faire un chargement conditionnel de la librairie. Ce chargement est pris en compte par jQuery (fonction \$.getScript()) dans notre exemple :

```
$(function(){
  if( isPhoneGap() ) {
    // je suis sur Cordova
    $.getScript("cordova-2.7.0.js", function() {
      // je m'abonne à l'événement deviceready
    })
  }
})
```

```
document.addEventListener(«deviceready»,
                        onDeviceReady, false);
});
}
else {
    // je suis sur un navigateur
}
});
```

Les API doivent être intégralement chargées avant d'être utilisées. Il faut écouter l'événement DeviceReady et l'associer à une fonction qui effectuera les traitements.

```
function onDeviceReady () {
    // l'API est prête
}
```

La fonction onDeviceReady() est le callback appelé lorsque l'API est disponible. C'est dans ce callback que les traitements sont réalisés.

> Découverte des API

PhoneGap propose un ensemble d'API pour manipuler les fonctionnalités de bas niveau non disponibles dans HTML5. Quelles sont-elles et comment y accéder ?

Notifications

```
navigator.notification.alert('message', null, 'titre', 'OK');
```

Notification sonore

```
navigator.notification.beeper(2);
```

Notification haptique

```
navigator.notification.vibrate(0);
```

Accéléromètre

L'API de l'accéléromètre est disponible sur l'élément navigator.accelerometer. La fonction watchAcceleration() est utilisée pour scanner l'orientation de l'accéléromètre. Il est nécessaire de lui indiquer une fonction de callback appelée à une fréquence donnée (indiquée par l'option options.frequency). Il est également possible d'interagir avec plusieurs accéléromètres. La variable *accelerationId* contient l'identifiant de l'instance en cours de surveillance.

Code HTML associé

```
<div id=«x»>0</div>
<div id=«y»>0</div>
<div id=«z»>0</div>

<button onclick="launchAcceleration();">launch !</button>
<button onclick="stopAcceleration();">stop !</button>
```

Code Javascript

```
var accelerationId = null;

function launchAcceleration () {
    var options = {};
    options.frequency = 1000;
    accelerationId = navigator.accelerometer.watchAcceleration(
        updateAcceleration, fail, options);
};
```

```
function roundNumber(num) {
    return Math.round(num * Math.pow(10, 3)) / Math.pow(10, 3);
}
```

```
function updateAcceleration(a) {
    $(«#x»).html(roundNumber(a.x));
    $(«#y»).html(roundNumber(a.y));
    $(«#z»).html(roundNumber(a.z));
}
```

```
function fail(ex) {
    alert(«accel fail (« + ex.name + «: « + ex.message + «)»);
}
```

Pour arrêter la surveillance de l'accéléromètre, il est nécessaire d'utiliser la variable *accelerationId* comme paramètre à la fonction clearWatch().

```
function stopAcceleration() {
    navigator.accelerometer.clearWatch(accelerationId);
    updateAcceleration({ x : «», y : «», z : «» });
    accelerationId = null;
}
```

Photo

La prise de photo n'est pas intégrée directement dans l'application, mais PhoneGap délègue la prise de photo à l'application native. La photo sera affichée dans une balise

Code HTML associé

```
<img id=«myPicture» height=«460»/>
```

Code Javascript

```
function takePicture() {
    navigator.camera.getPicture(dumpPicture, fail, {quality:25,
    sourceType:Camera.PictureSourceType.CAMERA,
    encodingType:Camera.EncodingType.JPEG,
    destinationType: Camera.DestinationType.DATA_URL} );}
On remarque que le type de destination est DATA_URL. Celui-ci indique que l'image est encodée en Base64. La fonction dumpPicture() est appelée lorsque la photo est prise. Le contenu de l'image imageData est placé dans l'attribut src de myPicture en spécifiant le type de donnée.
```

```
function dumpPicture(imageData) {
    $(«#myPicture»).attr('src', «data:image/jpeg;base64,» + imageData);
}
```

Géolocalisation

La fonction de géolocalisation est plus discutable car les navigateurs Web compatibles HTML5 intègrent cette fonctionnalité. Toutefois, PhoneGap permet de l'uniformiser puisque tous ne l'implémentent pas de la même manière. La fonction getCurrentPosition() est disponible sur l'objet navigator.geolocation pour acquérir les coordonnées géographiques.

Code HTML associé

```
<button onclick="getPosition()">Position</button>
```

Code Javascript

```
function getPosition() {
    navigator.geolocation.getCurrentPosition(updateLocation, fail);
}
```



```

}

function updateLocation(position) {
    $('#geolat').html(position.coords.latitude)
    $('#geolong').html(position.coords.longitude)
}

```

> Déployer et distribuer

Le déploiement peut se faire comme une application native, à savoir soit sur un émulateur, soit directement sur une tablette. Vérifiez que la tablette est connectée en USB et reconnue par ADT. Pour plus de détails sur ce point n'hésitez pas à aller voir sur (<http://d.android.com/tools/extras/oem-usb.html>). Dans Eclipse, sélectionner le projet, puis lancer > Run as « Android Application ». L'application s'exécute sur la tablette. Lors de cette étape, l'application Android native est créée (APK) et les ressources (assets) sont intégrées à l'application. Pour la distribution, il n'y a rien de particulier à PhoneGap. Le processus est identique aux applications natives, à savoir : disposer des certificats, avoir accès à un compte développeur avant de pouvoir déposer sur les stores dédiés (Apple Store, Google Play, BlackBerry...). Apple propose différentes formules pour générer ce certificat (99\$/an pour une licence développeur et 299\$/an pour une licence entreprise) tandis que le compte développeur Google Play se limite à 25 \$.

> PhoneGap:Build, tirer parti du Cloud pour construire sur différentes plateformes

Toutes ces étapes d'installation et de configuration peuvent s'avérer longues, notamment si vous souhaitez distribuer votre application sur de nombreuses plateformes. De surcroît il faudra disposer des connaissances sur les différents environnements de développement pour assurer une construction correcte des applications natives. Ceci peut s'avérer un frein pour certains développeurs.

C'est précisément là qu'intervient PhoneGap:Build (<https://build.phonegap.com>), un service en ligne qui permet de déléguer le processus de construction [Fig.2]. PhoneGap:Build permet de s'affranchir de la gestion des SDK, des IDE et des processus de construction. PhoneGap:Build traite ces points pour les développeurs dans le Cloud.

Adobe propose officiellement ce service depuis septembre 2012 : un service gratuit pour une seule application privée – idéal pour tester la plateforme – et un service payant à 9,99\$ pour construire jusqu'à 25 applications privées. Il est possible de coupler son compte à celui de GitHub (<https://github.com/>) et ainsi faciliter la récupération des sources dans PhoneGap:Build. L'idée est de déposer un ZIP contenant les ressources web (HTML, images, Javascript, CSS, fichiers de configura-

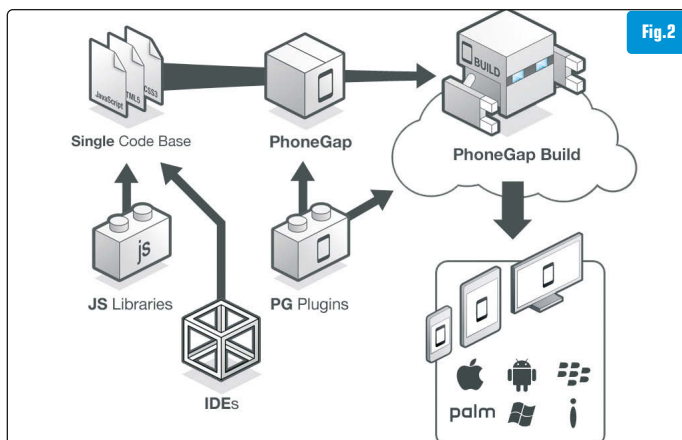


Fig.2

tion) auprès du serveur PhoneGap:Build et celui-ci va construire les applications sur les plateformes. A l'heure de l'écriture de cet article, PhoneGap:Build est compatible avec l'API phoneGap 2.5 et permet la création d'applications Android, iOS, Windows Phone, BlackBerry, Symbian et HP WebOS.

Note : il existe un décalage entre la dernière version de PhoneGap (2.7) et la version disponible de PhoneGap:Build (2.5) : ne vous jetez pas sur la toute dernière version de PhoneGap si vous souhaitez bénéficier de Build ! [Fig.3]. Il est possible d'utiliser ses certificats pour générer les applications (Edit Account > Signing Keys). Il est à noter qu'il est impératif d'avoir une clé de licence iOS pour pouvoir générer le package pour les terminaux Apple. Pour le reste, PhoneGap:Build, se charge de générer chaque package par plateforme. L'application une fois construite, peut être récupérée grâce au QRCode disponible sur le site de PhoneGap :Build et ainsi permettre de débiter les tests. Il ne faut pas oublier que si le code est commun à l'ensemble des plateformes, les tests quant à eux doivent être réalisés sur chaque environnement pour vérifier le comportement de l'application et s'assurer de l'absence de régression. Le coût des tests n'est donc pas celui d'une application mais bien celui du nombre d'environnements ciblés.

> Conclusion

Il ne faut pas voir PhoneGap comme une solution « magique » ou universelle pour tous les développements mobiles multi-plateformes. Cette solution constitue une alternative intéressante aux développements natifs pour en diminuer les coûts et le temps de mise à disposition. Il faut également garder à l'esprit que l'architecture PhoneGap peut amener les utilisateurs à rencontrer des performances amoindries par rapport aux applications natives. Il faudra, par ailleurs, accepter que l'expérience utilisateur soit moins intégrée à la plateforme. Il est essentiel de prendre en compte la finalité même de l'application. Le choix d'une solution hybride dépend donc des contraintes et priorités de l'éditeur, à savoir son budget, la diversité des terminaux, l'usage de l'application et la richesse de l'expérience utilisateurs qu'il souhaite proposer. PhoneGap et PhoneGap:Build s'avèrent être une solution intéressante pour produire rapidement des applications simples ou des prototypes destinés à valider un concept.

Florent Dupont (fdupont@sodifrance.fr)

Architecte logiciel – Expert mobilité chez Sodifrance

Anne-Lise Agoyé (alagoye@sodifrance.fr)

Chef de projet Web - Mobilité chez Sodifrance

Site : <http://www.sodifrance.fr> - Blog : <http://javateam.sodifrance.fr>

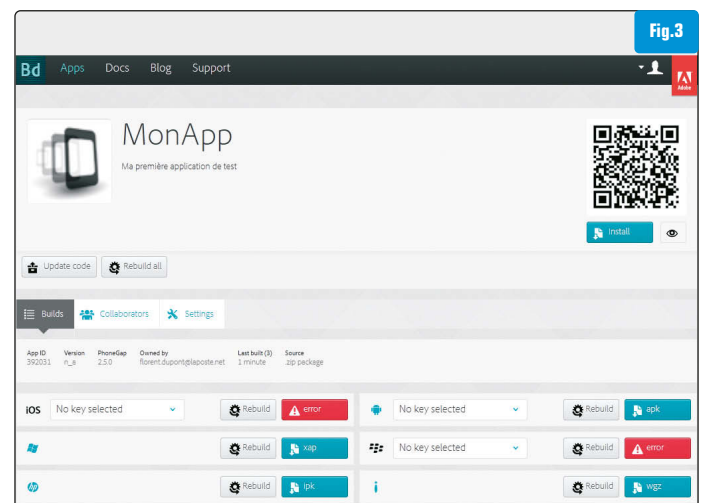


Fig.3

42 : born to code

Xavier Niel, le patron de Free, veut révolutionner l'école informatique et l'enseignement du développement et de la technologie, chez les jeunes. Ce projet a pour nom : 42. Pour ce faire, il s'appuie sur une solide équipe dont plusieurs anciens de l'Epitech, l'école qui a représenté une rupture dans la manière de former les étudiants en informatique. 42 s'inscrit dans la même démarche. Deux points de vue sont donc possibles : c'est une révolution bienvenue, ou une absence de réelle nouveauté ?

Former 1000 informaticiens, avec priorité au code

Xavier Niel définit dans la présentation de l'école très clairement l'origine du projet et une de ses motivations : « Aujourd'hui, le système français ne marche pas. Il est coincé entre, d'une part l'Université qui propose une formation pas toujours adaptée aux besoins des entreprises mais qui est gratuite et accessible au plus grand nombre, et d'autre part les écoles privées, chères, dont la formation est assez qualitative mais laisse sur le côté de la route le plus grand nombre de talents, voire de génies, que nous pourrions trouver en France ». Le système éducatif actuel n'aide pas à faire aimer la technologie, ni le développement. C'est un réel problème depuis de nombreuses années. La réponse proposée est une pédagogie plus moderne (selon 42).

« L'objectif est de former 1 000 informaticiens, avec priorité au code, au développement, à la technologie », s'enthousiasme



Nicolas Sadirac,
Directeur Général de 42

et (tout) le reste.

Ouverture à tout le monde et gratuité

Pour l'équipe de l'école, 42 doit représenter l'état de l'art de plus de vingt ans d'expérience et de pédagogie informatique. Ouverte à tous, car aucun diplôme n'est requis pour rentrer, 42 propose la gratuité du cursus, alors que la scolarité dans une école informatique coûte souvent plusieurs milliers d'euros. Les autres principes : la passion du code et des technologies, un esprit commu-



Hall d'entrée de 42

nautaire (le but est que les élèves travaillent ensemble, collaborent, échangent, s'entraident) et un état d'esprit proche de la réalité de la vie du développeur avec des projets réels, tout en favorisant les initiatives des élèves. Les locaux seront ouverts 24h/24 et 7j/7 : pour Nicolas Sadirac, il ne doit pas exister de contraintes d'horaires quand on est passionné.

Un cursus en 3 ans : des projets, rien que des projets !

Un cursus classique dans sa durée : 3 ans. Ils sont nommés 42.1, 42.2, 42.3. La pédagogie voulue par l'école est simple : faire du projet, du projet et encore du projet ! Le but, quand on rentre à 42, n'est pas de suivre un cursus pour apprendre l'informatique, les langages. Bref éviter le décalage entre la théorie enseignée et la réalité de la vraie vie. « On a du C, C++ (dans le cursus). On ne l'enseigne pas », poursuit Nicolas Sadirac. Et quand on regarde de plus près le programme des trois années, on constate effectivement l'absence de toute formation aux langages et à la théorie liée, même si plusieurs modules sont orientés langages : C++ (Piscine C++ équivalent de 4 crédits par

“Les écoles privées sont chères (et) laissent sur le côté de la route le plus grand nombre de talents”

Xavier Niel

exemple), Java. Les modules proposent de nombreuses thématiques : OpenGL, la sécurité, l'algorithmie, la culture informatique, Unix, l'IA, systèmes et réseaux, open source / libre, programmation fonctionnelle, le kernel, Android, etc.

Quand on regarde l'exemple du programme, le cursus apporte donc une compétence large mais toujours pratique. On ne vient pas à 42 pour apprendre, mais pour pratiquer. Le programme prévu est très dense et les étudiants seront soumis à une forte pression s'ils veulent réussir et suivre la cadence. En 3 ans, il faudra réaliser et participer à 67 mini-projets et 49 projets ! 42 joue aussi la communauté, les échanges, l'émulation entre les étudiants et les groupes. 42 insiste sur le fait qu'il n'y a pas de cours magistraux...

Mais le programme est-il si différent des autres écoles d'informatique ? Pas sûr.

La motivation au recrutement des promotions sera un des enjeux, au-delà des compétences techniques même si aucun pré-requis n'est imposé. Il faut avoir entre 18 et 30 ans ou avoir son bac (diplôme non obligatoire). « Notre objectif est de former des professionnels pour l'industrie », précise N. Sadirac.

Un cursus ++ : 42.4 et 42.5

Pour les étudiants ayant réussi les 3 années normales, 2 années supplémentaires sont possibles. Deux thèmes y seront abordés : l'expertise industrielle et l'entrepreneuriat dans l'innovation. Objectif : comment monter une société innovante ? Quelles innovations ? Comment structurer un projet ? Comment le vendre et avoir déjà un business plan ?

Rentrée scolaire : octobre 2013

Depuis le printemps, les inscrits se multiplient. Le processus est simple :

- inscription en ligne et réponse à des mini-jeux directement en ligne
- une première sélection est faite suite au questionnaire. Il ne restera plus que 4 000 candidats
- durant l'été : les admissibles subiront une épreuve intensive alliant le stress, les défis et la technique. C'est la fameuse « piscine », se déroulant sur un mois.
- la « piscine » déterminera les 1 000 étudiants qui pourront former la première promotion à partir de l'automne 2013.

Chaque promotion comptera 1 000 étudiants. Une cantine (payante) sera disponible dans l'école ultérieurement.

Diplôme : pas reconnu ? Et alors ?

« C'est une chance de ne pas faire reconnaître le diplôme. Notre secteur recrute sur les compétences pas sur le diplôme », martèle Nicolas Sadirac. Cependant, si l'expé-



Concept de la « piscine ».

rience et les compétences peuvent faciliter l'embauche, le diplôme demeure un sésame pour beaucoup d'entreprises. 42 sanctionnera le cursus par son propre diplôme. Combien d'étudiants, sur les 1000, auront-ils la joie de le recevoir au bout des 3 ans ?

Le fait que l'école ne fonctionne pas encore et qu'aucune promotion n'en soit sortie,

Epitech et 42, frères ennemis ?



