MOVING WebServices Read-Only

*Spécifications*

Ce document synthétise les spécifications logicielles relatives aux WebServices en lecture qui devront être exposées par le BackOffice Moving-BO

Benjamin Verney – Thomas Badin

06/05/2014

MOVING WebServices Read-Only

Spécifications

Table des matières

[Introduction 2](#_Toc387226757)

[Contrôle d’accès 3](#_Toc387226758)

[Méthode authenticate 3](#_Toc387226759)

[Méthode getAccessToken 3](#_Toc387226760)

[Accès aux éléments 3](#_Toc387226761)

[Méthode getListAllParcours 3](#_Toc387226762)

[Méthode getParcoursArchitectureById 3](#_Toc387226763)

[Méthode getParcoursById 3](#_Toc387226764)

[Méthode getSousParcoursById 3](#_Toc387226765)

[Méthode getTransitionById 3](#_Toc387226766)

[Méthode getSceneById 3](#_Toc387226767)

[Méthode getListElementsBySceneId 3](#_Toc387226768)

[Méthode getElementById 3](#_Toc387226769)

[Méthode getListAllElements 3](#_Toc387226770)

[Méthode getPointdInteretByTag 3](#_Toc387226771)

[Description des services (WSDL) 3](#_Toc387226772)

# Introduction

Les web services permettent d’exposer des objets métiers vers l’extérieur. Les principaux objets métiers sont les parcours et sous parcours, les scènes, et les éléments (médias ou artefact) liés aux scènes.

Les WebServices seront utilisé à la fois par le Front Office Web et les applications mobiles :



Ce document ne traite que des WebServices permettant la consultation. Si des WebServices de modification sont nécessaire (ce qui a été évoqué lors de la réunion du 28 Avril 2014 : [Wiki](https://tuleap.cervin.org/wiki/index.php?pagename=Compte%20rendu%2028%20avril%202014&group_id=105) ), ils feront l’objet d’un autre document et la partie sera traitée dans un second temps.

Les WebServices seront en [SOAP](http://fr.wikipedia.org/wiki/SOAP), protocole le plus répandu actuellement. Le Framework PHP utilisé pour le back office ([Zend Framework 2](http://framework.zend.com)), permet directement d’exposer des services (cf [Zend/Soap/Server](http://framework.zend.com/manual/2.0/en/modules/zend.soap.server.html) ). C’est donc ce que nous utiliserons pour créer les WebServices.

Dans l’ensemble de ce document, nous allons spécifier la réponse du WebService par un exemple. Ce sera moins formel qu’une définition des réponses au format XSD, mais plus simple à comprendre.

# Contrôle d’accès

A faire (Ben).

## Méthode authenticate

## Méthode getAccessToken

# Accès aux éléments

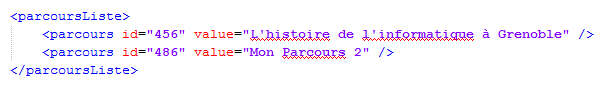
Cette partie regroupe les fonctions d’accès aux différents éléments de Movin.

## Méthode getListAllParcours

Méthode permettant de retourner tout les parcours « Public » du BackOffice. La méthode ne prend pas de paramètre.

### Cas standard

Il y a au moins un parcours « Public » dans le BO. Le XML retourné sera de la forme :



Note : La méthode ne prend pas d’argument, car un Back Office est associé à un seul Front. Dans le cas d’un fonctionnement multi-tenant (ce qui n’est pas prévus à l’heure actuelle), cette méthode devrait être supprimée, et remplacée par une méthode getListParcoursByClientId par exemple.

### Cas Limites

Il n’y a aucun parcours en visibilité « Public » dans le BO. La fonction renvoie une liste vide.

### Cas d’erreurs

Il n’y a pas de cas d’erreur.

