

# XMLIsland : rapport

David BORRY  
Justin GRIVON  
SI3 G4

May 27, 2016

Compte-rendu sur le développement de notre programme de conversion des rapports de championnats Island en format XML.

## Extraction des données JSON

Nous avons analysé la structure d'un championnat Island en JSON avant de commencer la conversion. Nous avons ainsi constaté que ces données sont représentées sous la forme d'un tableau d'éléments ayant des attributs commun (**time, meth et part**) mais un contenu différent. Le premier élément contient les données sur les contrats à accomplir. Les éléments qui suivent peuvent être représentés par des couples **action/réponse**. Le premier élément contient les informations sur l'action prise par l'explorateur et le second ses résultats. Tout le reste du tableau est constitué de cette manière. Un cas particulier : Le dernier élément peut être un message d'erreur (L'explorateur a franchi les limites de la carte ou a dépensé tout le budget par exemple).

Nous avons donc décidé d'extraire toutes ces données sous forme d'objets dérivant d'une classe Java que nous avons implémentée : **JSONData**. Cette classe a pour rôle d'extraire des données JSON et de les convertir en XML. En fonction du type d'objet, que ça soit une action de l'explorateur, une réponse, une erreur ou un contrat, on utilise toujours cette classe qu'on hérite pour traiter les attributs particuliers. Ces données sont alors toutes stockées dans la classe **EventList**.

## Conversion en XML et DTD

Une fois toutes les données extraites, on les convertit en XML. Pour cela, nous avons choisi une modélisation très proche de celle en JSON mais nous avons fait des choix sur des aspects propres à ce langage. Nous avons donc vite fixé une règle simple, toujours appliquée dans notre DTD : les objets à un seul attribut sont représentés par une balise avec pour contenu la valeur de l'attribut. Les objets aux attributs multiples en revanche sont représentés par une balise vide dans laquelle on stocke directement ces attributs. On peut alors les visualiser grâce aux attributs **::before** et **::after** du css.