Fabrice Bardèche,
vice-président exécutif
du groupe Ionis

La définition des frères ennemis, à la fois très proches et souvent en conflit, pourrait s'appliquer au rapport entre 42 et Epitech, école d'informatique du groupe Ionis. Force est de constater que Xavier Niel a pioché – sans vergogne ? – dans les équipes dirigeantes et d'encadrement d'Epitech pour bâtir 42. Dans un communiqué, le groupe Ionis « *salue le projet de Xavier Niel, largement inspiré de ses écoles. Nous sommes heureux de constater que les principes fondamentaux de la formation 42 ont été créés et mis en place au sein de l'EPITA et d'Epitech depuis très longtemps et qu'ils sont repris dans ce projet, reconnaissance de la modernité pédagogique de nos écoles.* » **Fabrice Bardèche**, vice-président exécutif du groupe Ionis, indique : « *Comme la structure d'Epitech est assez large, ceux qui sont partis pour créer 42 ont tous été remplacés. Les dirigeants de 42 ont emporté une méthode avec eux. Je trouve que 42 se rapproche, dans l'esprit, de notre école en alternance, ETNA.* »

L'ETNA, du groupe Ionis, proposera néanmoins, dès la rentrée 2013, un cursus en cinq ans, qui mène à un titre homologué niveau I par l'Etat, enregistré au Répertoire National des Certifications Professionnelles (RNCP) : architecte système réseaux et sécurité ou architecte logiciel développement d'applications. Bien sûr, 42 étant une toute nouvelle école, son diplôme sera à l'origine une certification de l'école sans valeur nationale. Aussi voit-il plus en 42 « *un concurrent de BTS ou IUT technologiques, avec toutefois une orientation très pratique et un encadrement privé.* »

Fabrice Bardèche est critique envers la méthode de sélection de 42 : 1000 étudiants de la première promotion, avec quatre sessions d'un mois d'immersion/sélection de 1000 jeunes chacune, dont sortiront 250 élus. Pour le dirigeant d'Epitech, la méthode est « *brutale, car elle met en situation d'échec les trois quarts des jeunes. Et inefficace, car certains doivent travailler pendant au moins deux ou trois mois avant de commencer à faire leurs preuves.* » Et de comparer avec les écoles du groupe Ionis : « *quand on accepte un étudiant, on pense qu'il va réussir et on va l'aider à réussir.* »

A noter, ce clin d'œil amusant dans la brochure de présentation 2013/2014 de l'Epitech : dans la page intitulée « *Vous leur trouvez quoi aux Epitech ?* » où des cadres et dirigeants expliquent pourquoi ils embauchent des diplômés d'Epitech, l'une des trois personnes interrogées est... Xavier Niel ! Le fondateur de Free y déclare : « *Un nouvel univers se constitue, évolue et participe à la croissance mondiale : le numérique. Ses principales sources de valeur ajoutée sont l'innovation et l'expertise technique, c'est là que se retrouvent Free et Epitech. Nous avons beaucoup de valeurs en commun : la volonté de faire bouger les lignes, la conviction que la technologie peut participer à un monde meilleur, le goût du progrès, la passion et le pragmatisme. Nous recrutons avec plaisir des étudiants d'Epitech, lesquels trouvent naturellement chez Free un terrain idéal à l'expression de leur passion.* » Remplacer étudiants par cadres, et cela convient aussi ! Savoureux, non ?

Christine Calais

interdit toute appréciation sur le niveau et la qualité du cursus et donc, au final, du niveau des étudiants à la sortie. On ne pourra juger que dans 3 ans. Malgré tout, saluons cette initiative qui a au moins le mérite de faire parler des développeurs !

La 1re promotion déjà embauchée... ou presque

Ametix a annoncé dès mars 2013 son intention de faire une offre d'embauche aux mille

premiers diplômés de 42, avec un salaire de base de 45 000 € brut (à voir si ce montant sera systématiquement pratiqué car il est au-dessus des salaires juniors actuels). Ametix est spécialisé dans le conseil en recrutement pour les start-ups et les grandes entreprises. La société a l'ambition de trouver une place à ces futurs diplômés dans ces entreprises. Résultat dans 3 ans...

François Tonic

EPITA, Epitech, ETNA : l'éducation en mode projet

Les écoles d'informatique du groupe Ionis se réclament d'une pédagogie par projet. Le groupe a une ambition internationale dans un proche avenir, souhaitant ouvrir des campus à l'étranger.

« **N**otre image de marque repose sur le pragmatisme et la pédagogie par projet, met en exergue Fabrice Bardèche, vice-président exécutif du groupe Ionis. *Le projet est à la source de la formation. Chez nous, ce n'est pas 'j'applique un cours dans un projet', mais 'j'apprends directement à travers le projet'.* » Le groupe Ionis compte vingt écoles dans l'enseignement supérieur, dont les écoles d'informatique suivantes : l'école d'ingénieurs EPITA, créée en 1984 et reprise par le groupe en 1994, Epitech, fondée en 1999, et l'ETNA, école en alternance créée en 2004. Sup'Internet est dédiée aux métiers de l'Internet. Le modèle de l'enseignement académique tel qu'il existe encore dans certaines universités n'y est pas de mise.

Certes, l'Epita et l'Epitech comptent plusieurs laboratoires de recherche. Un réseau de fibre optique à très haut débit (1,5 Gb/seconde) relie toutes les écoles du groupe. Mais ce qui les distingue est à chercher ailleurs.

Cultiver son jardin

Fabrice Bardèche a fondé Epitech en 1999 pour les passionnés d'informatique. « *L'informatique repose sur des langages et la compréhension de la logique, qui est à la portée de l'autoformation. Nous apprenons à nos étudiants à apprendre seuls, à chercher par eux-mêmes l'information et des solutions.* » Bref, ils apprennent à travailler ! Comme disait Voltaire dans Candide, « *le travail éloigne de nous trois grands maux: l'ennui, le*

vice, le besoin. Il faut cultiver notre jardin. ». Deux exemples : le « labt TXT » où des jeunes produisent des textes leur permettant d'arriver à un « *niveau de civilité littéraire* ». Les cours d'anglais séchés par des étudiants qui « *savent lire mais pas communiquer en anglais* » ont été remplacés par des séances d'échanges sur différents thèmes avec de jeunes tutrices anglophones, étonnamment plus appréciées par les mâles qui forment l'écrasante majorité des élèves des écoles informatiques. C'est le temps de connexion au réseau qui sert d'indicateur de présence aux élèves, dont la seule obligation est de passer des projets.

Plongée dans la Piscine de l'Epitech

Au début de leur cursus à l'EPITECH, les bizuths font le grand plongeon dans la « Piscine ». L'immersion en apnée dure trois semaines, avec un rythme de travail intensif pour donner les bases permettant de suivre leurs études par la suite, et les bons réflexes. Exercices, travaux pratiques encadrés sont notamment au programme d'une pédagogie concrète.

En première année, les étudiants sont amenés à participer à quarante projets ou mini-projets. Les rendus sont centralisés pour tous les étudiants dans les douze Epitech de France, au même moment, ce qui crée collaboration et émulation dans la promotion. Ils sont soumis à un jury composé de professionnels.

En deuxième année, les étudiants mènent de



EPITA, salle machines

nouveaux projets, plus avancés. Qui a dit qu'on ne bossait pas dans les écoles d'informatique ?

Aux petits soins avec les entreprises

Ce que veulent les entreprises, ce sont des jeunes diplômés capables de travailler et de s'intégrer rapidement, ce que doit amener la pédagogie par projet. Pour Gabriel Kepélian, responsable Recherche et développement d'Atos qui a recruté plusieurs diplômés d'Epitech, « *les étudiants possèdent un très bon bagage technique et surtout une vraie capacité à travailler. Ils savent aller au fond des choses, face à un problème.* » Constance de Rotalier, responsable des ressources humaines de 1000mercis, spécialiste de la publicité et du marketing interactifs, qui accueille régulièrement des étudiants de l'Et-na, s'écrit : « *ils ont la tête bien faite ! Ils possèdent une vision très acérée de leur métier au sein de l'entreprise. Les jeunes sont investis et passionnés dans ce qu'ils font. Ils sont flexibles et réagissent très vite.* » Jean-Marc Lacroix recrute régulièrement chez Thalès Communications & Security des étudiants de l'EPITA : « *Je chasse de nombreux profils d'ingénieurs, en particulier à l'EPITA car ils correspondent à nos métiers et aux personnes que nous recherchons. J'ai beaucoup apprécié les dernières générations car elles sont volontaires.* »



En outre, le groupe et chaque école travaillent ses relations entreprises, un point clef de l'enseignement supérieur aujourd'hui. Diverses manifestations sont organisées en partenariat avec des entreprises. En juin 2012, Microsoft et le groupe Ionis ont formalisé leur partenariat. Il instaure une relation privilégiée avec le Microsoft Campus Program. En 2013, un fonds de dotation sera créé en partenariat avec des entreprises qui vont être sollicitées au fur et à mesure des projets à financer.

Aller toujours plus loin

Etre multiple, décentralisé et présent dans plusieurs villes est le credo du groupe. Il a fait le choix de l'implantation en centre-ville, pour ne pas exclure les étudiants d'une grande partie de la vie. Comme un étudiant de province peut difficilement, au vu du coût exorbitant du logement à Paris, y passer cinq ans, mais que l'apprentissage de la mobilité est aujourd'hui nécessaire, Ionis souhaite que pendant ses cinq années d'études il acquiert une triple expérience : régionale, nationale et internationale. Marc Sellam, Pdg de Ionis Education Group, déclare : « Nous avons plus de 250 partenariats à l'international. Ce que je souhaite surtout, c'est que nous ouvrons plusieurs centres, notamment aux Etats-Unis et en Asie ». Epitech ouvre à la rentrée 2013 un site à Pékin, en partenariat avec la Beijing Jiatong University, et trois sites à la rentrée 2014 en Californie, en Grande-Bretagne et en Espagne. L'EPITA et SUP'Internet ont également des projets d'implantation.

L'engagement social

Le groupe Ionis réfléchit au financement des études. A la rentrée 2013, l'ETNA deviendra une école en cinq ans, avec désormais une prépa intégrée, PrepEtna, en deux ans, qui recrutera 250 jeunes par an en fonction des profils, des motivations et de critères sociaux. Elles sont entièrement gratuites. En contrepartie, l'école demande à contribuer de façon facultative à hauteur de 250 euros par mois pendant les trois dernières années d'étude, quand il commencera à travailler au sein d'une entreprise en alternance. « PrepEtna est un laboratoire innovant : nous verrons si le système est pérenne, finançable et reproductible, » confie Fabrice Bardèche. Et en 2010, Epitech a lancé Web@cadémie, une

formation gratuite de développeur-intégrateur Web en deux ans ouverte aux non bacheliers, en partenariat avec l'association de soutien scolaire aux enfants défavorisés Zup de Co.

Des passionnés, des concours et des robots

Les écoles d'informatique du groupe Ionis comptent également de nombreuses associations étudiantes bien évidemment dynamiques. Ainsi, Prologin (www.prologin.org) organise chaque année le Concours national d'informatique français, ouvert aux étudiants européens de moins de vingt ans. Elle est née en 1991 de l'initiative d'étudiants de l'EPITA et de l'École polytechnique. Félicitations à Le Thanh Dung Nguyen, le gagnant du concours 2013. « Des associations telles que Prologin ou France-IOI ont un rôle majeur à jouer dans la reconnaissance académique de l'algorithmique et de l'informatique en général, met en avant Jill-Jênn Vie, président de l'association interrogé dans ionismag du printemps 2013. L'Education nationale commence à peine à prendre en compte ces problématiques au travers de l'option ISN (Informatique et Sciences du Numérique en terminale scienti-



Forum des entreprise ETNA

fique depuis la rentrée 2012], et nous voulons participer à son développement ». Les membres d'Evolutek (<http://evolutek.org>), qui rassemble des étudiants de l'EPITA, d'Epitech et de l'école aéronautique IPSA, conçoivent des robots afin de participer à des concours comme la coupe de France de robotique, qui passionne les Japonais. La compétition est retransmise sur une chaîne de télévision japonaise. Le millésime 2013 s'est déroulé du 8 au 11 mai dernier à La Ferté-Bernard et a été remporté par l'Université d'Angers. Deux équipes françaises ont participé à la finale européenne Eurobot le 12 mai, et c'est – cocorico ! – encore l'université d'Angers qui l'a emporté face à RCVA (Robot Concept Ville d'Avray) et l'équipe allemande RCA. R2-D2 et 6PO n'ont qu'à bien se tenir, la relève est là !

Christine Calais

Ces entrepreneurs qui ont fait l'Epita ou l'Epitech

En solo ou avec des camarades de promotion, ils se sont lancés avec succès dans le grand bain entrepreneurial :

- Symble, SSII spécialisée sur les prestations autour de la suite Microsoft Office : Brice Pinsart et Quentin Veilhan, EPITA promotion 2012, anciens membres de l'association-entreprise Cristal, affirment : « L'EPITA nous a apporté les compétences techniques et algorithmiques dont nous nous servons au quotidien : la réflexion et la méthodologie de la programmation nous permettent de structurer notre approche du travail à effectuer de manière pertinente. »
- Melty, actualités en ligne et communauté jeune : Alexandre Malsch, Jérémy Nicolas, Jonathan Surpin, Epitech, promotion 2009.
- Prestashop, solution e-commerce open source : Bruno Lévêque, Epitech, promotion 2007.
- Prium Partners, cabinet de conseil en organisation, management et SI pour les secteurs banque, assurance et finance : Hélène Diep, Epita, promotion 2004.
- Kobojo, éditeur de jeux sur mobile et réseaux sociaux : Philippe Desgranges, Franck Tezaff et Vincent Vergonjeanne, Epita, promotion 2004.

L'exécutive MBA Epitech

Pour que les actifs ne fassent pas le choix entre formation continue et leur travail, Epitech a lancé son executive MBA en 2012, auquel participent 24 professionnels issus en partie d'Epitech et de l'EPITA. Il repose sur une plateforme en ligne. Chaque participant peut moduler les 400 heures selon son emploi du temps. Chaque mois, un nouveau module sur un grand thème comme le marketing, l'innovation ou la finance, est mis en ligne. Laurent Soubrevilla, EPITA promotion 2000 et fondateur de Visimmo3D, est venu « acquérir des notions plus pointues. » Romain Henri, Epitech promotion 2007, qui a travaillé cinq ans chez Sogeti, est satisfait du MBA : « il m'a permis de devenir moi-même un entrepreneur et de participer à la création d'une start-up ».

Mais les actifs plus ou moins jeunes ne sont pas priés uniquement de regarder leur écran. Fin février, le module leadership les a conduits sur une base d'entraînement du GIGN où ils ont dû se surpasser physiquement et mentalement. Avec ce parcours du combattant, ils ont appris à relativiser les problèmes et le stress en entreprise. Tout ce qui ne te tue pas te rend plus fort !

C.C.

WEB@CADEMIE

La persistance connectée dans les applications mobiles

La persistance connectée est un domaine quasi-obligatoire pour la plupart des applications mobiles. On nomme ici «persistance connectée» toutes les informations récupérées depuis un service web et sauvegardées sur le périphérique mobile de manière définitive ou non.

Celle-ci s'appuie sur 3 tâches différentes :

- Gestion des connexions réseaux
- Consommation d'un service web
- Sauvegarde des données

Nous vous proposons ici d'analyser ces trois parties distinctes sur les environnements de développement iOS et Windows 8.

LA PERSISTANCE CONNECTÉE SOUS iOS

> Gestion des connexions réseaux

A partir du moment où une application doit se connecter à un service web pour récupérer de l'information, la gestion réseau est à prendre en compte très vite. Sur iOS, le choix se fait entre deux possibilités :

- Le framework CFNetwork
- Les classe NSURL du framework UIKit

CFNetwork est un framework proposé par Apple dans le cadre du SDK d'iOS offrant une abstraction sur l'utilisation des protocoles réseaux. Celle-ci facilite l'utilisation de certaines tâches :

- Utilisation des sockets BSD
- Connexion cryptée avec TLS/SSL
- Résolutions de noms
- Connexion HTTP, HTTPS, FTP, etc.
- Découverte du réseau local (e.g. Bonjour)

Les classes NSURL* ont un périmètre d'utilisation bien plus limité. Elles ne permettent que la manipulation de tâches réseau manipulant des URL de types HTTP, HTTPS, FTP et FILE.

Cependant, l'utilisation des classes NSURL* est plus simple. De plus, dans le cadre d'une consommation d'un service web, l'utilisation de CFNetwork ne sera pas forcément nécessaire.

> Déterminer si une connexion réseau est disponible

Ce à quoi on ne pense pas toujours dans le cadre d'une connexion réseau, est la **disponibilité**. Dans le domaine du mobile, les connexions réseaux peuvent être de différents types : EDGE, 3G, Wifi, etc. et peuvent entraîner des comportements différents. Pour faciliter cette tâche, Apple fournit un exemple : Reachability

<http://developer.apple.com/library/ios/#samplecode/Reachability/Introduction/Intro.html>

Cet exemple propose une classe nommée Reachability possédant certaines fonctions pratiques pour déterminer le type de la connexion actuelle du mobile :

```
+ (Reachability*) reachabilityWithHostName: (NSString*) hostName;
+ (Reachability*) reachabilityWithAddress: (const struct sockaddr_in*) hostAddress;
+ (Reachability*) reachabilityForInternetConnection;
+ (Reachability*) reachabilityForLocalWiFi;
- (NetworkStatus) currentReachabilityStatus;
```

Elle propose aussi la gestion d'une notification permettant de signaler à l'application lorsqu'un changement significatif se produit sur la connexion réseau. Cependant, cette classe n'est pas compatible avec ARC (Automatic Reference Counting). Toutefois, certains développeurs ont réécrit la classe Reachability pour être compatible avec ARC (par exemple <https://github.com/tonymillion/Reachability>).

> Utilisation de frameworks tiers

Pour la consommation de service web, des frameworks tiers ont été développés pour faciliter le développement :

- ASIHTTPRequest - <http://allseeing-i.com/ASIHTTPRequest/>
- AFNetworking - <https://github.com/AFNetworking/AFNetworking>
- RestKit - <http://restkit.org/>

Le passage par ces frameworks permet de faciliter le développement en encapsulant les frameworks d'Apple avec des méthodes simples d'utilisation. Certains permettent aussi de faire la liaison avec un ORM ou l'encapsulation/désencapsulation des données au format JSON/XML.