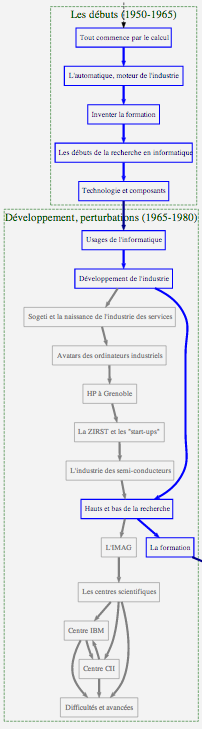
## Méthode getParcoursArchitectureById

Méthode permettant de récupérer l’architecture d’un parcours. L’architecture d’un parcours correspond à sa décomposition en sous-parcours et scène, et les relations entre les éléments. La méthode prend en paramètre l’identifiant du parcours.

### Cas Standard

Le parcours demandé existe dans la base de données, il est « Public » et possède au moins un sous parcours qui lui-même contient au moins une scène.

Prenons par exemple le parcours suivant :



Le XML reçu sera de la forme :



On ne renverra que les identifiants et noms, cette fonction permettra à afficher une vision globale du parcours, sans rentrer dans les détails.

### Cas Limites

Le parcours ne contient aucun sous parcours ou un sous parcours ne contient aucune scène :

* Cela ne devrait pas arriver, un parcours ne contenant pas de sous parcours ou un sous parcours ne contenant pas de scène est probablement du au fait que le parcours est en construction, il n’aurait donc pas du être passé en « Public »
* Il convient donc de logger le problème et de renvoyer un message d’erreur à l’utilisateur. (Parcours incomplet).

### Cas d’erreurs

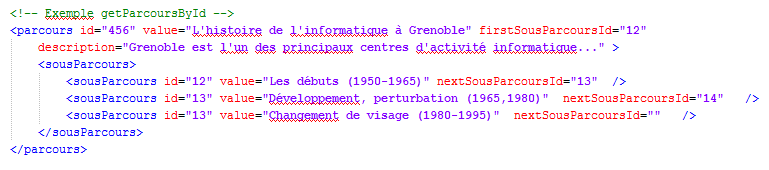
L’id spécifié n’existe pas, ou n’est pas en visibilité public : retourner un message d’erreur à l’utilisateur (Not Found).

## Méthode getParcoursById

Méthode permettant de récupérer les informations d’un parcours. On retournera toutes les informations du parcours, ainsi que la liste de ses sous-parcours. On ne détaillera pas les informations sur les sous-parcours, on se contentera de retourner l’id, le nom et les liens entre les sous-parcours.

### Cas standard

Le parcours existe, il est en visibilité « Public » et contient au moins un sous parcours. Dans ce cas le XML retourné sera de la forme :



### Cas Limites

Le parcours ne contient aucun sous parcours :

* Cela ne devrait pas arriver, un parcours ne contenant pas de sous parcours est probablement du au fait que le parcours est en construction, il n’aurait donc pas du être passé en « Public »
* Il convient donc de logger le problème et de renvoyer un message d’erreur à l’utilisateur. (Parcours incomplet).

### Cas d’erreurs

L’id spécifié n’existe pas, ou n’est pas en visibilité public : retourner un message d’erreur à l’utilisateur (Not Found).

## Méthode getSousParcoursById

Méthode permettant de récupérer les informations d’un sous-parcours. On retournera toutes les informations du sous-parcours, ainsi que la liste de ses scènes. On ne détaillera pas les informations sur les scènes, on se contentera de retourner l’id, le nom et les liens entre les scènes.

### Cas standard

Le sous-parcours existe, le parcours auquel il appartient est en visibilité « Public » et contient au moins une scène. Dans ce cas le XML retourné sera de la forme :



### Cas Limites

Le sous-parcours ne contient aucune scène:

* Cela ne devrait pas arriver un sous-parcours ne contenant pas de scène est probablement dû au fait que le parcours est en construction, il n’aurait donc pas du être passé en « Public »
* Il convient donc de logger le problème et de renvoyer un message d’erreur à l’utilisateur. (Parcours incomplet).

### Cas d’erreurs

L’id spécifié n’existe pas, l’id du sous parcours ne correspond pas à un sous parcours dont le parent (le parcours) est public : retourner un message d’erreur à l’utilisateur (Not Found).

## Méthode getTransitionById

## Méthode getSceneById



## Méthode getListElementsBySceneId

## Méthode getElementById

## Méthode getListAllElements

## Méthode getPointdInteretByTag

# Description des services (WSDL)