

TECHNOLOGIE XML

RAPPORT DE PROJET



MASTER 1 – MIAGE

Monsieur Philippe POULARD

Executive Summary

The project :

Realization of a RaWeb Inria data consultation web application "Activity report of research projects".
We are asked to make a web application to consult this data.

The application is composed of:

- A Java Web server with REST architecture
- An XML native database
- A Web client (CSS / JS, Framework of your choice)

In addition, the following data are available:

- RaWeb Inria "Activity Report of Research Projects" (total: 223 XML documents).
- A data feed from the list of Inria projects.
- A data feed from the list of Inria research centers.

Le projet :

Réalisation d'une application Web de consultation de données RaWeb Inria "Rapport d'activité des projets de recherche".

On nous demande de réaliser une application Web de consultation de ces données. L'application est composée de :

- Un serveur Web Java avec architecture REST
- Une base de données native XML
- Un client Web (CSS/JS, Framework au choix)

De plus, on dispose des données suivantes :

- RaWeb Inria "Rapport d'activité des projets de recherche" (total : 223 documents XML).
- Un flux de données de la liste des projets Inria.
- Un flux de données de la liste des centres de recherche Inria.

Partie 1. Travail réalisé

1. Partie client

Une fois le serveur Java lancé, le client peut accéder aux différentes informations concernant les rapports d'activités, les projets et les centres de recherches. En ce qui concerne les pages existantes, nous avons développé :

- Une carte listant tous les centres de recherches grâce à leurs coordonnées.
- La liste de tous les projets en fonction de leurs domaines.
- Des graphiques au format SVG représentant diverse information concernant les projets et les centres de recherches
- Les pages des centres de recherches

Chaque requête du client est directement transmise au serveur qui lui répond sous format JSON. De son côté, le client décode cette réponse et affiche les données nécessaires à l'utilisateur.

2. Partie serveur

Concernant la partie serveur, il gère les transactions entre le client et la base donnée. Le schéma suivant permet de mieux représenter cette architecture :



Le serveur peut interagir avec la base donnée native XML grâce à différentes requêtes XQuery. La réponse de la base se fait sous forme XML. Par la suite, le serveur encode ces données avec le format JSON et les envoie au client.

3. Partie base de données

La base de données native XML fonctionne avec « *ExistDB* ». Chaque table correspond à un rapport d'activité des projets de recherche (le total représente plus de 223 documents XML).

L'accès à ces tables se fait grâce à des requêtes XQuery envoyer directement par le serveur. La base lui répond en lui envoyant sa réponse sous forme XML.

Partie 2. Aspects techniques

Afin de réaliser ce projet, nous avons utilisé différent langages et outils.

Concernant les langages :

- Pour la partie client : HTML / CSS / Javascript
- Pour la partie serveur : Java / XQuery
- Pour la base de données : XQuery

Nous avons également fonctionné avec différents outils :

- Pour les IDE : NetBeans, IntelliJ
- Pour les navigateurs : Chrome, Firefox, Safari, Chromium
- Pour le serveur : Jetty Server
- Pour la base de données : ExistDB
- Pour la gestion de version : Bitbucket puis GitHub, Sourcetree

Afin de pouvoir utiliser l'application correctement, il faut dans un premier temps lancer la base de données « *ExistDB* ». Puis, dans un second temps, il suffit de lancer le serveur avec « *Jetty Server* ». Une fois ces deux étapes réalisées, le client peut être lancé correctement.