> Consommation d'un service web

Une des tâches les plus importantes dans la persistance connectée est la consommation de l'information provenant d'un service web. Nous avons défini un service web dédié à la lecture/écriture de messages en ligne [Fig.1]. Celui-ci accepte et renvoie des informations au format JSON, contenant :

- Le message et le destinataire pour l'envoi d'un message
- Le message et la source pour la récupération du message

> Ecriture du client REST avec les classes NSURL du framework UIKit d'iOS

Nous allons utiliser quatre classes pour la réalisation du client :

- NSURLConnection
- NSURLRequest
- NSHTTPURLResponse
- NSURL

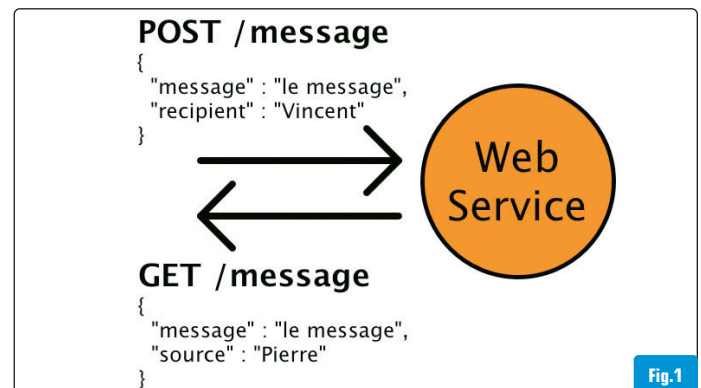


Fig.1

La classe `NSURLConnection` permet d'effectuer des requêtes synchrones ou asynchrones. Cependant, l'utilisation du mode asynchrone oblige l'utilisation d'un delegate. Afin de faciliter l'écriture du client, nous ferons ici du développement synchrone. Noter que si ce code est exécuté dans le thread principal, celui-ci sera bloqué le temps que le serveur réponde. Cela aura aussi pour effet de bloquer l'interface graphique. Voici un exemple de code permettant d'envoyer un message :

```
NSString* body = [NSString stringWithFormat:@"%s { \»message\» : \»%@\" , \»recipient\» : \»%@\" }", aMessage, recipient];

NSURL* url = [NSURL URLWithString:@"http://localhost/message"];
NSMutableURLRequest* request = [NSMutableURLRequest requestWithURL:url];
request.HTTPMethod = @"POST";
request.HTTPBody = [body dataUsingEncoding:NSUTF8Encoding];

NSError* error = nil;
NSHTTPURLResponse response = nil;
[NSURLConnection sendSynchronousRequest:request returningResponse:&response error:&error];

if(error) {
    NSLog(@"ERROR - Can't send message");
} else {
    if(response.statusCode == 200) {
        NSLog(@"Message sended !");
    } else {
        NSLog(@"SERVER ERROR");
    }
}
```

On remarque la nécessité de vérifier le remplissage ou non de la variable `error` afin de déterminer si une erreur s'est produite. L'écriture du code permettant de lire le message pourrait s'écrire de la sorte :

```
NSURL* url = [NSURL URLWithString:@"http://localhost/message"];
NSMutableURLRequest* request = [NSMutableURLRequest requestWithURL:url];
request.HTTPMethod = @"GET";

NSError* error = nil;
NSHTTPURLResponse response = nil;
NSData* receivedData = [NSURLConnection sendSynchronousRequest:request returningResponse:&response error:&error];
NSString* receivedMessage = nil;
if(error) {
    NSLog(@"ERROR - Can't receive message");
} else {
    if(response.statusCode == 200) {

        NSDictionary* dict = [NSJSONSerialization JSONObjectWithData:receivedData options:0 error:&error];
        if(!error) {
            receivedMessage = [dict objectForKey:@"message"];
        } else {
            NSLog(@"JSON ERROR");
        }
    }
}
```

```
} else {
    NSLog(@"SERVER ERROR");
}
}
if(receivedMessage) {
    NSLog(@"Vous avez reçu un message : %@", receivedMessage);
}
```

On remarque que l'écriture d'une simple méthode d'accès au service web avec les classes `NSURL` est très verbeuse, et ce pour les deux méthodes de notre service web.

> Ecriture du client REST avec le framework AFNetworking

`AFNetworking` ré-utilise les classes `NSURL` pour faciliter le développement. Cependant, il utilise aussi les classes `NSOperation` et `NSOperationQueue` pour gérer une file d'attente de requêtes et faciliter l'exécution de requêtes en tâche de fond.

Le code permettant d'envoyer un message :

```
NSString* body = [NSString stringWithFormat:@"%s { \»message\» : \»%@\" , \»recipient\» : \»%@\" }", aMessage, recipient];

NSURL* url = [NSURL URLWithString:@"http://localhost/message"];
NSMutableURLRequest* request = [NSMutableURLRequest requestWithURL:url];
request.HTTPMethod = @"POST";
request.HTTPBody = [body dataUsingEncoding:NSUTF8Encoding];

[[AFJSONRequestOperation JSONRequestOperationWithRequest:request success:^(NSURLRequest *request , NSHTTPURLResponse *response , id JSON) {

    NSLog(@"Message Sended");

} failure:^(NSURLRequest *request , NSHTTPURLResponse *response , NSError *error , id JSON) {

    NSLog(@"ERROR");

}]] start];
```

Le code permettant la récupération du message :

```
NSURL* url = [NSURL URLWithString:@"http://localhost/message"];
NSMutableURLRequest* request = [NSMutableURLRequest requestWithURL:url];
request.HTTPMethod = @"GET";

[[AFJSONRequestOperation JSONRequestOperationWithRequest:request success:^(NSURLRequest *request , NSHTTPURLResponse *response , id JSON) {

    NSLog(@"Vous avez reçu un message : %@", [JSON objectForKey:@"message"]);

} failure:^(NSURLRequest *request , NSHTTPURLResponse *response , NSError *error , id JSON) {

}
```



```
NSLog («ERROR»);

}} start];
```

L'utilisation d'AFNetworking facilite l'utilisation de JSON. Pas besoin d'analyser les informations reçues depuis le service web, le framework se charge de le convertir automatiquement.

> La sauvegarde des données

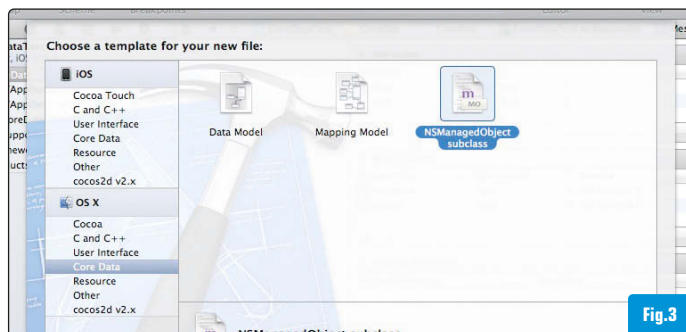
Dans certains cas, les informations récupérées depuis un service web ont besoin d'avoir une certaine pérennité au sein de l'application. Il est donc nécessaire d'avoir les bons outils pour sauvegarder ces informations dans une mémoire non-destructrice du système. (Mémoire flash)

Sur iOS, nous avons plusieurs possibilités :

- Sauvegarde des informations sous forme de fichier
- Sauvegarde des informations dans une base de données
- Sauvegarde des informations en utilisant l'ORM d'Apple

L'avantage de la dernière solution est qu'elle permet d'avoir un seul et unique mode de gestion des informations, tout en gardant la possibilité de faire évoluer le système de sauvegarde, soit par base de données, soit par fichier. Cette couche ORM fournie par Apple s'appelle **Core Data**. Dans la majeure partie des cas, Core Data sauvegardera les informations dans une base de données SQLite, cependant, il est tout à fait possible de choisir d'autres modes de sauvegarde :

- Fichier texte .plist d'Apple
- Fichier texte (XML, JSON, CSV, etc.)
- Fichier binaire



Cependant, il sera nécessaire au développeur de développer des mapping pour indiquer à Core Data comment faire la liaison entre la sauvegarde physique et les objets Core Data.

Trois objets sont nécessaires à l'utilisation de Core Data :

- **Managed Object Model** : Pour faire simple, c'est la définition des objets sauvegardables (à la manière d'une table SQL et de ses champs). On appelle aussi ces objets des entités. C'est ici que l'on définit le modèle de notre persistance.
- **Persistent Store Coordinator** : C'est le connecteur entre l'abstraction fournie par Core Data et le mode de sauvegarde des données.
- **Managed Object Context** : C'est l'objet principal qu'on utilise dans Core Data. Il a pour rôle de nous permettre de récupérer/créer/supprimer/modifier des objets.

Apple fournit un ensemble d'outils au sein de XCode pour permettre l'utilisation de Core Data.

> Le modèle de données

Ces fichiers au format *.xcdatamodeld, connu par XCode, permettent le design d'une modélisation [Fig.2].

> Les objets Core Data

Afin de faciliter l'utilisation de Core Data, Apple a aussi créé une classe spécifique héritant de NSObject : NSManagedObject.

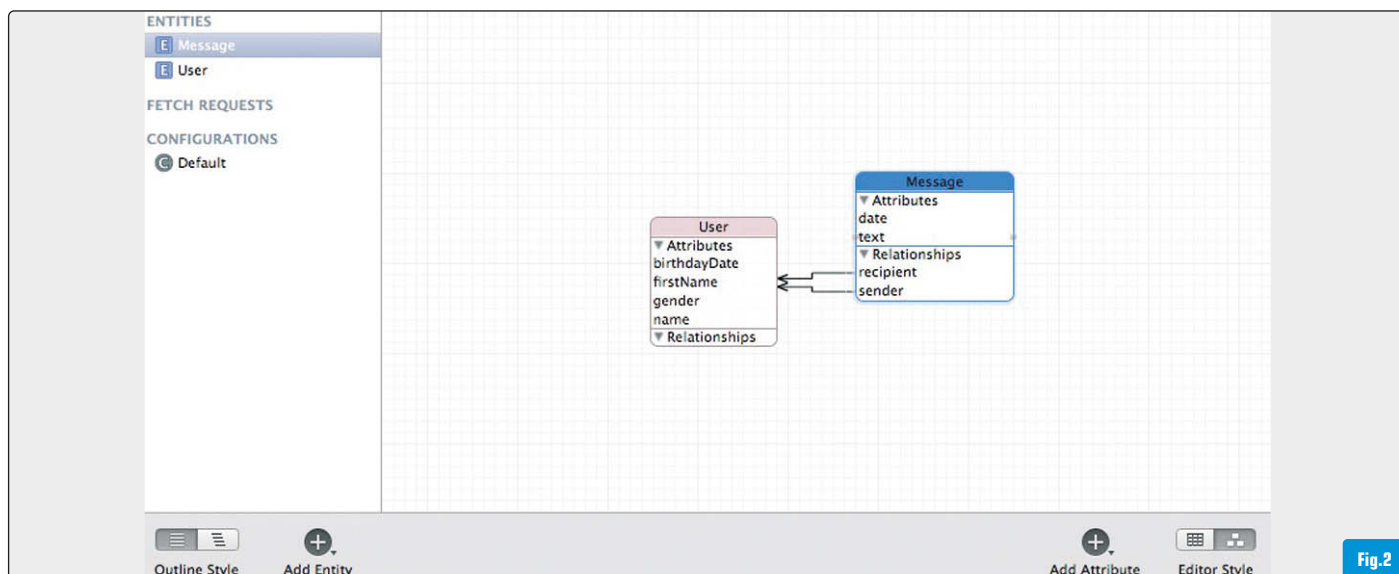
Attention, cette classe ne doit jamais être instanciée directement, mais doit être récupérée via l'utilisation du Managed Object Context. Afin de faciliter la création de sous-classes, Apple a intégré dans XCode un modèle de fichier permettant de générer les NSManagedObject correspondant à chaque entité définie dans le modèle de données. Pour cela, allez dans File > New > File ..., dans le menu de gauche, sélectionnez le sous-menu iOS puis NSManagedObject subclass [Fig.3]. Sélectionnez ensuite le répertoire de création et cliquez sur Create. Une classe pour chaque entité, héritant de NSManagedObject sera créée dans votre projet.

LA PERSISTENCE CONNECTÉE SOUS WINDOWS 8

> Gestion des connexions réseaux

Le runtime Windows fournit un ensemble d'API réseau dédiées à la communication sur des réseaux privés ou sur internet :

- L'API Windows.Web.AtomPubet Windows.Web.Syndication : Ces



API permettent de récupérer des flux RSS ou Atom dans ses multiples versions très aisément

- L'API Windows.Networking.BackgroundTransfer : Cette API est utilisable dans le cas où l'on nécessite le téléchargement de contenu sur le web.
- L'API System.Net.Http.HttpClient : Celle-ci est principalement dédiée à la réalisation de clients de service web.
- L'API Windows.Networking.Proximity : API permettant la liaison entre deux appareils proches

> Déterminer l'état de la connexion

Pour déterminer l'état de la connexion d'un appareil nous disposons de l'espace de nom Windows.Networking.Connectivity. Cet espace contient la classe NetworkInformation que l'on peut utiliser de la façon suivante :

```

ConnectionProfile profile = NetworkInformation.GetInternetConnectionProfile();
if (profile.GetNetworkConnectivityLevel() == NetworkConnectivityLevel.InternetAccess)
{
    Debug.WriteLine("connected to internet");
}
{
    Debug.WriteLine("Not connected to internet");
}

```

Il est aussi possible de s'abonner à un évènement afin d'être informé à chaque changement d'état de la connexion grâce au code suivant :

```

NetworkInformation.NetworkStatusChanged += NetworkInformation_NetworkStatusChanged;

void NetworkInformation_NetworkStatusChanged(object sender)
{
    CheckConnection();
}

```

> Consommation d'un web service

Sous Windows 8 plusieurs types d'architectures de service web peuvent être consommés par une application :

- Services SOAP
- Service REST

La consommation de service SOAP a pour avantage de nécessiter très peu de développement car grâce à Visual Studio tout est généré automatiquement et prêt à la consommation.

Contrairement aux services SOAP, les services REST ne sont pas gérés par Visual Studio, Il est donc nécessaire d'implémenter les appels services ou d'utiliser un outil pour cela.

> L'utilisation d'un service SOAP

La gestion des services SOAP se fait très facilement dans Visual Studio. Pour cela il vous suffit d'ajouter une référence web et laisser

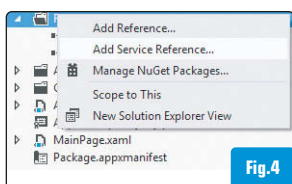


Fig.4

Visual Studio générer les méthodes d'appel vers le service web, ainsi que toutes les entités nécessaires, automatiquement [Fig.4 et 5]. L'utilisation d'une méthode du service web s'effectue ensuite de la manière suivante :

```

Service1Client client = new Service1Client();
Double result = await client.SumAsync(5, 6);

```

Service1Client est l'objet qui va nous permettre d'accéder à toutes nos méthodes de services. Il est important de noter que dans Windows 8 tous les appels services doivent être asynchrones, c'est pourquoi notre objet Service1Client contient une méthode asynchrone nommée SumAsync qui, ici, effectue une simple addition de manière asynchrone.

> L'utilisation d'un service REST

Pour la consommation de services REST en C#, plusieurs possibilités s'offrent à nous :

- Utiliser la classe System.Net.Http.HttpClient et coder les appels vers le service web et la sérialisation/désérialisation des données à partir de zéro.
- Utiliser un outil tiers spécialisé tel que RestSharp (<http://restsharp.org>) ou autre

Dans cet article nous allons étudier l'utilisation de System.Net.Http.HttpClient car l'implémentation reste très simple et est à la base de tous les frameworks tiers existants.

Voici comment envoyer un message :

```

HttpClient client = new HttpClient();
HttpContent content = new System.Net.Http.StringContent("hello world !!");
HttpResponseMessage response = await client.PostAsync("http://localhost/message", content);

if (response.IsSuccessStatusCode)
{
    String msg = await response.Content.ReadAsStringAsync();
    Debug.WriteLine("Message sent!");
}
else
{
    Debug.WriteLine(String.Format("Sending message failed with status code {0}", response.StatusCode));
}

```

Un peu comme pour les services SOAP, on possède une classe nommée HttpClient qui va nous servir d'outil afin d'effectuer des appels

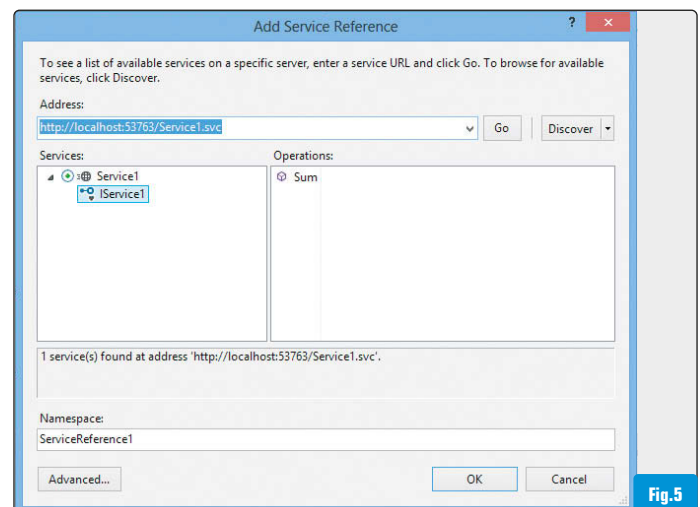


Fig.5

via le protocole HTTP, puis on envoie le message «hello world» à l'adresse suivante <http://localhost/message>.

Pour récupérer un message, le code serait le suivant :

```
HttpClient client = new HttpClient();
HttpResponseMessage response = await client.GetAsync("http://localhost/message");

if (response.IsSuccessStatusCode)
{
    String msg = await response.Content.ReadAsStringAsync();
    Debug.WriteLine(String.Format(msg));
}
else
{
    Debug.WriteLine(String.Format("Request failed with status code {0}", response.StatusCode));
}
```

> Stockage des données

Afin de stocker des données en local sous Windows plusieurs techniques sont disponibles afin de convenir à une grande majorité des scénarios :

- Stocker les données dans les Settings de l'application
- Sérialiser les données et les stocker dans un fichier
- Utiliser une base de données embarquée

> Stockage via «Application Data»

Les principales fonctions de stockage se situent dans l'espace de nom Windows.Storage qui contient lui-même une classe nommée ApplicationData qui va nous servir d'interface pour la sauvegarde des données dans notre application.

Voici un exemple d'utilisation de la classe :

```
var localSettings = ApplicationData.Current.LocalSettings;

// Create a setting
localSettings.Values["message"] = "Hello Windows";

// Read data from a setting
Object value = localSettings.Values["message"];

if (!value) {
    // No data
}
else {
    // Access data in value
}

// Delete a setting
localSettings.Values.Remove("message");
```

On remarquera l'utilisation de l'objet ApplicationData.Current.LocalSettings, celui-ci garantit que toutes les informations sont stockées dans le domaine de l'application et restent inaccessible à toute autre application.

