

GIPSI-GSI : Projet DWJ (XML, Java, Webservice)

Projet de DWJ

environ 10 heures/personne

Objet

Vous devez réaliser une application permettant de gérer les paris sportifs de l'euro 2016. Cette application a pour objectif de :

- Récupérer les pronostiques des joueurs.
- Récupérer les résultats réels des matchs.
- La mise à jour des points des joueurs en fonction de la comparaison entre le pronostique et le résultat réel.
- La comparaison entre joueurs.

Cette application sera utilisée par n personnes, allant de 2 à n, qui feront des pronostiques et compareront leur score de pronostiques.

Liste des fonctionnalités

Un pronostique consiste à donner le nombre de chaque équipe au cours d'un match. De plus, ce pronostique est réalisé à une date antérieure au match. Pour ce dernier point, le plus simple est de réaliser deux sites (1 pour parier, 1 pour voir le résultat de ses pronostiques) et de couper le premier avant le début du premier match.

Dans le cadre de ce projet, nous nous limiterons aux pronostiques des matchs de poules

Ainsi un joueur pourra :

- S'identifier sur le site de pronostique
- S'identifier sur le site de visualisation de ses pronostiques
- Parier sur un match de poule

L'administrateur pourra :

- Créer un joueur
- Importer les résultats des matchs

Le système devra :

- Calculer le nombre de points de chaque joueur en fonction du résultat de ses pronostiques.

Méthode de calcul des points

Suite à la saisie d'un résultat réel d'un match, chaque joueur se verra attribuer un nombre de point en fonction de la qualité de son pronostique selon la répartition suivante :

- Mauvais résultat (i.e. pronostique victoire equipe 1 alors que c'est une égalité) → 0 points
- Bon résultat mais mauvais score (bien victoire equipe 1 mais le nombre de but est faux) → 2 points
- Bon résultat et bon score → 5 points

Architecture

Afin de réaliser ce projet, l'architecture représentée par la figure 1 est proposée. Cette application est composée de deux serveur Web, de deux Web Services et d'une gestion de fichier XML.

