# TP 820 Architecture Orientés Services

Jordan MAURICE - Grégoire Badin



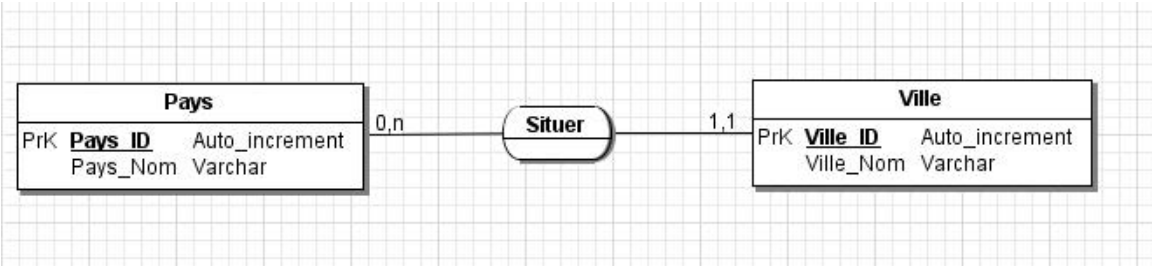
## Introduction

Le but de ce projet est de crée une application qui appel divers services web, deux services utilisant SOAP et deux utilisant REST, et en objectif secondaire de déployer « dans le Cloud » une partie de l'appli. Pour cela nous devrons développer une interface utilisant un service soap pour communiquer avec une base de données contenant un catalogue de voyages à faire et un service rest pour récupérer et enregistrer la liste des séjours retenus, les séjours choisies seront stockés « dans le Cloud ». Nous utiliserons aussi deux services existants, Flickr et google map, respectivement pour SOAP et REST, pour aller chercher des photos et une carte en rapport avec la ou les localisations recherché et apparaissant sur le catalogue.

# I. Général

## *A. Structure de la base de donnée*

Nous avons tout d'abord tenu à simplifier la base de données pour ne pas se compliquer le travail et pour démarrer facilement et rapidement, en effet l'intérêt du projet étant de travailler sur SOAP et REST il n'est donc pas nécessaire de construire des tables et une base de données très réaliste pour simuler nos besoins, c'est aussi pourquoi nous n'avons pas inclus de durée à nos séjours. Il y a donc deux tables et chaque ligne de la table ville correspond à un voyage du catalogue.




## *B. Format des messages qui transitent*

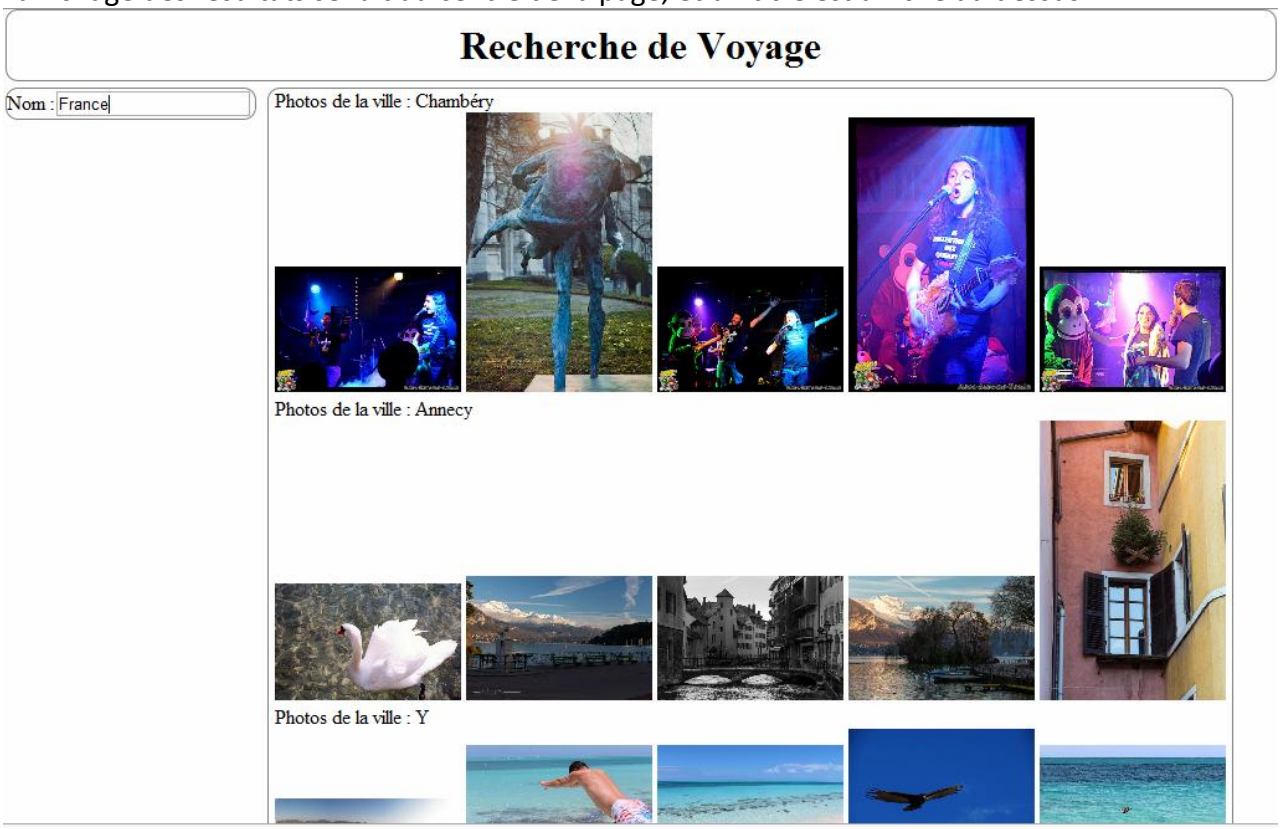
Les messages qui transitent sont des requêtes http contenant des données au format XML utilisant la norme SOAP ou REST suivant le cas du web service (cf schéma)