Bien que ApplicationData.Current.LocalSettings soit très pratique, son utilisation est limitée à des objets très simples et n'est pas adaptée au stockage de données importantes.

> Sérialisation et stockage de données

La sérialisation va nous permettre de stocker tout type de données en faisant abstraction de la taille de celles-ci et d'avoir une lecture / écriture très rapide. Tout d'abord définissons un type d'objet à stocker :

```
public class Message
{
    public String Sender {get; set;}
    public String Content {get; set;}
}
```

Puis la définition d'une méthode générique afin de stocker tout type d'objet :

```
public async Task SaveAsync<T>(T Obj, string FileName)
{
    try
    {
        if (Obj != null) //dont save a null objet
        {
            //Create a file in the application folder
            StorageFile file = null;
            StorageFolder folder = ApplicationData.Current.LocalFolder;
            file = await folder.CreateFileAsync(FileName, CreationCollisionOption.ReplaceExisting);

            //Serialize the data into the file using JSON format
            var writeStream = await file.OpenAsync(FileAccessMode.ReadWrite);
            DataContractJsonSerializer serializer = new DataContractJsonSerializer(typeof(T));
            Stream outputStream = writeStream.AsStreamForWrite();
            serializer.WriteObject(outputStream, Obj);
        }
    }
    catch (Exception e)
    {
        //something wrong appended
        System.Diagnostics.Debug.WriteLine(e.ToString());
    }
}
```

Ensuite on définit une méthode de lecture d'objet générique :

```
public async Task<T> LoadAsync<T>(string FileName)
{
    try
    {
        //Open the file
        StorageFile file = null;
        StorageFolder folder = ApplicationData.Current.LocalFolder;
        file = await folder.GetFilesAsync(FileName);
        var readStream = await file.OpenAsync(FileAccessMode.Read);

        //Read the object
        Stream inputStream = readStream.AsStreamForRead();
        DataContractJsonSerializer serializer = new DataContractJsonSerializer(typeof(T));
        T obj = (T)serializer.ReadObject(inputStream);
    }
}
```




```

    return obj;
}
catch (FileNotFoundException)
{
    //No persistence file found
    return default(T);
}
catch (Exception)
{
    //something wrong appended
    return default(T);
}
}

```

Il est important de noter qu'ici les fichiers sont sauvegardés en clair dans le système de fichier de l'utilisateur et que son contenu est facilement modifiable. Il est donc important d'ajouter une notion de sécurité et de chiffrement si l'on désire sauvegarder des informations sensibles. Il ne nous reste plus qu'à sauvegarder et lire notre objet de la manière suivante :

```

messages.add(new Message() { Sender = «admin», Content = «It works !!»});

//Save the list of messages
SaveAsync(messages, «messages.json»)

//read the list of messages
var readMessages = LoadAsync<List<Messages>>(«messages.json»);

```

> Sauvegarde de données via SQLite

Pour installer SQLite, cliquez sur Outils > Extension et mise à jour ensuite cherchez et installez «SQLite for windows runtime» [Fig.6 et 7]. Une fois Visual Studio redémarré, vous pouvez ajouter la référence SQLite à votre projet ainsi qu'installer le package NuGET sqlite-net. Nous pouvons maintenant définir le modèle de données à écrire en base :

```

public class Message
{
    [SQLite.PrimaryKey]
    [SQLite.AutoIncrement]

```

```

    public int Id { get; set; }
    public String Sender { get; set; }
    public String Content { get; set; }
}

```

C'est le même que dans l'exercice précédent, sauf qu'il contient maintenant une variable Id afin de pouvoir identifier chaque message de manière unique.

La création de la base de données s'effectue très facilement grâce à SQLite. Si la base de données ou une table particulière n'existe pas, elles sont automatiquement générées grâce au code suivant :

```

var dbPath = Path.Combine(Windows.Storage.ApplicationData.
Current.LocalFolder.Path, «db.sqlite»);
using (var db = new SQLite.SQLiteConnection(dbPath))
{
    // Create the table if they don't exist
    db.CreateTable<Message>();
}

```

L'écriture de l'objet en base de données s'effectue ensuite de la façon suivante :

```

using (var db = new SQLite.SQLiteConnection(dbPath))
{
    db.Insert(new Message() { Sender = «Admin», Content = «It works!» });
}

```

Et la lecture :

```

using (var db = new SQLite.SQLiteConnection(dbPath))
{
    // Create the table if they don't exist
    var messages = db.Table<Message>().ToList();
}

```

Tout comme la sérialisation, les informations restent facilement accessibles et le stockage d'information sensible doit être soumis à une politique de sécurité adéquate.

Vincent Saluzzo

Ingénieur étude et développement - «Expert en développement iOS», Itelios

Fitzgerald Muiseroux

Ingénieur étude et développement -

«Spécialisé dans le développement C#/XAML», Itelios

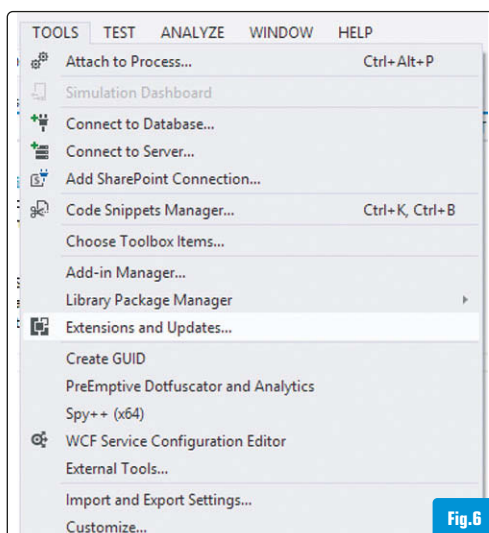


Fig. 6

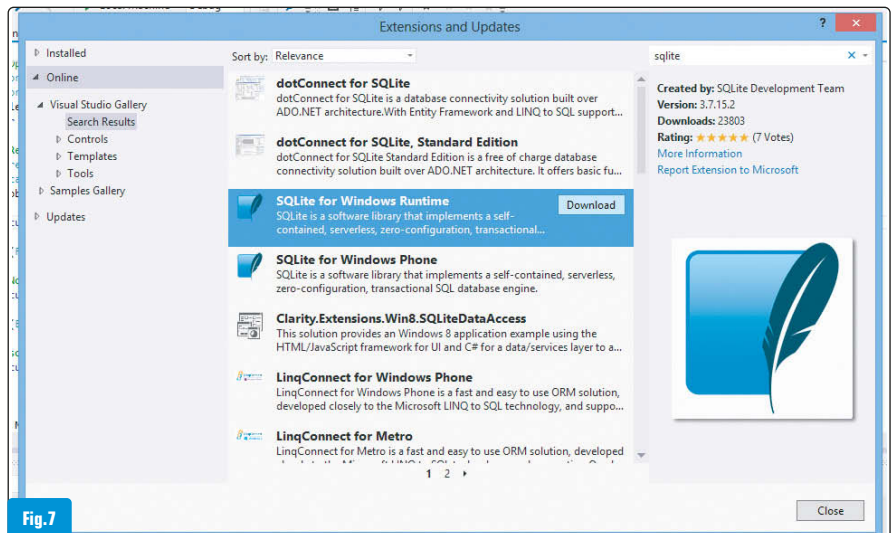


Fig. 7

Une application Windows Store réactive en 5 étapes

Si Microsoft prône la réactivité de son dernier système d'exploitation, il n'en reste pas moins à la charge des développeurs de créer des applications responsive. Voici quelques étapes à suivre pour que vos applications répondent au doigt et à l'œil.

Quand on utilise Windows 8, on s'attend à avoir des applications réactives, qui ne subissent aucun temps de latence, sur un ordinateur comme sur une tablette, avec une souris comme en touch. Lorsqu'un utilisateur effectue une action, l'application doit répondre immédiatement en fournissant au minimum à ce dernier un feedback visuel ; un indicateur montrant un chargement en cours par exemple. Un responsive design passe avant tout par la perception de l'utilisateur de l'application qu'il utilise. Une application réactive se doit également d'être une application performante. Inutile d'afficher des indicateurs de chargement si toutes les pages de votre application prennent 30s à être chargées. En 30s, l'utilisateur aura le temps d'aller sur le Store pour télécharger l'application concurrente. Là encore, il existe diverses astuces, facilement applicables, pour garder une application responsive. Les 5 étapes ci-dessous appartiennent aux deux catégories citées ci-dessus, la perception et les performances. Les exemples sont en C# et XAML, mais les notions présentées s'appliquent à toutes les technologies.

> Utilisez des transitions et animations

Vous l'avez sans doute déjà remarqué, les applications Windows Store utilisent de nombreuses animations. Les données n'apparaissent pas brusquement à l'écran mais arrivent généralement en douceur, accompagnées d'un effet visuel. Nativement, plusieurs contrôles embarquent des animations de ce style. C'est le cas de la GridView qui, par exemple, anime par défaut l'ajout ou la suppression d'un élément. Fonctionnellement, ces animations n'apportent rien à l'application. En revanche, elles offrent à l'utilisateur une impression de réactivité accrue. Elles permettent de rendre vivantes et dynamiques des applications qui pourraient être plates et inintéressantes. Il est possible d'utiliser différentes animations prédéfinies sur chaque contrôle. Ces animations s'appellent des transitions. Les transitions s'exécutent automatiquement lorsque le contrôle sur lequel elles sont définies se trouve dans un état précis (lorsque celui-ci est modifié par exemple). Il existe plusieurs transitions prédéfinies : AddDeleteThemeTransition, ContentThemeTransition, EntranceThemeTransition, PopupThemeTransition, ReorderThemeTransition, RepositionThemeTransition. Ces transitions peuvent s'appliquer à trois niveaux différents.

1/ Sur un contrôle via la propriété **Transitions**.

```
<TextBlock Text="Hello World !" >
  <TextBlock.Transitions>
    <TransitionCollection>
      <ContentThemeTransition />
    </TransitionCollection>
  </TextBlock.Transitions>
</TextBlock>
```

Ici, le TextBlock sera animé lorsque son texte changera.

2/ Sur les enfants d'un contrôle via la propriété **ChildrenTransitions**.

```
<StackPanel>
  <StackPanel.ChildrenTransitions>
    <TransitionCollection>
      <EdgeUIThemeTransition Edge="Top" />
    </TransitionCollection>
  </StackPanel.ChildrenTransitions>

  <!-- Childrens -->
</StackPanel>
```

Dans ce second cas, ce sont les éléments appartenant à ce StackPanel qui seront animés lors de leur « entrée » sur la page.

3/ Sur les éléments d'un ItemsControl via la propriété **ItemContainerTransitions**.

```
<GridView>
  <GridView.ItemContainerTransitions>
    <TransitionCollection>
      <EntranceThemeTransition IsStaggeringEnabled="True" />
    </TransitionCollection>
  </GridView.ItemContainerTransitions>
</GridView>
```

Il est également possible d'utiliser ces transitions dans des storyboards. Dans ce cas, vous contrôlez l'instant où sera déclenchée l'animation.

```
<Storyboard x:Name="Storyboard1" Storyboard.TargetName="TextBlox1">
  <FadeOutThemeAnimation></FadeOutThemeAnimation>
</Storyboard>
```

> Chargez vos données de façon incrémentale

Il n'est pas impossible que vous deviez afficher un nombre très important d'éléments sur une page de votre application. Par exemple, si vous faites une application pour un site e-commerce et que vous devez afficher tous les articles d'une certaine catégorie, vous pouvez vous retrouver avec des milliers d'éléments à afficher. Afficher tous ces articles en une seule fois n'est pas la bonne solution. Le chargement sera long, l'application va probablement ramer sur des devices peu puissants, et dans la plupart des cas, l'utilisateur n'a certainement pas besoin de voir tous les éléments. Quand bien même il voudrait tout voir, il ne pourra pas voir plus d'éléments que ce qui est affiché en une fois sur son écran. Il faut donc opter pour un chargement incrémental. Cette technique consiste à char-

ger un nombre restreint d'éléments dans un premier temps, puis de charger les autres lorsque l'utilisateur le demande. Si l'on utilise une GridView pour afficher les éléments, cette demande sera matérialisée par le scroll dans la GridView. L'interface `ISupportIncrementalLoading` définit un contrat permettant de réaliser cela.

```
public interface ISupportIncrementalLoading
{
    bool HasMoreItems { get; }

    IAsyncOperation<LoadMoreItemsResult> LoadMoreItemsAsync(uint count);
}
```

La méthode `LoadMoreItemsAsync` permet d'initialiser le chargement incrémental, tandis que la propriété `HasMoreItems` renvoie un booléen indiquant s'il y a d'autres éléments à charger. Eric Vernié, évangéliste chez Microsoft France, propose sur son blog une implémentation de cette interface au sein d'une classe nommée **IncrementalLoadingCollection**. Cette classe générique possède un constructeur prenant en paramètre une fonction renvoyant `List<T>` (T étant l'élément affiché dans votre `ItemsControl`) et prenant en paramètre un `CancellationToken` (pour annuler la tâche asynchrone si besoin) et un entier représentant le nombre maximum d'items à charger.

```
IncrementalLoadingCollection<Movie> _data;
_data = new IncrementalLoadingCollection<Movie>((CancellationToken cts, uint count) =>
{
    return Task.Run<List<Movie>>(async () =>
    {
        List<Movie> movies = new List<Movie>();

        if (_page == 1)
        {
            movies = await repo.GetPopularMoviesAsync();
        }
        else
        {
            movies = await repo.GetPopularMoviesAsync(_page);
        }

        _page += 1;
        return movies;
    });
});

MoviesGridView.DataContext = _data;
```

Dans l'exemple précédent, une liste de films est renvoyée. La tâche (`Task.Run`) ligne 3 est appelée dès que l'utilisateur scrolle et qu'il est nécessaire d'afficher de nouveaux éléments. La variable `_page` permet de savoir quelles données charger. Il existe plusieurs paramètres permettant de paramétrer le chargement des données ; vous trouverez toutes ces informations sur le blog d'Eric Vernié (blogs.msdn.com/b/devosaure).

> Mettez en cache vos ressources

Rares sont les applications qui n'affichent aucun média, aucune image ou vidéo, ou qui ne font appel à aucun service tiers nécessi-

tant un accès à Internet, et donc un téléchargement de données. Aujourd'hui, plus que jamais, il est nécessaire de gérer un système de cache de façon à accéder aux données le plus rapidement possible dès que celles-ci ont été téléchargées une première fois. Leur mise en cache accélérera drastiquement leur chargement et donc la réactivité de l'application. C'est le cas pour n'importe quelle donnée métier, qui peut généralement être sérialisée dans les `ApplicationData`. Mais c'est aussi et surtout le cas pour les images, qui sont bien souvent lourdes et mettent du temps à s'afficher.

La classe `BackgroundDownloader` permet de télécharger des ressources lourdes (audio, vidéos, images HD). Elle permet en quelques lignes de récupérer une ressource à une URI donnée et de la télécharger dans un fichier local.

```
Uri imageUri = new Uri(@"http://www.site.com/image.png");
StorageFile image = await ApplicationData.Current.LocalFolder.
CreateFileAsync(@"image.png");

BackgroundDownloader downloader = new BackgroundDownloader();
DownloadOperation download = downloader.CreateDownload(image
Uri, image);
await download.StartAsync();
```

L'exemple ci-dessus permet de télécharger une image dans le fichier `image.png`, situé dans le `LocalFolder`. Plutôt que de télécharger cette image à chaque fois pour l'afficher, il suffira par la suite d'aller la chercher directement en local.

Dans les applications Windows Store, les pages peuvent elles aussi être mises en cache. La façon dont est géré le cache est définie par la propriété `NavigationCacheMode` de chaque Page. Celle-ci prend l'une des trois valeurs suivantes :

- **Disabled** : désactive le cache. Une nouvelle instance de la page est créée à chaque affichage.
- **Enabled** : active le cache. Après sa première instanciation, la page est mise en cache et restaurée lorsque l'utilisateur navigue à nouveau vers cette page. La frame mettant en cache cette page peut toutefois la détruire si elle ne possède plus d'espace pour le cache.
- **Required** : active le cache et passe outre les limites d'espace du cache de la frame.

```
public Page1()
{
    this.InitializeComponent();

    this.NavigationCacheMode = NavigationCacheMode.Enabled;
}
```

Concrètement, chaque frame gère le cache pour toutes les pages qu'elle affiche. La classe `Frame` possède une propriété `CacheSize`, de type entier, permettant d'indiquer le nombre de frames qu'elle peut contenir en cache (10 par défaut). Une fois en cache, une page visitée est réaffichée immédiatement dans l'état dans lequel elle était lorsque l'utilisateur l'a quittée.

> Virtualisez l'UI

Si la virtualisation des données (étape 2) est importante, la virtualisation de l'UI l'est tout autant. Celle-ci va influencer directement sur les performances de votre application, et notamment des pages ayant de grosses listes de données à afficher. Explications.

Chaque contrôle possède ce que l'on appelle un `viewport`. Ce view-

port représente la zone d'affichage dudit contrôle. Par exemple, pour une GridView, le viewport est défini par sa hauteur et sa largeur. Pour un ScrollView en revanche, le viewport est extensible puisqu'il est défini par son contenu. La virtualisation de l'UI ne peut s'appliquer qu'aux contrôles ayant des viewport restreints. Cela consiste en la virtualisation des éléments en dehors du viewport, c'est-à-dire en dehors de la partie visible du contrôle.