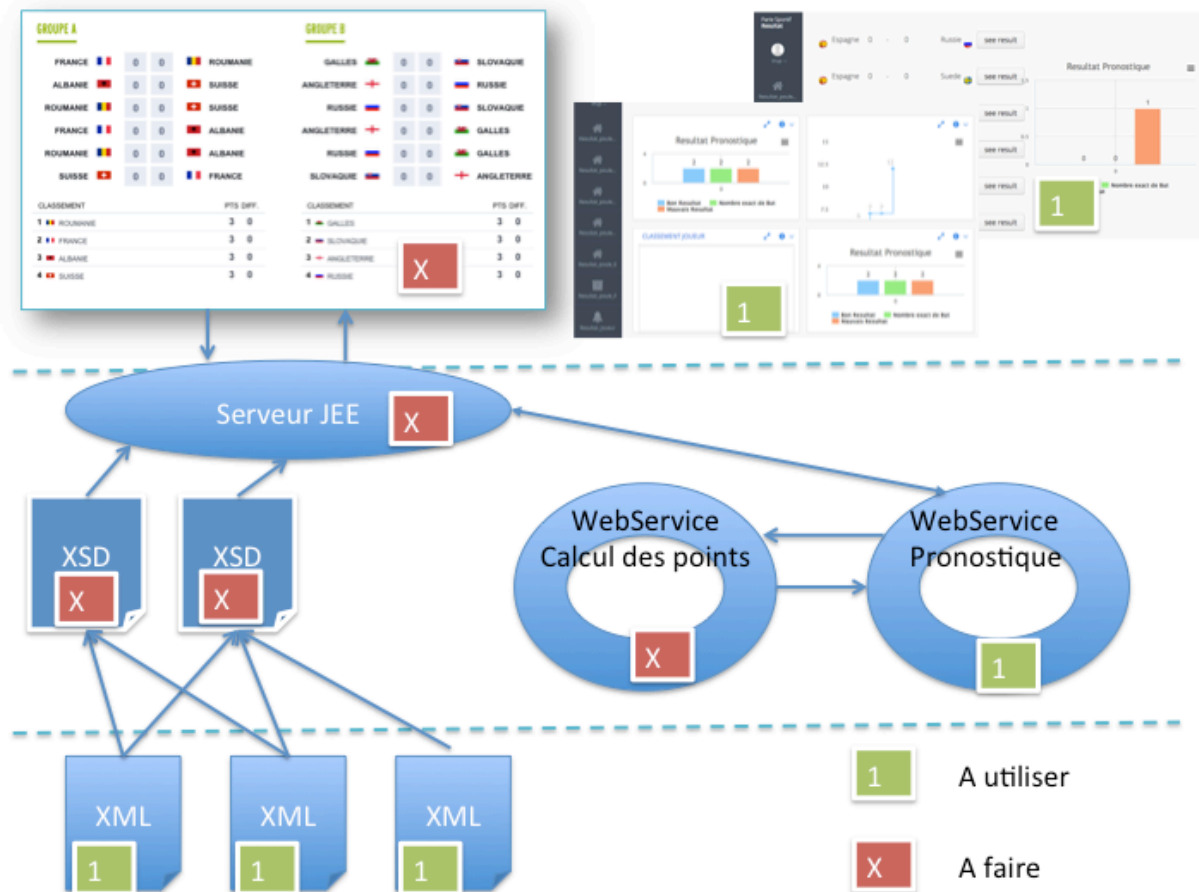


Figure 1: architecture de l'application

Serveur Web

Il y a deux serveurs web :

- 1) un serveur web qui suit le résultat des pronostiques des joueur. Cette partie vous est fournit elle correspond au projet ResultatPronostique. Pour lancer ce projet vous devrez utiliser la commande suivante à partir d'un terminal positionné sur le dossier du projet :

```
mvn vaadin:update-widgetset install jetty:run
```

Le site sera ensuite disponible à l'adresse : <http://localhost:8080>

- 2) Un site de pronostique que vous devez réaliser en JEE. Ce site devra permettre au joueur de faire ses pronostiques.

Web Services

Il y a deux webService :

- 1) Le premier Pronostique vous est fourni et correspond au projet WSPronostique. Pour le lancer il faut faire un run as java application sur la class [PronostiqueInterface_PronostiqueSOAP_Server.java](#)
Vous devrez générer les class permettant d'appeler ce webService en Java grâce au fichier creationClientWS.sh qui recense les commandes à executer.
- 2) Vous devez créer un webService qui calcul les points d'un joueur lorsqu'un résultat réel arrive.

XML

Afin de récupérer les différents résultats de match tout au long du championnat, des flux XML sont utilisés. Ceux-ci sont produits par deux sites sportifs différents. Ainsi, sur 6 poules, les matchs des trois premiers seront traités indépendamment des trois autres.

En conséquence, vous avez en entrée deux fichiers de matchs dont la structure est différente Match1.xml et Match2.xml.

Ces deux fichiers consistent chacun en deux parties : la première partie du fichier correspond aux résultats en eux-mêmes (score obtenus en fonction du numéro des équipes), tandis que la deuxième partie correspond à la liste des équipes (correspondance des numéros d'équipes avec leur nom ainsi que leur image drapeau).

1. Dans un premier temps, il vous est demandé de définir les Schema XML (XSD) correspondants à chacun de ces fichiers XML. Pour Matchs1.xml (resp. Matchs2.xml) vous réalisez le fichier Matchs1.xsd (resp. Matchs2.xsd). Pour vous aider, nous vous donnons un fichier XSD (Equipes.xsd) pour la deuxième partie *équipes* de chacun des XML. Vous pouvez donc vous focaliser sur la partie *résultats de match*.

Aide : Le nœud `<listeEquipes>`, directement structuré dans le fichier Equipes.xsd peut être directement inclus dans vos fichiers XSD en utilisant un *include*. Cela vous permet de pouvoir utiliser selon votre besoin l'élément *listeEquipes*, d'une manière plus propre (meilleure visibilité). Pour cela, vous pouvez ajouter la ligne suivante dans vos fichiers XSD, directement après le nœud `<xs:schema ...>` défini en tout début :

```
<xs:include schemaLocation="Equipes.xsd"/>
```

2. Pour chacun des fichiers XSD, vous réaliserez un fichier XSL permettant de passer à une structure unique de fichier (Matchs_recap.xml). Pour cela le Schema XML Matchs_recap.xsd vous est fourni.

