



Rapport de projet XML

Enseignant : Philippe Poulard

13 décembre 2016

Créé par : Aichatou Traoré, Julien Hubert, Hamza Chahi, Aimé Niska

Rapport de projet XML

Enseignant : Philippe Poulard

Aspects techniques :

Partie serveur :



Nous avons choisi pour implémenter notre serveur, de le faire avec **Node.js** et **ExpressJS**. **eXistdb** est lui utilisé comme SGBD. Il contient la base **XML** native qui est requêtée en **xQuery** par le serveur Node.js. **SPARQL** est utilisé par le Node.js vers **Wikidata** pour compléter les données.



Partie cliente :

Pour la partie cliente c'est une application Web utilisant le framework CSS **Bootstrap** et construite en **AngularJS** qui va requêter le serveur Node.js puis traiter et afficher les informations reçues.

Communication client-serveur :



L'application cliente communique avec le serveur grâce à un **WebService AngularJS/ExpressJS** qui va demander au serveur Node.js d'effectuer pour lui telle ou telle requêtes xQuery/SPARQL. Les réponses renvoyées dépendent de la **route** sur ExpressJS demandée par l'application cliente et du contenu de la requête **GET** émise par le client.

Pour la carte, nous avons choisi l'API JS de Google Map, et en ce qui concerne la génération de PDF, nous avons choisi d'utiliser le module jspdf que l'on a jugé le plus pratique à utiliser.

Pour la base de données, nous avons préféré eXistdb à Base-X car c'est ce que nous avons utilisé lors du TP d'XML.

Fonctionnalités principales :



Google Map :

Il s'agit d'afficher une carte mondiale grâce à l'API de Google Maps. En effet, sur cette carte après la recherche d'un monument, il est affiché par-dessus celle-ci un pointeur disposé en fonction des coordonnées GPS du monument récupérés grâce aux requêtes SPARQL vers WikiData.



Recherche :

Cette fonctionnalité permet à travers différentes étapes de filtration telles que le choix de la région ainsi que le choix du département, de trouver le monument souhaité et de l'afficher sur la map grâce à la fonctionnalité citée auparavant.



Informations :

Après avoir recherché un monument grâce à la fonctionnalité de recherche une fiche descriptive est affichée sur le côté de la carte avec des informations sur le monument telles que la ville où il se situe, la région, son époque ou son statut. Ces informations ont été récupérées grâce à des requêtes XQuery sur la base de données Mérimée et on peut aussi voir une photo du monument grâce aux requêtes SPARQL qui viennent compléter ces données.



Export PDF:

Il s'agit, ici, d'un bouton qu'on a rajouté afin de créer un pdf. Cette fonctionnalité n'a malheureusement pas pu être finalisée. Il a été prévu de rajouter des informations sur le monument sélectionné sur le pdf afin que l'utilisateur puisse décider ou pas d'enregistrer les informations du monument souhaité.



Statistiques :

Après la sélection d'un département on peut voir quel pourcentage de monuments détient-il par rapport à l'ensemble du territoire Français (informations obtenues en xQuery via le Webservice).



Responsive Web Design :

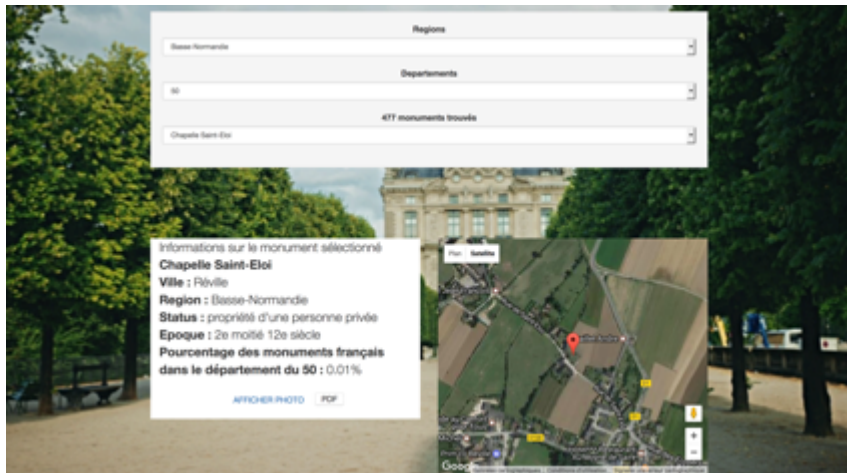
Grace à l'utilisation du framework Bootstrap l'application Web cliente est responsive (voir captures d'écrans page suivante).

Lancement du projet :

Pour lancer notre projet, il suffit de :

- Installer **eXistdb** et avoir comme collection la base de donnée Merimee-MH
- Lancer le serveur avec : `node srv.js` et installer si nécessaire les modules manquant (`npm install <nomModuleManquant> -- save`)
- Sur **Firefox**, taper dans la barre de navigation : `localhost:1337`

Quelques captures d'écrans



Responsive Web Design