Par défaut, les ItemsControls (ListView et GridView) supportent la virtualisation de l'UI. Cela est rendu possible grâce à l'utilisation respective d'un WrapGrid et VirtualizingStackPanel (VSP) dans leur ItemsPanel.

```
<GridView>
  <GridView.ItemsPanel>
    <ItemsPanelTemplate>
      <WrapGrid />
    </ItemsPanelTemplate>
  </GridView.ItemsPanel>
</GridView>
```

Dans ce cas, les éléments de la GridView seront virtualisés.

```
<GridView>
  <GridView.ItemsPanel>
    <ItemsPanelTemplate>
      <StackPanel />
    </ItemsPanelTemplate>
  </GridView.ItemsPanel>
</GridView>
```

Dans ce second cas, ils ne le seront pas. Il faut utiliser un VSP à la place du StackPanel pour rendre possible la virtualisation de l'UI dans ce cas. Pour les ItemsControls, seuls les éléments affichés à l'écran sont créés, les autres non. En fait, chaque élément est créé lorsqu'il est sur le point d'être affiché (soit, lorsque le scrollviewer interne à ces contrôles est sur le point d'atteindre ces éléments). Lorsqu'un élément n'est plus affiché, son conteneur en mémoire est réutilisé pour afficher un autre élément. De cette façon, l'utilisateur à l'impression que tous les éléments sont chargés, alors que seuls quelques-uns sont en mémoire.

Le mode de virtualisation des éléments peut être défini via l'attached property VirtualizingStackPanel.VirtualizationMode. Celle-ci possède deux valeurs :

Standard : Crée et supprime les items en mémoire lorsque ces derniers doivent être affichés / retirer

Recycling : Réutilise les conteneurs en mémoire dès que possible pour afficher les éléments

> Optimisez le chargement de l'application

En toute logique, la première chose qu'un utilisateur fait après avoir téléchargé une application, c'est de la lancer. Ce lancement occasionne bien souvent le chargement de nombreuses données et ressources. Il est nécessaire d'optimiser ce temps de chargement afin d'offrir à l'utilisateur, le plus rapidement possible, un feedback visuel. Lorsque le chargement de votre application est long (plusieurs secondes), il est conseillé d'afficher un « extended splash screen ». Il s'agit d'un simple écran reprenant l'aspect du splash screen (à savoir une couleur de fond unie et une image de 620*300 pixels, centrée). La différence avec le splash screen, c'est que vous pouvez utiliser l'extended splash screen pour afficher des informations à

l'utilisateur. Par exemple, un simple ProgressRing permet d'indiquer à ce dernier que des données sont en train d'être chargées. Même si cela n'accélèrera pas le chargement des données en question, l'utilisateur sera rassuré d'avoir un feedback visuel, plutôt que d'avoir un splash screen fixe durant plusieurs secondes. Vous trouverez un sample et des articles pour réaliser cet extended splash screen sur dev.windows.com.

Vient ensuite le chargement des ressources de votre application. Lorsqu'une application Windows Store est lancée, le fichier app.xaml est parsé. De même, tous les fichiers contenant des ressources référencées dans le fichier app.xaml sont parsés.

```
<Page.Resources>
  <Style TargetType=>TextBlock<>
    <Style.Setters>
      <Setter Property=>Style Value=>{StaticResource TextButton
Style}> />
    </Style.Setters>
  </Style>
</Page.Resources>
```

Dans l'exemple précédent, la page définit un style où l'un des setters fait référence à la ressource TextButtonStyle. Cette ressource se situe dans le dictionnaire StandardStyles.xaml. Cela signifie que le dictionnaire en question sera entièrement parsé lors de l'affichage de la page. Ce temps ne se ressent pas directement auprès de l'utilisateur, mais l'accumulation de ces quelques millisecondes peut, à terme, impacter négativement l'expérience utilisateur.

De même, il est recommandé de :

- Factoriser les ressources utilisées : ne pas définir deux fois la même Brush à deux endroits différents par exemple
- Placer les ressources au bon endroit : dans la page si elle n'est utilisée que dans cette page, dans app.xaml si elle est utilisée sur plusieurs pages ou dans un dictionnaire pour la rendre portable
- Éviter d'empiler des contrôles inutilement. Par exemple, plutôt que de mettre une Grid dans un Border en utilisant sa propriété Background, utilisez directement cette propriété sur la Grid

Pour cette dernière recommandation, vous pouvez initialiser DebugSettings.IsOverdrawHeatMapEnabled à true (dans le constructeur App par exemple), afin de visualiser facilement les contrôles qui se chevauchent inutilement.

> Conclusion

Il est aujourd'hui indispensable de penser optimisation et réactivité lorsque l'on développe une application. Cette réactivité est obtenue principalement de deux façons, en jouant sur la perception de l'utilisateur, c'est-à-dire en lui donnant un feedback visuel dès que possible, même si les données auxquelles il souhaite accéder ne sont pas encore prêtes, et en améliorant les performances brutes de l'application, en jouant sur la virtualisation de l'UI notamment. Construire une application réactive, c'est également porter une attention particulière à chaque bout de code, à chaque section d'une vue, de façon à éviter l'instruction qui plomberait toute l'expérience utilisateur. Pour conclure, gardez à l'esprit qu'il est toujours possible d'améliorer la réactivité d'une application.

Loïc Rebours

Consultant .NET, Avanade France - Microsoft MVP C#

@loicrebours - www.blog.loicrebours.fr

Développer un job Map/Reduce pour Hadoop



Hadoop Map/Reduce est un framework de calcul distribué basé sur le paradigme fonctionnel. Les données vont être manipulées par un ensemble de fonctions ne partageant (en théorie) aucun état mutable.

Ainsi, avant de commencer par les spécificités d'Hadoop, il est utile de comprendre ce que signifient les concepts de *map* et de *reduce* en général. Ce sont des abstractions que vous utilisez probablement tous les jours, même si vous n'avez pas nécessairement l'habitude de les identifier ni de les nommer.

> Map

Soit deux types A et B ; si on possède une fonction capable de transformer A en B, alors avec map on est capable de transformer une collection de A en une collection de B.

Plus concrètement, si on possède une classe *Person* avec une fonction *getFirstname()* retournant son prénom,

* en java8, on pourrait écrire :

```
persons.stream().map(person -> person.getFirstname())
```

* et dans le langage de tous les jours :

```
Collection<String> firstnames = new ArrayList<String>(persons.size());
for (Person person : persons) {
    firstnames.add(person.getFirstname());
}
```

L'opération map permet d'abstraire l'implémentation réelle d'un ensemble de transformations sur une collection qui ne partagent pas d'état mutable. Dans les cas simples, sur un faible volume de données, une boucle séquentielle est sûrement la meilleure approche. Dès que le volume va grossir, paralléliser en local les transformations peut améliorer le temps de réponse mais il faut alors gérer la synchronisation. Distribuer le traitement sur plusieurs machines permet de traiter des volumes encore plus gros mais il faut alors gérer les cas de pannes. C'est ce que fait Hadoop map/reduce. Étant donné que l'implémentation n'est pas à la charge du client, il n'a plus à se soucier du nombre de fois ni de l'endroit où les transformations ont eu lieu. Seul le résultat compte.

> Reduce

Reduce est une opération se réalisant aussi sur une collection mais permettant le partage d'un état local au détriment de la parallélisation des opérations. Faire une somme est sûrement l'exemple le plus

concret. Supposons que nos précédentes *Person* soient des employés, on peut vouloir calculer le salaire total en se basant sur les rémunérations personnelles *getSalary()*. Le partage d'état local est nécessaire pour ce besoin, on ne peut donc pas uniquement se baser sur map.

L'implémentation interne de reduce est dans le cadre d'Hadoop peu intéressante et entièrement à la charge de développeur. Mais il est important de voir qu'un reduce sur une collection peut être envisagé comme une opération de transformation sur un élément (la collection). Lorsque plusieurs collections doivent être transformées en parallèle, il est possible de faire un map de reduces. Cela permet de profiter une fois de plus de la gestion automatique de la parallélisation et de la distribution des tâches par le framework.

> Hadoop Map/Reduce gère des couples clef/valeur

Hadoop Map/Reduce est une implémentation spécifique de map et reduce pour faire du traitement distribué en se basant sur un système de fichier distribué (HDFS). Conceptuellement, une contrainte a été ajoutée : le framework va travailler sur des collections de clefs/valeurs.

Soit 6 types : A, B, C, D, E, F. Map va transformer une collection de (A,B) en une collection de (C,D). Toutes les différentes valeurs D associées à une même clef C vont être regroupées. Et reduce va transformer une collection de (C, collection<D>) en une collection de (E,F). Chaque reduce garantit de traiter en même temps toutes les valeurs D associées à une même clef C ; de recevoir les clefs C dans leur ordre naturel, mais pas nécessairement de recevoir toutes les clefs, puisque le travail est partagé.

Les types sont arbitraires car c'est au développeur de spécifier ce qu'il souhaite. (A,B) est lié à la manière dont le fichier est lu. (C,D) est un format d'échange intermédiaire entre map et reduce. Et (E,F) est le format final, pouvant être relu par un autre job.

Le nombre de map est déterminé par le framework en fonction des fichiers lus.

Par défaut, il y a au minimum une tâche de map par fichier et au maximum une tâche de map par block composant chaque fichier. Étant donné que pour chaque tâche correspondant à un split des données une jvm est lancée, il est pertinent de faire attention à la



ABONNEMENT
PDF

30 € par an soit 2,73 € le numéro
www.programmez.com

Abonnement INTÉGRAL
Pour un supplément de 10 € an
Accès illimité aux archives*

* Option réservée aux abonnés au magazine.

fois aux nombres de fichiers et à la taille des blocks afin de minimiser le surcoût lié à la parallélisation, la distribution et la gestion des erreurs.

C'est une problématique qui peut nécessiter un travail commun de la part du développeur et de l'administrateur.

Le nombre de reduce est quant à lui explicité manuellement et vaut par défaut 1. Le nombre de reduce est lié au volume de [C,D] mais également à leur distribution. Par défaut, le reducer est choisi en fonction de la valeur du hash de la clef.

Si la fonction de hashage est non uniforme ou si une clef possède beaucoup plus de valeurs qu'une autre clef, la charge entre les reducers peut être mal équilibrée. C'est encore une fois une problématique devOps où chaque partie a un rôle critique à jouer.

Un job se compose nécessairement d'une étape de map (qui peut être une identité). Le reducer est cependant optionnel. Certaines problématiques peuvent être traitées exclusivement en parallèle mais pas toutes.

> Qu'est-ce qu'un job?

Un job est essentiellement constitué de deux fichiers. Le premier est un jar (archive java) qui contient le code devant être exécuté, propre à Hadoop. Et le second est un fichier xml contenant sous forme de clef/valeur la configuration du job.

Cela dit, ce fichier est très rarement renseigné tel quel. Les API vont vous permettre de le configurer de façon transparente. Mais il est important de comprendre que c'est par son biais que la configuration (statique) peut être passée aux différentes tâches map et reduce fonctionnant sur des machines différentes.

> Enfin, des API

Hadoop Map/Reduce a un historique. En ce qui concerne la version 1, la version stable actuelle, il existe deux API permettant de faire sensiblement la même chose.

org.apache.hadoop.mapred est l'ancienne API, qui fut un moment dépréciée. Étant donné que beaucoup de documentations mentionnent celle-ci, elle sera utilisée durant cet article. Il faut cependant savoir que *org.apache.hadoop.mapreduce* est une alternative, plus récente. Dans la mesure où les concepts sont assimilés, la migration d'une API vers une autre n'est pas une difficulté en soi.

> Définir un job

Traditionnellement, le premier programme affiche "Hello World". Dans le monde d'Hadoop, il s'agit en fait d'un wordcount : calculer le nombre d'occurrences de chaque mot dans un texte.

La partie map va lire les fichiers ligne à ligne et émettre des couples (mot,1). La partie reduce recevra ensuite l'ensemble des occurrences associées à chaque mot (celui-ci étant la clef) et il suffira de faire la somme.

L'utilisation du **combiner** sera expliquée en fin d'article.

Pour la définition du job, étendre la classe *Configured* et implémenter l'interface *Tool* est une bonne pratique à adopter.

Cela permet de réutiliser le parsing d'options génériques à Hadoop et il est ainsi possible de préciser au lancement du job n'importe quelle clef/valeur devant se retrouver au final dans le fichier xml de configuration.

Par exemple, le nombre de reducer peut être augmenté à 4 en ajoutant simplement "-D mapred.reduce.tasks=4".

Il faut bien noter l'espace entre le D et la propriété. Il ne s'agit pas de

propriétés système propres à la jvm, ces dernières n'ont aucun intérêt ici car le traitement est effectué sur d'autres jvms. *ToolRunner* va parser ces arguments génériques, renseigner la *Configuration* fournie puis appeler la fonction run avec les arguments non interprétés. La configuration récupérée par *getConfig()* possédera bien la configuration additionnelle présente sur la ligne de commande.

Le reste de la configuration est essentiellement sans surprise.

```
public class WordCount extends Configured implements Tool {
    public static final String REMOVE_KEY = «wordcount.remove.regex»;

    public static void main(String[] args) throws Exception {
        int res = ToolRunner.run(new Configuration(), new WordCount(), args);
        System.exit(res);
    }

    public int run(String[] args) throws Exception {
        // préciser une classe permet au client de savoir
        // quel est le jar qui doit être envoyé au cluster
        JobConf conf = new JobConf(getConf(), WordCount.class);
        conf.setJobName(«wordcount»);

        // on précise les types des données
        // sauf pour l'entrée qui est à la charge de l'input format
        conf.setMapOutputKeyClass(Text.class);
        conf.setMapOutputValueClass(IntWritable.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);

        // on définit le mapper et le reducer
        conf.setMapperClass(Map.class);
        conf.setCombinerClass(Reduce.class);
        conf.setReducerClass(Reduce.class);

        // et comment les données sont lues et écrites
        conf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);

        // enfin on parse les arguments pour un argument custom
        List<String> other_args = new ArrayList<String>();
        for (int i = 0; i < args.length; ++i) {
            if («-remove».equals(args[i])) {
                conf.set(REMOVE_KEY, args[++i]);
            } else {
                other_args.add(args[i]);
            }
        }
        // et les fichiers à lire et la destination des résultats
        FileInputFormat.setInputPaths(conf, new Path(other_args.get(0)));
        FileOutputFormat.setOutputPath(conf, new Path(other_args.get(1)));
        // puis on lance le job
        JobClient.runJob(conf);
        return 0;
    }
}
```


> Le format d'échange : Writable

Les données échangées doivent utiliser la sérialisation d'Hadoop et être des *Writable*. Les types primitifs et les String sont gérés par défaut, comme le montre `IntWritable` et `Text` dans cet exemple. S'il est nécessaire d'avoir des types plus complexes, il suffit alors de créer une classe implémentant *Writable* et encapsulant d'autres *Writable* déjà disponibles. Les clefs devront implémenter *WritableComparable*.

> Le mapper et le reducer

L'API pour le mapper et le reducer sont très proches. Dans les deux cas, il faut implémenter une interface *Mapper* ou *Reducer*. Etendre la classe abstraite *MapReduceBase* permet d'éviter de devoir implémenter à vide les deux fonctions du cycle de vie : *configure* et *close*.

Le mapper reçoit en valeur chaque ligne des fichiers un par un. Elles sont découpées et chaque mot est émis un par un. Sa méthode *configure* permet de récupérer le paramètre précédemment renseigné afin de ne pas compter certains mots qui correspondent à une expression régulière. La réutilisation des instances *Writable* permet de soulager la gestion de la mémoire. Et enfin un système de compteurs permet en fin de job d'obtenir un aperçu de son fonctionnement.

```
public class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {

    static enum Counters {
        REMOVED
    }

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    private Pattern pattern;

    @Override
    public void configure(JobConf job) {
        String remove = job.get(WordCount.REMOVE_KEY);
        if (remove != null) {
            pattern = Pattern.compile(remove);
        }
    }

    @Override
    public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
        String line = value.toString().toLowerCase();

        StringTokenizer tokenizer = new StringTokenizer(line);
        while (tokenizer.hasMoreTokens()) {
            word.set(tokenizer.nextToken());
            if (pattern != null && pattern.matcher(line).matches()) {
                reporter.incrCounter(Counters.REMOVED, 1);
            } else {
                output.collect(word, one);
            }
        }
    }
}
```

Le reducer s'explique de lui-même :

```
public class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {
    @Override
    public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
        int sum = 0;
        while (values.hasNext()) {
            sum += values.next().get();
        }
        output.collect(key, new IntWritable(sum));
    }
}
```

> Et le combiner?

Le combiner est une optimisation de performance. Il utilise la même API que le reducer, consomme les données en sortie du mapper et doit produire en sortie le même format. Le framework est responsable de son appel éventuel et potentiellement multiple. Le reducer n'est pas conscient de son existence.

Ici, il permet de regrouper des instances d'un même mot. Transformant `{(mot,1),(mot,1),(mot,1)}` en `{(mot,3)}`, il peut ainsi réduire considérablement le volume de données à transférer depuis la tâche de map vers la tâche de reduce.

> Il ne reste plus qu'à lancer

Le lancement ressemble à l'exécution standard d'un jar :

```
hadoop jar wordcount.jar fr.programmez.WordCount /a/lire /a/ecrire -remove le
```

> La fin du début

Hadoop est un sujet vaste et cet article ne prétend pas pouvoir tout présenter. Il serait possible d'écrire des livres entiers sur ce sujet et cela a été fait. "Hadoop: The Definitive Guide" par Tom White est la référence à consulter.

Il est cependant possible de lister quelques sujets à creuser pour ceux souhaitant aller plus loin. Il faut mentionner l'existence d'un cache distribué pour les fichiers de références couramment utilisés. Le pattern "secondary sort" est intéressant pour mieux comprendre le fonctionnement interne de la plateforme. Maven est un outil permettant de simplifier la gestion de dépendances et MRunit une librairie pour écrire plus facilement vos tests.

Tout cela dit, même si le potentiel de l'API présentée est énorme, il faut aussi se poser la question du niveau d'abstraction. D'autres outils comme Pig, Hive ou Cascading permettent de simplifier considérablement le développement.

Mais la présentation de chaque outil séparément pourrait faire l'objet d'un article en soi.

Pour conclure, tout projet, même informatique, est avant tout une aventure humaine. Hadoop possède des mailing lists et un channel irc. Des groupes utilisateurs existent aussi tout autour de la planète, la France n'étant pas une exception (hugfrance.net).

Bertrand Dechoux
Expert BigData chez Xebia
bdechoux@xebia.fr
<http://xebia.fr>

Comment choisir une stratégie de Build et l'intégrer dans le Cloud

L'intégration continue est une pratique de plus en plus utilisée et les Builds en sont un des éléments les plus importants. L'arrivée de solutions hébergées dans le Cloud permet aujourd'hui de s'affranchir des problèmes d'infrastructures et donc d'apporter de la souplesse dans sa mise en œuvre. Nous allons voir dans cet article comment facilement mettre en place différents types de Builds et plus spécifiquement les Nightly Builds.

