

Parcours Développeur d'application - Python

P12 – Construisez une veille technologique

Ce document a pour objectif de présenter ma réalisation d'un outil de veille technologique dans le cadre du projet 12 du parcours développeur d'application (Python).

Le but de ce projet est de réaliser un outil permettant de centraliser les sources d'informations relatives aux technologies sur lesquelles doit se tenir à jour un développeur fullstack Python.

1. Outils utilisés

Plusieurs outils peuvent-être utilisés pour mettre en œuvre une veille technologique. Parmi ces outils on peut citer les lecteurs de flux RSS, permettant de réaliser des tableaux de bord structurés avec des sources d'informations choisies, ou encore utiliser des réseaux sociaux tels que Twitter pour suivre différentes sources d'informations et influenceurs. Dans mon cas, j'ai choisi d'utiliser Twitter pour réaliser ma veille technologique et d'améliorer la gestion de mon fil d'actualité grâce à l'outil Tweetdeck afin de me construire un tableau de bord.

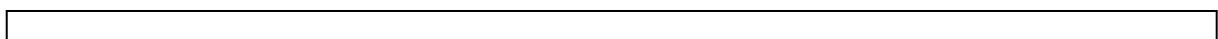
1.1 Twitter

J'ai choisi d'utiliser Twitter pour plusieurs raisons. Tout d'abord c'est un outil que j'avais l'habitude d'utiliser à titre personnel, son fonctionnement m'était donc familier. De plus, ayant l'habitude de le consulter quotidiennement, cela me permettait d'être rapidement opérationnel sur la veille. Outre le fait que je sois familier avec l'outil, l'avantage qu'a Twitter est la notion de partage des sources. En effet, le système de « Retweet » permet de découvrir de nouvelles sources pertinentes car partagées par des fils d'actualités que l'on a choisi de suivre. Cela permet donc d'élargir facilement ses sources d'informations. Il faut cependant faire attention à ne pas tomber dans la facilité en ne suivant que de nouvelles sources issues de retweet afin de garder une forme d'objectivité lors de sa veille technologique.

1.2 Tweetdeck

L'utilisation de Twitter est efficace pour pouvoir suivre un certain nombre de sources, cependant son utilisation seule est un peu limitée car le réseau social ne permet pas de trier les différents abonnements par catégorie. On ne dispose donc que d'une seule forme de tri qui est la chronologie des publications. Or ce n'est pas forcément efficace lorsque l'on cherche à se documenter sur un sujet en particulier.

Afin d'améliorer ma veille sur Twitter, j'ai choisi d'utiliser l'outil Tweetdeck pour de créer un tableau de bord. Ce tableau est découpé en catégories dans lesquelles j'ai listé les sources qui m'intéressaient. Cela permet de créer en quelque sorte plusieurs fils d'actualités Twitter triés par



thème.



Tableau de bord Tweetdeck

2. Choix des sujets à suivre

Le but étant de se tenir à jour des différents sujets concernant l'écosystème python, mais également du développement en général. En tant que développeur python, il faut bien évidemment s'intéresser à toutes les nouveautés apporté à ce langage et à ses principaux frameworks, mais il est aujourd'hui difficile de faire l'impasse sur les technologies front-end (HTML, CSS ou encore Javascript). J'ai donc également intégrés ces sujets là dans le choix de mes sources d'information.

Enfin, il ne faut pas, à mon sens, se limiter à une veille concernant uniquement les technologies que l'on utilise au quotidien. Il est également important de s'informer grâce à des sources généralistes, de tout ce qui se passe dans le mode du développement au sens large, et même dans le monde du numérique en général.

Voici un tableau détaillant la logique des rubriques créées dans mon tableau Tweetdeck :

Thèmes	Développement	Python	Développement Front-End	Actualités numériques
Détails	Permet de suivre les actualités du monde du développement et de la programmation en général.	Permet de suivre les actualités liées au langage python et à ces principaux frameworks tels que Flask ou	Permet de suivre l'actualité des différentes technologies liées au développement Front-End (HTML,	Permet de suivre l'actualité numérique au sens large (études économiques liées au numérique,

		Django	CSS, Javascript)	avancée technologique, info hardware...)
Exemple de sources	Developepez.com, StackOverflow...	Compte officiel python, compte officiel Django, Real Python, Dan Bader (influenceur)...	Javascript, HTML, Eric Meyer (Influenceur)	Blog du modérateur, Numérama, Zdnet, les Echos Tech...