NB : il vous est donc demandé ici d'être capable de faire une correspondance entre les numéros d'équipes (donnés dans la partie *résultats de match* des fichiers XML en entrée), et les noms des équipes ainsi que leur drapeau (donnés dans la partie *équipes* des fichiers XML données en entrée).

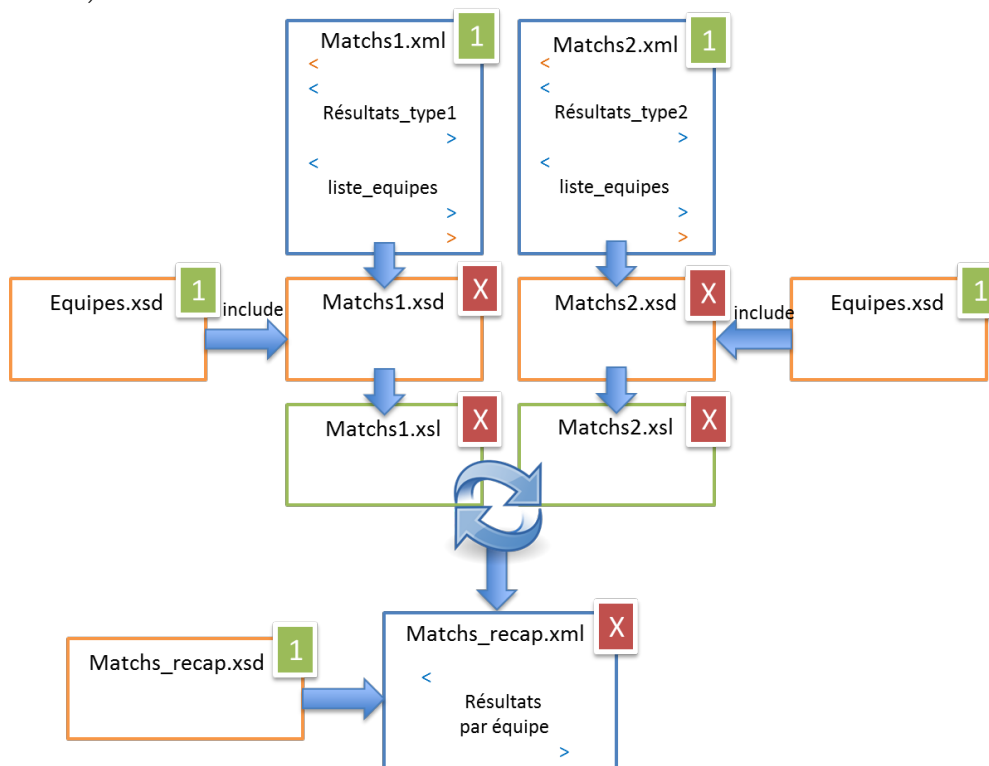


Figure 2: Récapitulatif des fichiers XML utilisés.

Travail demandé

Prérequis :

Vous devez télécharger et lancer mongodb : <https://www.mongodb.org/downloads#production>

Partie XML : 5 fichiers à rendre.

- Matches1.xsd
- Matches2.xsd
- Matches1.xsl
- Matches2.xsl
- Matches_recap.xml

Partie Java :

Un site de pronostique que vous devez réaliser en JEE. Ce site devra permettre au joueur de faire ses pronostiques

Partie Webservice

Vous devrez générer les class permettant d'appeler le webservice Pronostique en Java grace au fichier creationClientWS.sh qui recense les commandes à executer.

Vous devez créer un webservice qui calcul les points d'un joueur lorsqu'un résultat réel arrive.

Rendu :

Vous devrez rendre votre travail **sur le git**. Vous devrez créer un dossier : **GIPSI-DWJ-VotreLogin1-VotreLogin2**

Ce dossier sera constitué :

- un projet Java qui contient le code de l'application JEE pour le serveur web de pronostique
- un projet Java qui contient le code du Webservice Calcul des points utilisateurs
- un dossier contenant les 5 fichiers XML/XSD demandés
- vous pourrez rajouter des fichiers textes si vous souhaitez ajouter des précisions.