Une Build est un processus dont la fonction principale consiste à automatiser l'ensemble des actions contribuant, à partir de données sources, à la production d'un logiciel opérationnel. Avec le temps et l'évolution des techniques, plusieurs autres tâches s'y sont ajoutées telles que l'exécution des tests, la mesure de la qualité du logiciel, la mise en place d'environnements ou encore le déploiement. Ce processus est au cœur du développement et permet de donner une image de l'état d'avancement du logiciel aussi bien d'un point de vue qualitatif que quantitatif. La Build est donc devenue un élément incontournable pour tout développement logiciel et cela, quelle que soit la méthodologie utilisée (cycle en V, RUP, 2TUP, XP, SCRUM,...). Afin d'être exhaustive, elle doit réaliser de manière automatisée et ce aussi souvent que nécessaire les actions suivantes :

- S'assurer que le logiciel compile toujours avec les sources se trouvant dans le contrôleur de code source : parfois, les développeurs ne travaillent pas avec les dernières versions des fichiers et cela arrive que le logiciel compile sur leur poste mais pas avec les versions les plus récentes. La Build s'assure donc de toujours avoir une version opérationnelle du logiciel dans le contrôleur de code source.
- Détecter les bugs et régressions le plus rapidement possible et prévenir le développeur responsable. Cela passe par l'exécution des différents types de tests automatisés (tests unitaires, tests d'intégration, tests fonctionnels,...).
- Calculer les métriques liées à la qualité du produit : identification des duplications de code, mesure du niveau de documentation, respect des règles de programmation, détection des bugs potentiels, évaluation de la couverture de code par les tests unitaires, analyse de la répartition de la complexité et analyse du design et de l'archi-

ture du produit sont autant de données permettant d'évaluer l'état du logiciel.

- Gérer le versionning du produit et dans les cas les plus avancés, générer des packages et les déployer sur les différents environnements: test, intégration, pré-production voire même production.

La plupart de ces tâches ne peuvent être réalisées à la main car elles nécessitent des outils spécifiques. Les autres telles que la gestion des environnements, packaging, déploiement qui sont souvent la responsabilité d'une équipe spécifique, demandent un grand nombre de personnes et sont souvent un frein pour le « Time to Market ». La Build est donc là pour exécuter de manière automatisée l'ensemble de ces tâches sous la forme d'un Workflow et de remonter le maximum d'informations aux différents acteurs. On peut donc en tirer les avantages suivants :

- Permettre à l'équipe de se concentrer sur le développement du logiciel et donc d'augmenter sa productivité,
- Réduire le « Time to Market »,
- Réduire les erreurs humaines sur certaines tâches de packaging, déploiement,
- Augmenter la réactivité grâce à un feedback obtenu plus rapidement,
- Réduire les coûts.

POURQUOI UNE NIGHTLY BUILD

Cas idéal : Intégration Continue

Dans le cas idéal, toutes ces étapes doivent être exécutées à chaque modification de code source (check-in ou commit) pour que le développeur puisse avoir un feedback rapide et complet de l'impact de ses modifications sur le produit et réagir rapidement en cas de régression. De plus, cela permet de toujours avoir un logiciel prêt à être déployé en production, à tout moment, avec les développements les plus récents : C'est ce qu'on appelle l'intégration continue.

Plus concrètement, la mise en pratique de

l'intégration continue n'est pas toujours possible pour les raisons suivantes :

- Exécuter l'ensemble de ces tâches peut prendre beaucoup de temps (parfois plusieurs heures). Le développeur ne connaît donc l'impact de ses modifications que beaucoup trop tard.
- Si l'équipe est trop grande, on obtient un ratio "check-in/temps de Build" trop élevé, ce qui entraîne une saturation au niveau de la Build: on se retrouve donc avec un grand nombre de check-in en attente de validation et donc, un feedback très tardif.
- Le déploiement en continu sur un environnement de tests empêche les tests manuels (souvent, les tests d'intégration, fonctionnels et de performances ne sont pas complètement automatisés). De plus, il n'est pas forcément pertinent de déployer en continu sur ces environnements si l'on n'est en mesure d'effectuer qu'un nombre restreint de tests manuels.
- Le coût logistique : tout faire à chaque check-in consomme beaucoup de CPU, occupe de la place sur le disque et peut saturer le serveur de base de données.

Ces raisons font donc qu'en pratique, il n'est pas toujours possible et pertinent de faire de l'intégration continue bien que cela reste la pratique qui assure la meilleure qualité et la meilleure productivité. L'utilisation de Nightly Builds permet de résoudre cette problématique tout en conservant les avantages de l'intégration continue.

Utilisation de Nightly Builds

La problématique est donc la suivante: si nous implémentons une solution qui consiste à avoir le maximum de feedback (et donc à utiliser l'ensemble des outils disponibles), nous allongeons le temps de réponse du processus de Build. Il en résultera une baisse de l'efficacité car l'attente du résultat sera très longue. Par contre, si nous privilégions le temps de réponse, nous serons obligés

d'écarter certaines tâches et donc, de fait, la qualité de notre logiciel baissera.

Pour répondre à cette double problématique, la solution est assez simple et fait preuve de bon sens: comme nous l'avons dit, il n'est pas pertinent (voire nécessaire) d'avoir l'ensemble des informations à chaque check-in: le packaging et le déploiement en environnement d'intégration ne peut être fait qu'une fois par jour, l'idéal étant la nuit pour éviter toute interruption de service. La qualité globale du code (avec des outils tels que SONAR par exemple) est également une information qu'on remonte une fois par jour et qu'on peut aborder au standup meeting par exemple. On peut également évoquer les tests d'intégration et de performance qui souvent prennent énormément de temps et peuvent également être joués une fois par jour. Par contre, la compilation, l'exécution des tests unitaires les métriques liées à la qualité statique du code (FxCop ou Gendarme) doivent être exécutées à chaque modification car elles peuvent freiner la productivité à court et moyen terme de l'équipe. On va donc adapter notre stratégie à notre outillage et à la taille de notre logiciel en utilisant différents types de Builds :

- Une Build rapide qui valide au minimum la compilation et les tests unitaires et qui est exécutée à chaque check-in. Son but est de remonter le maximum de feedback utile à la validation des modifications et d'assurer que rien n'entrave la productivité de l'équipe.
- Une Build tous les soirs ou « Nightly Build » qui exécute l'ensemble du processus décrit précédemment (compilation, tests, versionning, packaging, déploiement, calcul des métriques...). De cette manière; les équipes de tests ont un ou plusieurs environnements opérationnels tous les jours avec les derniers développements effectués. L'équipe va pouvoir consulter les métriques et prendre les mesures adéquates en cas de régression et les managers vont pouvoir mesurer l'évolution du logiciel jour après jour.

Bien sûr, dans plusieurs situations, nous

sommes obligés d'adapter cette stratégie en ajoutant par exemple d'autres types de Build (pour la gestion des bases de données par exemple). Il est également courant d'avoir des Builds qui s'exécutent à mi-journée : les bugs détectés la nuit peuvent être fixés dans la matinée et déployés par une Build qui aura un Workflow adapté.

PRÉSENTATION DE TEAM FOUNDATION SERVICE

Team Foundation Service est la version hébergée dans le Cloud de TFS. On y retrouve donc la majeure partie des fonctionnalités de TFS dont la partie Build, celle qui nous intéresse aujourd'hui. Cette version SAAS de TFS apporte beaucoup de souplesse dans sa mise en œuvre car elle permet de créer un Team Project en quelques minutes et en quelques clics. De plus, cette solution étant accessible de partout, il est très facile de fédérer une équipe autour de ce produit, d'autant plus si toutes les personnes ne travaillent pas au même endroit. En revanche, le fait que tout ceci soit hébergé dans les datacenters de Microsoft implique qu'il existe un certain nombre de limitations, en particulier au niveau du processus de Build. En effet, il n'est pas possible d'avoir accès au serveur lui-même pour y installer des outils tiers par exemple. Il faut donc avoir conscience de ces limitations :

- Le processus de Build ne doit pas utiliser d'autres logiciels que ceux qui sont présents sur le contrôleur de Build. Il est néanmoins possible de connecter un contrôleur on-premise pour contourner ce problème. La liste des logiciels installés est disponible à l'adresse suivante : <http://listof-softwareontfshostedbuildserver.azurewebsites.net>
- Le projet ne doit pas être une application Windows 8 Store.
- Le temps de Build ne doit pas excéder une heure et l'espace de stockage utilisé ne doit pas dépasser 1GB.
- Le processus de Build ne doit pas avoir

besoin des privilèges administrateurs pour s'exécuter.

- Vous ne devez pas avoir besoin de vous authentifier sur le contrôleur de Build.
- La Build ne doit pas tourner en mode interactif.

PRÉSENTATION DU SCÉNARIO DE BUILD ET MISE EN PLACE

Afin de répondre au besoin précédemment décrit, nous allons mettre en place deux Builds. La première qui se nommera « Développement Build » se déclenchera à chaque check-in et sera en charge de compiler le code et d'effectuer les tests unitaires. La seconde sera en charge d'effectuer la compilation, l'exécution de tous les tests ainsi que les mesures de qualité de code. Elle sera déclenchée à heure fixe toutes les nuits d'où son nom : « Nightly Build » [Fig.1].

Mise en place de la « Développement Build » Cette Build est la plus simple à mettre en place car elle se base sur le Template par défaut et nécessite uniquement de la configuration. Pour pouvoir la créer, il suffit de choisir « New Build Definition » dans la rubrique « Builds » du « Team Explorer ».

Ensuite voici les paramètres à saisir [Fig.2] :

- General : Saisir le nom de la Build « Développement »
- Trigger : Sélectionner Gated Check-in : cela signifie que toute modification sera rejetée si la Build en charge de sa validation échoue.
- Process : Dans cet onglet, il faut filtrer pour ne prendre en compte que la dll contenant les tests unitaires, dans notre cas *Test.TU.dll. Il faut également désactiver l'analyse de code.

Une fois cette Build enregistrée, elle se déclenchera à chaque check-in.

Mise en place de la « Nightly Build »

Contrairement à la Build précédente, l'objectif de celle-ci est d'avoir un retour complet sur la qualité du projet. En effet, en plus de

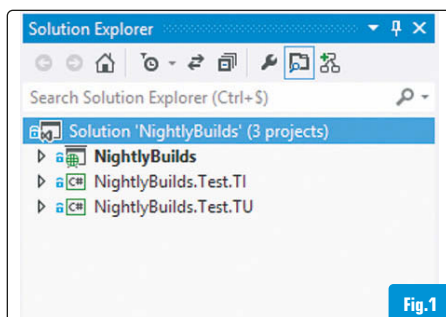


Fig.1

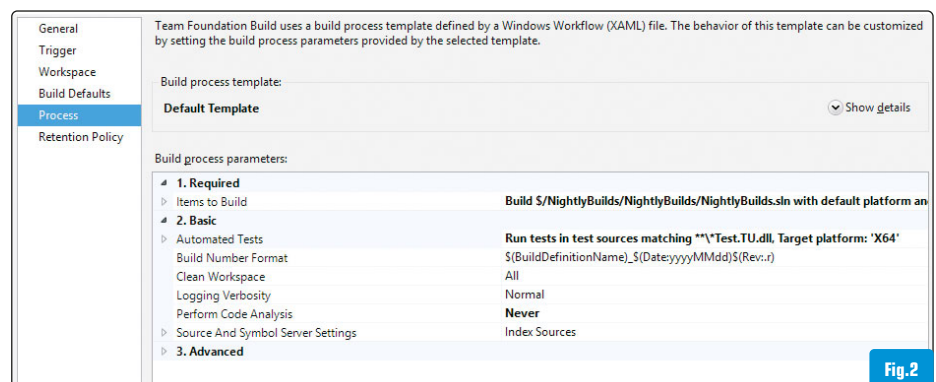


Fig.2

la compilation et des tests unitaires, nous allons vérifier les tests d'intégration ainsi qu'effectuer une analyse du code à l'aide de StyleCop et de l'outil « Code Analysis » équivalente à FxCop. StyleCop n'est pas disponible en standard dans TFS mais il est possible de customiser le Template de Build pour l'inclure en utilisant les composants de StyleCop et ceux du « Community TFS Build Extensions », tous deux disponibles sur CodePlex.

Tout d'abord il faut créer la nouvelle Build, de la même manière que la précédente, avec les paramètres suivants [Fig.3]:

- General : Saisir le nom de la Build « Nightly »
- Trigger: Sélectionner « Schedule » et choisissez l'horaire de la Build.
- Process : Dans cet onglet il faut tout d'abord créer un nouveau Template de Build en sélectionnant « New » et en copiant le Template par défaut. Ensuite il faut choisir « Always » dans la rubrique « Perform Code Analysis ».

Cette première étape effectuée, il faut maintenant passer à la customisation du nouveau Template de Build. Pour ce faire, il faut éditer le Workflow pour y ajouter les différentes briques qui permettront d'effectuer l'analyse StyleCop. Il est possible de customiser entièrement le processus afin de répondre de façon très précise aux différents besoins. Dans notre cas, j'ai donc ajouté les différents composants qui ensemble permettent de répondre à celui que l'on a décrit précédemment : analyser tous les fichiers et produire un rapport contenant la liste des recommandations. L'ajout de l'activité StyleCop est très bien exposé dans la

documentation des extensions TFS, je ne vais donc pas vous la décrire ici. Une fois réalisée vous devriez arriver au résultat suivant [Fig.4].

Le point le plus important à retenir pour que l'analyse fonctionne est que le contrôleur de Build doit avoir accès aux composants de StyleCop pendant le processus de Build. Il faut archiver l'ensemble des composants téléchargés dans un répertoire spécifique du contrôleur de source. Ensuite il faut paramétrer le contrôleur de Builds pour lui donner le chemin d'accès à ces composants. Pour cela, allez dans Team Explorer – Builds et sélectionnez la Build nouvellement créée. Puis dans le menu Action, sélectionnez « Manage Build Controllers ». Ensuite sélectionnez « Hosted Build Controller » et cliquez sur « Properties ». Dans la fenêtre qui s'ouvre, saisissez le chemin des composants dans le champ « Version control path to custom assemblies ».

Un des gros avantages de ce système est que chaque fois que vous allez mettre à jour vos composants et votre template de Build, les Agents seront automatiquement mis à jour : vous n'avez donc pas à gérer le déploiement de vos serveurs de Build [Fig.5]. Lors de la création de la nouvelle Build, le

Workspace de cette dernière sera le même que celui de votre Workspace courant. Si plusieurs projets sont présents dans celui-ci, alors l'analyse StyleCop s'effectuera sur l'ensemble des projets et non pas sur le projet ou la solution qui sera compilé. Il est donc important de bien vérifier l'onglet Workspace lors de la création de la Build afin que du code extérieur au projet ne

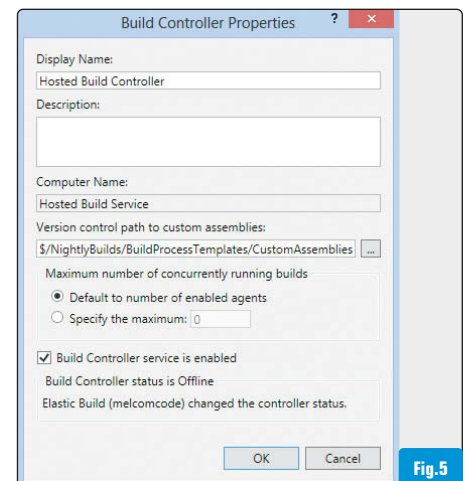
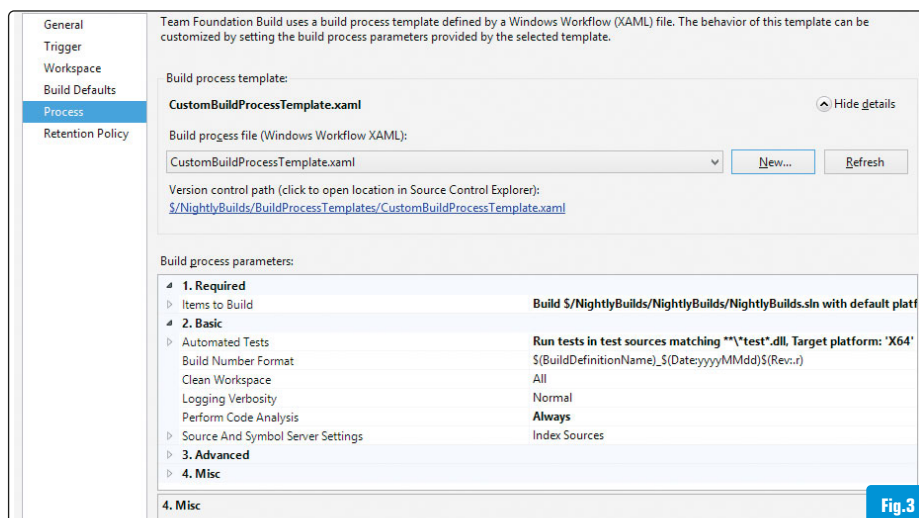
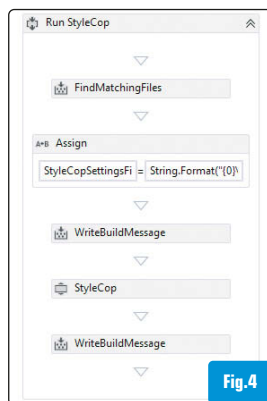
viennne pas perturber les résultats. Cette méthode peut s'appliquer à d'autres outils ou tâches, comme NUnit par exemple. Une fois votre template de Build customisé de manière satisfaisante, vous pouvez également l'utiliser pour d'autres projets ou solutions même s'ils ne se trouvent pas dans le même Team Project. Concernant la partie « Code Analysis », la configuration se fait dans les propriétés du projet dans l'onglet du même nom.