3. Choix des sources de documentation

Une fois les sujets sur lesquels rester en veille identifiés, la dernière étape consiste à choisir les sources à suivre pour rester en veille sur ces sujets. Pour ma part j'ai choisi trois catégories de sources différentes qui me paraissaient adaptées.

3.1 Sources « Officielles »

La meilleure source d'information pour connaître les principales évolutions d'un langage ou d'un framework reste encore les diffusions officielles réalisées par les créateurs de ces langages/framework. L'avantage de telles sources est la fiabilité de l'information car celle-ci n'est pas déformée par les différents intermédiaires qui pourraient la relayer.

Il existe cependant des inconvénients à ce genre de source. Tout d'abord, le côté techniques des informations peut parfois rebuter. Une version résumée et vulgarisée de l'information est parfois intéressante pour attirer l'attention et donner envie de se plonger plus profondément sur l'étude d'un sujet.

3.2 Source spécialisées

Par sources spécialisées, j'entends ici les médias (presse, blogs...) spécialisés dans le domaine du développement ou plus généralement du numérique. Elles ont l'avantage de donner des informations souvent plus accessible en première lecture (en opposition avec les sources officielles présentées précédemment), ce qui permet d'avoir rapidement une vue globale d'un sujet et de savoir si l'on souhaite approfondir ou non notre connaissance sur ce sujet. Un autre avantage de ce genre de source est que le point de vue adopté est parfois plus neutre que celui utilisé par une source « officielle ». En effet, cette dernière pourrait, par exemple, adopter un ton commercial dans la présentation d'un nouvel outil.

3.3 Influenceurs

Les influenceurs sont une source différente d'informations mais qui constituent aujourd'hui un très bon complément aux sources détaillées ci-dessus. En effet, ces personnes font eux même une veille technologique et vont partager à leurs *followers* des ressources qui leur paraissent intéressante dans leur domaine d'activité.

3.4 Récapitulatif et exemples

Voici quelques exemples de sources que j'ai utilisées répartie selon les différentes catégories présentées ci-dessus.

Source	Officielle	Spécialisée	Influenceurs
Exemple	Python, Django, GitHub...	Developpez.com, Blog du modérateur, Sam Et max (blog)...	Dan Bader, Eric Meyer...

4. Cas concret d'utilisation de l'outil : Le changement des conditions d'utilisation de l'API Google Maps.

Courant du mois de Mai 2018, Google a annoncé que le recours à l'API de Google Maps allait devenir payant à partir du mois de Juin. Jusqu'alors, les développeurs pouvaient utiliser cette API pour réaliser jusqu'à 25 000 « appels » par jour au service et ce gratuitement. Cette nouvelle a été fortement relayée par la presse spécialisée (Zdnet, L'informaticien, developpez.com, clubic...). Ayant choisi de suivre les informations relative au monde du développement dans son ensemble sur mon outil de veille technologique, c'est tout naturellement que j'ai eu connaissance de cette nouvelle. Cette information m'a directement interpellé car dans le cadre du parcours développeur d'application python, nous utilisons cette API pour la réalisation du projet n°7.

Sans veille technologique, cette information peut passer inaperçu pour un développeur (bien que celle-ci ait été particulièrement relayée et qu'il soit difficile de passer à côté). Pourtant les conséquences découlant de l'ignorance de ces modifications d'un service peuvent être importantes. En effet, le service devenant par défaut payant, il est désormais nécessaire de remplir quelques formalités afin de pouvoir continuer à utiliser l'utiliser dans ses applications (ex : mise à jour ses informations de facturation), faute de quoi l'application pourrait ne plus être fonctionnelle car l'appel au service serait refusé.

Dans mon cas, j'ai utilisé l'API Google Maps pour un projet d'étude avec un but pédagogique. Les conséquences d'une application qui deviendrait non fonctionnelle aujourd'hui sont donc peu importantes. Mais on peut imaginer les conséquences que cela pourrait avoir pour un développeur qui a mis en place une telle application pour un client et que celle-ci devenait « hors service » du jour au lendemain.

Finalement, la date de mise en place des changements de conditions d'utilisation a été repoussée au 16 juillet, et Google va offrir 200\$ d'utilisation mensuelle gratuite afin que les petits projets puissent continuer d'utiliser ce service. Encore une fois, c'est grâce à la veille technologique mise en place que j'ai pu obtenir ces nouvelles informations.