

Programmation web avec Ur\web

Xavier Van de Woestyne

Octobre 2015

Même si je pratique de moins en moins la programmation web pour autre chose que des expérimentations, je suis toujours ouvert aux nouvelles technologies (même si j'ai beaucoup de mal à suivre les évolutions liées au langage JavaScript). Dans cet article, je vous propose de sommairement découvrir le [langage Ur](#) au travers de son cadriciel: **Ur/Web**.

Dans cet article, nous observons les motivations d'un tel cadriciel, et nous survolerons un exemple de projet en Ur/Web.

Redécouverte du web avec Ocsigen

Après avoir programmé pendant 10 ans (de mon plein gré) en PHP, en 2009, j'ai décidé de me tourner vers d'autres types de langages et de problématiques. J'avais laissé le Web de côté et c'est récemment, que grâce au cadriciel [Ocsigen](#) (pour le langage OCaml), j'ai décidé de me réintéresser à ce que se faisait dans les outils de développement web.

Ocsigen m'aura fait comprendre les architectures orientées **services**, le fondement d'une application **client-serveur** et les bénéfices de la programmation web statiquement typée.

Attention, la partie qui suit n'est pas une critique péremptoire, Ocsigen reste un des outil que j'ai préféré utiliser jusqu'à présent pour faire du web et je reste un utilisateur de Js_of_ocaml !

Même si Ocsigen est un outil moderne et sécurisé, capable de réaliser très bien des tâches complexes (typage des données, inter-communication entre le client et le serveur, programmation fonctionnelle réactive), je l'ai trouvé assez dur à prendre en main. (En effet, le cadriciel ayant subi beaucoup de modifications durant son développement, ses exemples ne sont pas très nombreux).

De plus, la construction de requêtes SQL au moyen de [Macaque](#) était, de mon point de vue, un peu trop verbeux, difficilement lisible et n'offrait pas la possibilité d'écrire des jointures externes. Tout en restant un grand admirateur

du projet, je l'ai trouvé un peu trop rugueux pour écrire une application web complète et j'ai délaissé toute la suite pour me focaliser sur [Js_of_ocaml](#) (le compilateur OCaml en byte-code JavaScript) pour le client, et utiliser simplement [Erlang](#) et [Yaws](#) pour le serveur.

Même si je suis encore le développement de Ocsigen (et que je compte m'y replonger dans le futur), j'ai voulu chercher des alternatives convaincantes. Il y en a plusieurs mais pour ma part, je me suis focalisé sur Ur (pendant une petite poignée d'heure). Cet article n'est cependant pas "*Ur pour le développeur Ocsigen*", mais une petite introduction à un outil qui lui ressemble. Aucun pré-requis en Ocsigen n'est donc demandé (et je ne passerai pas mon temps à ajouter "comme Ocsgien" pour les traits lui ressemblant).