CONCLUSION

La Build est au cœur même du développement logiciel et adopter de bonnes pratiques et une bonne stratégie peut s'avérer être un facteur déterminant dans la réussite de vos projets. Pour les méthodes agiles, en particulier XP, c'est un acteur incontournable car il est impossible de produire un logiciel fonctionnel sur des sprints de 2 semaines sans avoir automatisé un maximum de tâches. Nous avons également pu voir qu'il est désormais possible d'avoir un processus complet de Build dans le Cloud avec Team Foundation Service. Ce dernier reprend le même mode de fonctionnement que la version on-premise et permet donc d'adapter le processus de Build aux besoins de chacun. Cet outil sera un avantage indéniable pour votre équipe car il est complètement intégré aux outils de développements tels que Visual Studio 2012 et offre un grand nombre de rapports permettant de suivre l'évolution de votre logiciel en temps réel.

Fathi Bellahcene
Manager / MVP C# chez Cellenza
Cellenza - Software Development Done Right
Blog: <http://blogs.codes-sources.com/fathi>

Pierre-Henri Gache
Consultant Senior chez Cellenza
Cellenza - Software Development Done Right
Blog : <http://blog.cellenza.com/>



Comprendre et exploiter le système de fichiers /proc de Linux

Vous avez sans doute remarqué que tout Linux récent qui se respecte monte, en général très tôt au démarrage, un système de fichiers nommé /proc. Mais savez-vous que /proc est un formidable outil de suivi et de configuration du système ?

Le répertoire /proc est une entité familière dans le monde d'Unix. On le voit apparaître dès les années 80. Il sert alors à montrer, sous la forme d'une arborescence, les processus du système à un moment donné, d'où son nom **processus mapping**. Linus Torvalds a repris cette idée dès la version 0.99 du noyau, mais très vite, il lui a donné beaucoup plus d'envergure. /proc n'est plus un simple répertoire, mais un système de fichiers à part entière. S'il fournit toujours un instantané des processus dans le système, il va beaucoup plus loin en fournissant un instantané du système entier. /proc peut être compris comme une fenêtre ouverte sur le noyau et son fonctionnement, ce qui permet non seulement un suivi très fin, mais aussi, bien souvent, d'effectuer des configurations du système à la volée.

1 UN SYSTÈME DE FICHIER VIRTUEL

/proc est normalement nécessaire au bon fonctionnement du système Linux. De fait, la plupart des distributions le montent très tôt lors du démarrage. Ainsi c'est la première chose que fait une distribution Gentoo après avoir chargé et démarré le noyau et lancé le processus init, père de tous les autres [Fig.1]. Toutefois, ce système de fichiers ne laisse aucune trace sur aucun des disques de votre système. Il s'agit d'un pseudo-système de fichiers, ou système de fichiers virtuel. Des commandes shell comme ls ou cat fonctionnent, comme si nous avions affaire à des fichiers, mais en fait /proc expose une hiérarchie d'objets résidant en mémoire, et à ces objets, sont associés des fonctions. En réalité /proc est une interface en lecture, et souvent en écriture, sur ce qui se passe dans le noyau. Ce que nous venons de dire s'illustre très simplement par un exemple très classique. Soit la commande :

```
ls -l /proc/version
```

Cette commande retourne :

```
-r--r--r-- 1 root root 0 2012-12-11 19:52 /proc/version
```

```
[ 3.988910] Write protecting the kernel read-only data: 2104k
[ 4.103529] usb 1-1: New USB device found, idVendor=80ee, idProduct=0021
[ 4.103529] usb 1-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
[ 4.103529] usb 1-1: Product: USB Tablet
[ 4.113156] usb 1-1: Manufacturer: VirtualBox
[ 4.155053] input: VirtualBox USB Tablet as /devices/pci0000:00/0000:00:06.0/usb1/1-1/1-1:0/input/input4
[ 4.157066] generic-usb 0003:80EE:0021.0001: input,hidraw0: USB HID v1.10 Mouse [VirtualBox USB Tablet] on usb-0000:00:06.0-1/input0
INIT: version 2.88 booting
[ 5.200038] kbd_mode used greatest stack depth: 6704 bytes left
[ 5.205050] loadkeys used greatest stack depth: 5052 bytes left
[ 5.209059] init-early.sh used greatest stack depth: 5724 bytes left

OpenRC 0.9.8.4 is starting up Gentoo Linux (i686)
* Mounting /proc ... [ ok ]
* Mounting /run ... [ ok ]
* /run/lock: creating directory [ ok ]
* /run/lock: correcting owner [ ok ]
* Mounting /sys ... [ ok ]
* Mounting debug filesystem ... [ ok ]
* Mounting cgroup filesystem ... [ ok ]
* Mounting /dev ... [ ok ]
```

Fig.1

Monter /proc est la première chose que fait un système Gentoo Linux

On voit clairement que ce "fichier" version a une taille nulle. Logiquement, il ne devrait rien contenir. D'autant plus que la commande `file` vous confirme bien que ce fichier est vide :

```
file /proc/version
/proc/version: empty
```

Cependant nous pouvons lire le contenu de ce fichier vide :

```
cat /proc/version
Linux version 3.0.0-12-generic (build@vernadsky) (gcc version 4.6.1
(Ubuntu/Linaro 4.6.1-9ubuntu3) ) #20-Ubuntu SMP Fri Oct 7 14:50:42 UTC 2011
```

Car cat invoque l'appel système `read`, ce qui aboutit à l'appel d'une routine associée au fichier /proc/version. Donc dans /proc nous n'avons pas de fichiers, mais tout se passe comme si nous en avions, dans une belle arborescence, comme une commande `ls` le montre [Fig.2]. En bon pseudo-fichiers qu'ils sont, les objets résidant dans /proc se voient attribués des droits en lecture/écriture. A quelques exceptions près, root peut lire tous les fichiers, et écrire dans certains. Les droits de lecture/écriture peuvent être visualisés de manière tout à fait classique avec une commande `ls -l`. Un utilisateur lambda ne pourra accéder qu'à certains des éléments de /proc, toujours en accord les droits d'accès.

2 SAVOIR SE DOCUMENTER

Tous les fichiers de /proc n'ont pas des contenus aussi humainement lisibles que /proc/version, loin s'en faut. Il faut donc se documenter pour en comprendre le contenu. La première chose à faire est de consulter la page de man pour proc (faire `man proc` ou `man 5 proc`). Ensuite pour approfondir, il faut lire la documentation du noyau, ce qui d'abord implique d'en télécharger les sources via votre

```
root@mars: ~
root@mars:~# ls /proc
1      18      2355    2524    3522    772      dri          partitions
10     1816    2356    2536    3523    781      driver       sched_debug
1009   1839    2360    2541    3575    8        execdomains  schedstat
1058   1872    2363    2543    3582    801      fb           scsi
11     19      2365    2553    3586    803      filesystems  self
1151   1929    2369    2559    3591    810      fs           slabinfo
1163   1983    2374    2560    36      822      interrupts  softirqs
1177   1997    2377    2570    39      830      iomem       stat
1185   2      2379    2575    395     831      ioports     swaps
12     20      2380    2613    398     882      irq         sys
1257   21      2383    2691    421     9        kallsyms    sysrq-trigger
1259   2130   24      2692    4380    904      kcore       sysvipc
1262   214   2414   2693    439     910      key-users   timer_list
1264   2147   2422   2694    447     915      kmsg        timer_stats
13     2155   2424   2695    448     916      kpagecount  tty
1301   2160   2426   270    449     917      kpageflags  uptime
1326   2163   2428   272    464     918      buddyinfo   version
1346   2169   2431   3      467     919      bus          loadavg
1348   2171   2434   3257    5      920      cgroups     locks
1355   22      2467   3264    5245    921      cmdline     meminfo
1377   2208   2475   3266    535    922      consoles   vmstat
14     2241   2484    33     561     923      cpufreq     vmlinux
15     23      2490   3324    568    924      crypto      modules
16     2305   2496   3325    6      925      devices     mounts
17     2345   25      3330    660    926      device-tree nrtt
1769   2347   2502   3470    661    927      diskstats   net
1799   2351   2504    35      7      928      dma         pagetypeinfo

root@mars:~#
```

Fig.2

Avec /proc tout se passe comme si nous avions une arborescence de répertoires et de fichiers.

distribution. Consultez ensuite le fichier `proc.txt` situé dans `Documentation/filesystems`. Pendant que vous y êtes, le fichier `sysfs.txt`, relatif au système de fichiers `/sys` peut aussi vous intéresser, mais cela sort du cadre de cet article. Si vous voulez monitorer les activités réseau de votre système, vous aurez sans doute besoin de consulter `ip-sysctl.txt` dans `Documentation/networking`. Enfin outre le côté geek et ludique du travail avec `/proc`, il n'est peut-être pas toujours nécessaire de réinventer la roue. De nombreuses commandes du shell, comme `top` ou `lsmod`, ne font rien d'autre que lire dans `/proc` et de traduire le résultat de leur lecture. Cela dit, bidouiller `/proc` soi-même permet de se forger des outils personnalisés très pointus, et c'est très amusant :-)

3 EXAMINER DES PROCESSUS

Nous savons que sous Linux les processus se voient attribués un nombre en guise d'identifiant par le système (PID). C'est bien entendu la même chose au sein de `/proc`. A coup sûr vous trouverez un répertoire de nom 1, correspondant au processus `init`. Faites :

```
cd /proc/1
```

puis ls [Fig.3] Vous constatez que le processus est lui-même matérialisé par une arborescence. Dans cette arborescence vous trouverez des liens symboliques, comme `'exe'` qui pointe sur le fichier exécutable correspondant au processus. Le fichier `maps`, comme son nom le suggère donne le mapping mémoire du processus

```
cat maps
00110000-00118000 r-xp 00000000 08:01 1032708      /lib/i386-linux-gnu
/libnss_compat-2.13.so
00118000-00119000 r--p 00007000 08:01 1032708      /lib/i386-linux-gnu
/libnss_compat-2.13.so
00119000-0011a000 rw-p 00008000 08:01 1032708      /lib/i386-linux-gnu
/libnss_compat-2.13.so
```

Etc. Vous obtiendrez l'état de la pile comme ceci

```
cat stack
[<c1138aee>] poll_schedule_timeout+0x3e/0x60
[<c113953e>] do_select+0x47e/0x570
[<c113978c>] core_sys_select+0x15c/0x2b0
[<c1139a94>] sys_select+0x84/0xb0
[<c152c8e4>] syscall_call+0x7/0xb
[<ffffffff>] 0xffffffff
```

En plus des fichiers et liens vous trouverez des répertoires. Par exemple `fd` qui, comme son nom le suggère, contient les descripteurs de fichiers ouverts par le processus. Nous comprenons que les informations qui peuvent être extraites de `/proc` sont d'une grande richesse. Il n'y a plus qu'à faire travailler son imagination pour

créer des utilitaires sympathiques. Nous n'irons pas plus loin dans l'aspect lecture de `proc`, car la documentation est relativement abondante, en ce qui concerne l'examen des processus, ou la charge CPU. Ces possibilités sont d'ailleurs couramment exploitées par les hébergeurs de serveurs pour rendre compte de la charge des machines. Nous allons maintenant nous concentrer sur des recoins peu connus de `/proc`, dans lesquels il est intéressant d'écrire, et qui impliquent plus de programmation.

4 CONFIGURER UN NOM D'HÔTE À LA VOLÉE

Ecrire dans `/proc` permet de configurer le système à la volée, très rapidement et très simplement. Un exemple très simple est la reconfiguration du nom d'hôte. Soit un système configuré ainsi :

```
cat /etc/hostname
mars
```

La commande shell `hostname` retourne donc `mars`. Dans `/proc` nous avons fort logiquement :

```
cat /proc/sys/kernel/hostname
mars
```

maintenant faisons :

```
echo fred > /proc/sys/kernel/hostname
```

Nous pouvons maintenant vérifier que le nom de notre machine a bien été modifié :

```
hostname
fred
```

Il est à remarquer qu'il existe une commande pour modifier le contenu des fichiers du sous-répertoire `sys` de `/proc`. Cette commande, `sysctl`, est bienvenue lorsqu'il s'agit d'agir sur des fichiers dont le contenu est plus complexe qu'une simple valeur.

5 ACCÉDER À LA CONFIGURATION DU NOYAU

Voici maintenant quelques possibilités que `/proc` offre aux geeks qui utilisent volontiers plusieurs noyaux sur une même machine. Tout d'abord, il est possible de connaître la ligne de commande de noyau, c'est-à-dire les paramètres qui lui sont passés par le chargeur (`lilo`, `grub`...) au moment du démarrage : Par exemple sur un Ubuntu :

```
cat /proc/cmdline
root=UUID=284dae9c-3984-4ea6-aaef-0e121ee13b90 ro quiet splash
```

ou sur un Gentoo :

```
cat /proc/cmdline
root=/dev/sda3
```

Si le noyau est compilé avec les options qui vont bien, il est possible de retrouver ces options dans `/proc`. Cette possibilité se révèle très utile si vous bidouillez avec plusieurs noyaux et qu'à un instant t vous ne savez pas dire si le noyau en cours d'utilisation supporte telle fonctionnalité, si c'est sous la forme d'un module, etc. Pour cela, lorsque vous configurez la compilation de votre noyau, sous 'General setup', cochez les options 'Kernel .config support' et 'Enable access to .config through `/proc/config.gz`' [Fig.4]. Ceci fait, compilez votre noyau puis démarrez votre système avec lui. Un fichier `config.gz` est

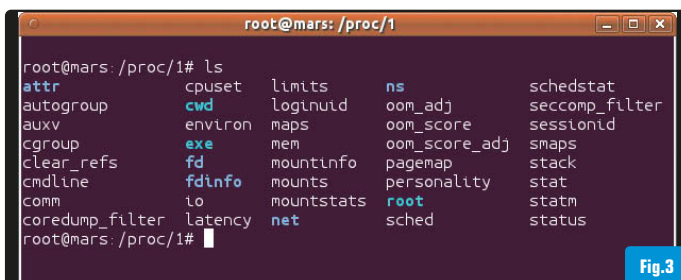
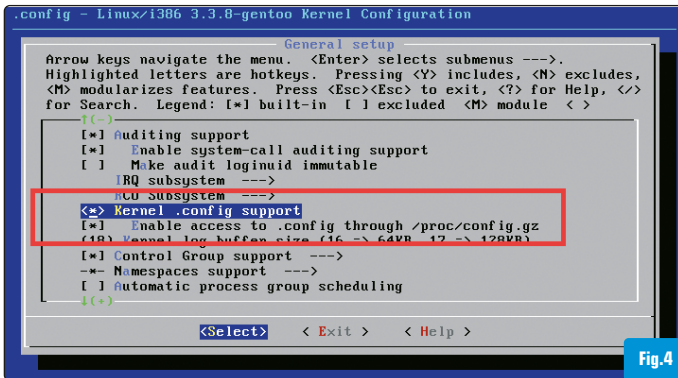


Fig.3

Dans `/proc` le processus `init` est matérialisé par une arborescence.



Il est possible d'intégrer la configuration d'un noyau à celui-ci lors de sa compilation.

apparu sous `/proc`. Il s'agit d'un fichier compressé. Il est possible de consulter facilement son contenu via la commande `zgrep`. Voulez-vous savoir si votre noyau supporte le système de fichier ext2 ?

```
zgrep EXT2 /proc/config.gz
CONFIG_EXT2=y
# CONFIG_EXT2_FS_XATTR is not set
CONFIG_EXT2_FS_XIP is not set
```

6 CHANGER UN INTERPRÉTEUR À LA VOLÉE

Cette possibilité se révèle par exemple intéressante lorsque vous écrivez des scripts complexes, et que vous vous posez des questions quant à leur compatibilité avec différentes versions de l'interpréteur du langage. Il est possible via `/proc` de définir un interpréteur qui sera lancé soit à partir du nombre magique du fichier, soit à partir de son extension, et ceci, même si un shebang est déclaré dans le script. Pour l'exemple prenons un script Python `hello.py`, qui a été rendu exécutable par `chmod +x hello.py`.

```
#!/usr/bin/env python

maliste = range(5)

for i in maliste:
    print i
```

Pour la démonstration, modifions le shebang :

```
#!/usr/bin/env perl
```

Il est évident que le script fonctionnera maintenant beaucoup moins bien :) Mais cela nous permet de démontrer qu'il est possible de redéfinir l'interpréteur de façon à ce qu'il soit prioritaire pour exécuter des scripts d'extension `.py`.

```
echo :Python:E::py::/usr/bin/python: > /proc/sys/fs/binfmt_misc/register
```

et vous constaterez qu'un `./hello.py` exécute le script en dépit de son shebang :)

7 NE PAS CÉDER À LA PANIQUE

Supposons que vous développiez un module, très probablement il va vous arriver, au cours de la mise au point, que le noyau panique. C'est-à-dire qu'il ne sache pas traiter une situation et termine avec l'appel système `panic` qui bloque tout. Vous êtes alors obligé d'éteindre brutalement votre machine, ce qui n'est pas satisfaisant.

Il est pourtant possible d'éviter cela. Commençons par écrire un module qui crée le plantage (les sources de tous nos exemples sont disponibles sur notre site).

```
/* fichier modulepanic.c */

#include <linux/module.h>
#include <linux/init.h>
#include <linux/kernel.h>

static int debut(void)
{
    panic("Houla la c'est la panique !");
    return 0;
}

static void fin(void)
{
    return;
}

module_init(debut);
module_exit(fin);
```

A partir du noyau 2.6 il y a une procédure spéciale pour compiler un module. Cette procédure repose sur l'outil `kbuild` que vous devez avoir installé sur votre système. Ensuite, vous devez écrire un `makefile` spécifique.

```
obj-m += modulepanic.o

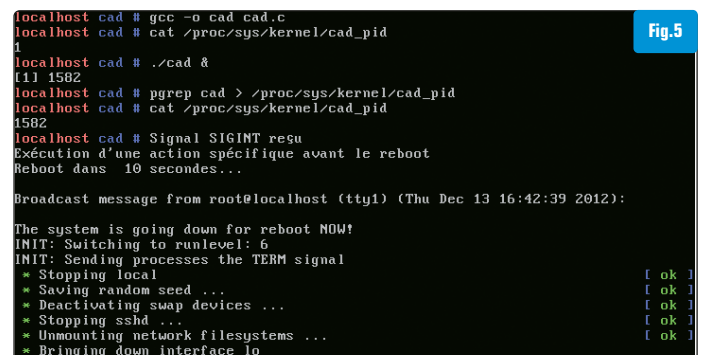
KVERSION = $(shell uname -r)
all:
    make -C /lib/modules/$(KVERSION)/build M=$(PWD) modules
clean:
    make -C /lib/modules/$(KVERSION)/build M=$(PWD) clean
```

Pour compiler, il suffit alors de lancer `make` de façon tout à fait classique. Il est à remarquer qu'il n'est pas nécessaire que le fichier source du module soit déposé dans l'arborescence de sources du noyau. Une fois le module compilé, chargez le dans le noyau. Vous devez bien entendu avoir les droits du super utilisateur pour cela :

```
insmod modulepanic.ko
```

Et là votre système est complètement gelé. Vous devez éteindre votre machine. Une fois celle-ci redémarrée, faites d'abord :

```
echo 10 > /proc/sys/kernel/panic
```



Insertion d'une tâche personnalisée dans le processus de reboot initié par Ctrl-Alt-Del.

puis chargez le module avec la même commande que précédemment :

```
insmod modulepanic.ko
```

Si votre système est configuré pour cela vous verrez sur la console :

```
Kernel panic: Houla la c'est la panique !
<0>Rebooting in 10 secondes..
```

Si vous ne voyez rien, gardez confiance, dans 10 secondes votre système redémarrera:-)

8 PERSONNALISER UN REDÉMARRAGE

Au sujet du redémarrage, il est intéressant de savoir qu'il est possible de personnaliser le comportement du système à la suite de l'appui sur la combinaison de touches Ctrl-Alt-Del. Comme nous le savons, cette combinaison provoque le reboot du système. Mais attention, si ce que nous allons dire s'applique parfaitement sur un Linux en environnement terminal, il est probable qu'un environnement graphique interfère. En ce qui concerne la réaction à cette combinaison de touches, /proc offre deux variables rigolotes à bidouiller. Tout d'abord :

```
cat /proc/sys/kernel/ctrl-alt-del
0
```

Si l'on change cette valeur par défaut de 0 pour la valeur 1, le système rebootera immédiatement. C'est-à-dire que l'arrêt des services et autres processus sera court circuité, pour relancer immédiatement le noyau. Le procédé est brutal car le système de fichiers ne sera pas démonté proprement, mais il est intéressant si un processus gèle lors de la phase arrêt du redémarrage. Que se passe-t-il normalement avec la valeur 0 par défaut ? Le système envoie un signal SIGINT, l'équivalent d'un Ctrl-C, au processus dont le PID est stocké dans la seconde variable rigolote cad_pid :

```
cat /proc/sys/kernel/cad_pid
1
```

Par défaut cette variable contient 1 qui est le processus init, père de tous les autres. Quand init reçoit le signal SIGINT, le système redémarre. Ou, pour être précis, c'est l'action définie pour ctrlaltdel dans le fichier /etc/inittab qui est exécutée. Comme cette action est par défaut "shutdown -r now", le système redémarre :-). Si l'on change la valeur du PID contenu dans cad_pid pour le PID d'un programme de notre cru, c'est celui-ci qui sera exécuté à l'appui sur la combinaison de touche Ctrl-Alt-Del, ce qui permet de greffer des actions personnelles sur la phase de redémarrage. Voici un exemple minimaliste en langage C qui illustre cela :

```
/* fichier cad.c */
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>

void gestionnaire(int num_signal)
{
    puts("Signal SIGINT reçu");
}
```

```
int main(void)
{
    char *argv[] = {"shutdown", "-r", "now", NULL};

    signal(SIGINT, gestionnaire);
    pause();
    puts("Exécution d'une action spécifique avant le reboot");
    puts("Reboot dans 10 secondes...");
    sleep(10);
    execvp("shutdown", argv);
}
```

Pour que cela fonctionne, le programme devra être lancé avec les droits de root, bien évidemment, car seul root peut redémarrer le système. Pour redémarrer, notre exemple lance la commande shutdown. Il est ici tentant d'envoyer un signal SIGINT à init, mais cela ne peut fonctionner car le système l'interdit. Compilez le programme et essayez-le comme suit :

```
gcc -o cad cad.c
cat /proc/sys/kernel/cad_pid
1
./cad &
pgrep cad > /proc/sys/kernel/cad_pid
```

et là, appuyez sur Ctrl-Alt-Del. Vous observerez le comportement attendu, comme illustré [Fig.5].

9 SYSRQ ET SON DÉCLENCHEUR

Il est possible d'agir sur le système avec de nombreuses autres combinaisons de touches. Pour cela il est nécessaire que la variable sysrq soit positionnée à 1 :

```
echo 1 > /proc/sys/kernel/sysrq
```

Après quoi sont disponibles une quantité de combinaisons de touches :

```
Alt - Impécr - autre_touche
```

Il est à noter, pour ces combinaisons de touches, que le système lit un clavier QWERTY. Toutes les combinaisons de touches sont documentées dans le fichier Documentation/sysrq.txt dans les sources du noyau. Ce document est à consulter absolument. Par exemple, avec p (en guise de autre_touche) vous obtenez les registres de CPU0. Avec i vous tuez toutes les tâches sauf init. Avec t vous avez un état détaillé de toutes les tâches. Avec l, une trace de la pile pour tous les CPU, etc. Que signifie obtenir ? Les informations sont écrites dans le log du système. Cependant, si vous réglez le niveau de sortie avec la touche qui va bien, les informations sortent sur la console. Cette touche est la touche 8. Bien sûr vous penserez que c'est très bien ces combinaisons de touches, mais cela donne des crampes. Justement il se trouve que cette interface est programmable via la variable sysrq-trigger. Ainsi pour régler le niveau de sortie et lister l'état de toutes les tâches vous ferez simplement :

```
echo 8 > /proc/sysrq-trigger
echo t > /proc/sysrq-trigger
```

Linux est un système d'exploitation formidable ! :-)

Frédéric Mazué - fmazue@programmez.com

Spring Security : détecter les timeouts de session utilisateur lors d'appels Ajax

Le framework Spring Security s'est imposé comme la solution de référence pour la sécurisation côté serveur des applications Java EE. Très puissant, le framework trouve néanmoins ses limites dans la détection des timeouts de session utilisateurs lors d'appels Ajax.

PROBLÈME

De nos jours, les appels Ajax sont légion sur les applications web Java EE basées sur JSF ou des services REST. De même, les applications mobiles utilisent massivement ce type d'appels pour récupérer des données côté serveur. Bien souvent, le backend serveur nécessite une protection particulière afin de sécuriser les données. Sur des applications Java EE, cette sécurité est mise en place à l'aide du framework Spring Security. Aussi puissant que soit ce dernier, lorsque le timeout de session utilisateur est dépassé, les appels Ajax se produisant par la suite auront pour effet de bloquer la partie IHM de l'application puisque le retour de ces requêtes n'indique en rien que l'utilisateur est désormais déconnecté. De fait, la partie client exécutant la requête ne pourra réagir correctement.

ANALYSE

Ce problème nécessite tout d'abord d'analyser les appels Ajax lancés depuis la partie cliente. Sur une application web, cela peut se faire par exemple sous Chrome en utilisant les Web Developer Tools inclus au sein du navigateur. Sur une application mobile, cela nécessite d'effectuer un debug pas à pas de l'application et de ces appels HTTP vers le serveur. L'examen des appels Ajax réalisés vers le serveur montre clairement que la requête ne renvoie aucun résultat mais s'exécute correctement puisque le serveur retourne 200 comme code HTTP.

Un debug côté serveur de l'application s'avère nécessaire afin de mieux comprendre le fonctionnement de Spring Security lors des appels Ajax réalisés une fois le timeout de session dépassé. Pour observer le comportement lors de la réception de requêtes et éventuellement effectuer des traitements spécifiques, le framework propose de définir un bean de type `GenericFilterBean` au sein du fichier de configuration de Spring Security comme suit :

```
<http auto-config="true" use-expressions="true">
  ...
  <custom-filter ref="ajaxTimeoutRedirectFilter" after="EXCEPTION_
    _TRANSLATION_FILTER" />
  <session-management invalid-session-url="/login.jsf" />
</http>
<beans:bean id="ajaxTimeoutRedirectFilter" class="fr.appli.filter.
  AjaxTimeoutRedirectFilter" />
```

Ainsi, au sein de la méthode `doFilter` de la classe `AjaxTimeoutRedirectFilter`, héritant de `GenericFilterBean`, nous allons pouvoir mieux étudier les requêtes traitées côté serveur. En reproduisant un appel Ajax vers le serveur placé en mode debug une fois le timeout de session utilisateur dépassé, on s'aperçoit que la requête alors traitée par le serveur concerne la page `login.jsf`. Cela est d'autant plus étrange que nous avons requêté `page.jsf` depuis la partie cliente de l'application. L'explication tient au fait que Spring Security constate

bien que la session utilisateur a expiré et qu'il redirige donc la requête vers la page `login.jsf` comme défini dans le fichier de configuration. Côté client, au niveau de l'objet `XMLHttpRequest`, il n'est pas possible de détecter cette redirection et donc d'être averti directement de l'expiration de la session. Ce comportement est bien en accord avec les spécifications du W3C qui précisent que la redirection doit être transparente pour l'utilisateur final. Ici, c'est bien le cas et c'est ainsi qu'un code retour HTTP 200 est renvoyé côté client puisque la page `login.jsf` est correctement requêtée.

SOLUTION

Ne pouvant pas se baser sur une détection de redirection HTTP au niveau Javascript côté client, nous allons devoir effectuer un traitement côté serveur pour réaliser cette détection. L'idée ici consiste à s'appuyer sur l'`AjaxTimeoutRedirectFilter` défini précédemment pour interagir avec Spring Security lors du traitement des requêtes. En détectant la redirection HTTP à ce niveau, il sera alors possible de retourner un code HTTP spécifique permettant d'avertir le côté client de la redirection. A charge ensuite à ce dernier de réagir comme il se doit à ce code d'erreur. L'algorithme permettant de détecter la redirection a l'allure suivante :

```
SI utilisateur non authentifié
  Redirection vers la page de login
SI accès à la ressource demandée refusé
  SI session utilisateur a expiré ET requête de type Ajax
    Renvoyer un code d'erreur HTTP spécifique
  SINON
    Redirection vers la page de login
  FIN SI
SINON
  Redirection vers la page de login
FIN SI
```

IMPLÉMENTATION SERVEUR

Une partie de l'algorithme ainsi décrit est déjà implémentée au sein de Spring Security en ce qui concerne la redirection vers la page de login lorsque l'utilisateur n'est plus authentifié. Le reste de l'implémentation se fait comme suit :

```
public class AjaxTimeoutRedirectFilter extends GenericFilter
  Bean {
  private ThrowableAnalyzer throwableAnalyzer = new DefaultThrow
    ableAnalyzer();
  private AuthenticationTrustResolver authenticationTrustResolver
    = new AuthenticationTrustResolverImpl();
  private int customSessionExpiredErrorCode = 666;

  @Override
```



```

public void doFilter(ServletRequest request, ServletResponse
response, FilterChain chain) throws IOException, ServletException {
    try {
        chain.doFilter(request, response);
    } catch (IOException ex) {
        throw ex;
    } catch (Exception ex) {
        Throwable[] causeChain = throwableAnalyzer.determineCauseChain(ex);
        RuntimeException ase = (AuthenticationException) throwable
Analyzer.getFirstThrowableOfType(
AuthenticationException.class, causeChain);

        if (ase == null) {
            ase = (AccessDeniedException) throwableAnalyzer.getFirst
ThrowableOfType(AccessDeniedException.class, causeChain);
        }

        if (ase != null) {
            if (ase instanceof AuthenticationException) {
                throw ase;
            } else if (ase instanceof AccessDeniedException) {
                if (authenticationTrustResolver.isAnonymous(SecurityContext
Holder.getContext().getAuthentication())) {
                    String ajaxHeader = ((HttpServletRequest) request).getHeader
("X-Requested-With");

                    // Appel Ajax
                    if ("XMLHttpRequest".equals(ajaxHeader)) {
                        HttpServletResponse resp = (HttpServletResponse) response;
                        resp.sendError(this.customSessionExpiredErrorCode);
                    } else {
                        // redirection login ...
                        throw ase;
                    }
                } else {
                    throw ase;
                }
            }
        }
    }
}

private static final class DefaultThrowableAnalyzer extends Throw
ableAnalyzer {
    protected void initExtractorMap() {
        super.initExtractorMap();
        registerExtractor(ServletException.class, new ThrowableCause
Extractor() {
            public Throwable extractCause(Throwable throwable) {
                ThrowableAnalyzer.verifyThrowableHierarchy(throwable, Servlet
Exception.class);
                return ((ServletException) throwable).getRootCause();
            }
        });
    }
}
...
}

```

Le contenu de cette classe est repris de la classe `ExceptionHandler` de Spring Security avec l'ajout de la détection des appels Ajax. Dans ce cas, l'exécution de la requête est interrompue et une réponse est alors envoyée au client avec un code d'erreur HTTP défini au niveau de la classe et devant nécessairement être différent des codes HTTP existants. La classe ainsi créée est définie au sein du fichier de configuration Spring Security d'une application web Java EE de la sorte :

```

<http auto-config="true" use-expressions="true">
    ...
    <custom-filter ref="ajaxTimeoutRedirectFilter" after="EXCEPTION
_TRANSLATION_FILTER" />
    <session-management invalid-session-url="/login.jsf" />
</http>

<beans:bean id="ajaxTimeoutRedirectFilter" class="fr.appli.filter.
AjaxTimeoutRedirectFilter">
    <beans:property name="customSessionExpiredErrorCode" value=
"666" />
</beans:bean>

```

On remarque que le filtre est placé après l'`EXCEPTION_TRANSLATION_FILTER` de Spring Security dans la chaîne d'exécution des requêtes côté serveur.

IMPLÉMENTATION CLIENT

Avec notre implémentation côté serveur, les appels Ajax réalisés une fois le timeout de session dépassé renverront un code HTTP d'erreur spécifique. Côté client, le travail va de fait consister à configurer une fonction Javascript appelée à chaque fois que ce code d'erreur HTTP sera retourné suite à des appels Ajax. Dans le cas d'une application web utilisant JQuery, le code Javascript à inclure au sein des pages est le suivant :

```

<script type="text/javascript">
    function ajaxSessionTimeout()
    {
        // gestion du timeout
        ...
    }

    !function($) {
        $.ajaxSetup({
            statusCode : {
                666 : ajaxSessionTimeout
            }
        });
    }(window.jQuery);
</script>

```

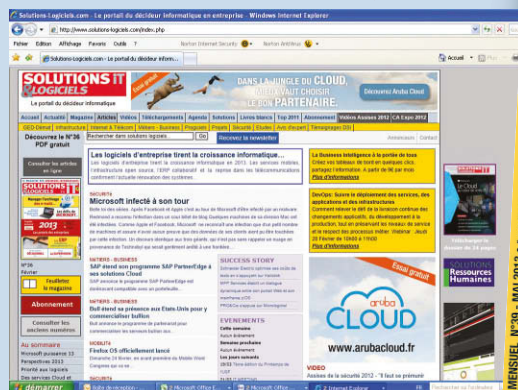
La fonction `ajaxSetup` de JQuery permet de définir la méthode à appeler lorsque le code de retour HTTP 666 sera renvoyé pour l'ensemble des requêtes Ajax qui seront envoyées depuis cette page. Basée sur un traitement côté serveur, notre solution permet ainsi de résoudre le problème de détection des timeouts session utilisateur lors d'appels Ajax dans le cas d'applications utilisant un backend sécurisé via Spring Security.

Sylvain Saurel – Ingénieur d'Etudes Java / JEE - sylvain.saurel@gmail.com

Les outils des Décideurs Informatiques

*Vous avez besoin d'info
sur des sujets
d'administration,
de sécurité, de progiciel,
de projets ?
Accédez directement
à l'information ciblée.*

Cas clients
Actu triée par secteur
Avis d'Experts



Actus / Evénements | Newsletter | Vidéos



www.solutions-logiciels.com

☐ **OUI, je m'abonne** (écrire en lettres capitales)

Envoyer par la poste à : Solutions Logiciels, service Diffusion, GLIE - 17 chemin des Boulangers 78926 Yvelines cedex 9 - ou par fax : 01 55 56 70 20
1 an : 50€ au lieu de 60€, prix au numéro (Tarif France métropolitaine) - Autres destinations : CEE et Suisse : 60€ - Algérie, Maroc, Tunisie : 65€ , Canada : 80€ - Dom : 75€ Tom : 100€
10 numéros par an.

☐ M. ☐ Mme ☐ Mlle Société

Titre : Fonction : ☐ Directeur informatique ☐ Responsable informatique ☐ Chef de projet ☐ Admin ☐ Autre

NOM Prénom

N° rue

Complément

Code postal : Ville

Adresse mail

☐ Je joins mon règlement par chèque à l'ordre de SOLUTIONS LOGICIELS ☐ Je souhaite régler à réception de facture

Multi-Plateforme

Bureau

Créez des applications d'entreprise performantes, scalables et conçues pour le tactile sur la plateforme de votre choix.

Construisez des expériences tactiles qui s'appuient sur les standards avec HTML 5 et JQuery, pour le bureau, les tablettes et les smartphones ; ou en ciblant plusieurs de ces appareils grâce à des frameworks comme PhoneGap et MVC.



Natif pour Mobiles

Développez des applications riches, spécifiques à chaque expérience utilisateur sur iOS, sur Android, et sur Windows Phone ; aussi bien que des applications multi-plateformes avec Mono-Touch.



Prototypage d'Interfaces

Explorez des idées de design en réalisant rapidement des Wireframes centrés sur l'utilisateur, en prototypant et en faisant des évaluations avant d'écrire la moindre ligne de code.



Téléchargez votre version d'évaluation gratuite
infragistics.com/telechargez



Infragistics Ventes France  0800 667 307 • Infragistics Ventes Europe +44 (0) 800 298 9055

Copyright 1996-2013 Infragistics, Inc. All rights reserved. Infragistics and NetAdvantage are registered trademarks of Infragistics, Inc. The Infragistics logo is a trademark of Infragistics, Inc.