## C. *L'interface utilisateur*

Tout d'abord, nous avons décidé pour l'interface utilisateur d'utiliser la technologie JavaEE pour les raisons suivantes :

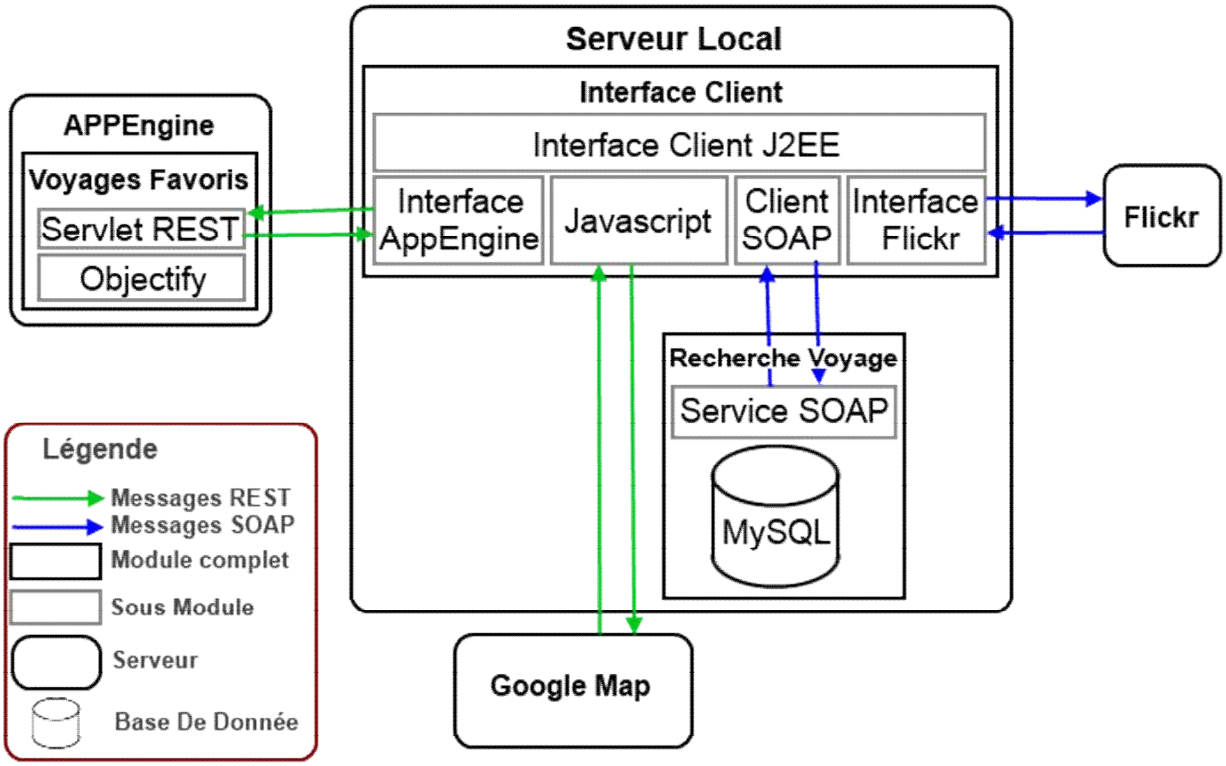
- Etant une technologie Web, cela nous permettait d'avoir un client léger, ne nécessitant pas d'installation, ce qui paraissait logique pour une solution logicielle de ce type.
- Une fois la mise en place finie, la mise en forme était plus simple à faire qu'avec une application Swing.
- Cela restais du java on pouvait donc réutiliser ce que nous avions vu en TD, ce qui étais le but de ces derniers.
- Notre expérience en swing était plus ancienne et peu poussée, Utiliser une technologie vue récemment paraissait donc une meilleure option pour démarrer rapidement.

L'Interface consiste donc en une simple page web avec un champ de recherche à gauche, l'affichage des résultats se fait au centre de la page, et un titre est affiché au dessus.




**Note** : on voit ici que les photos ne correspondent pas forcément à la ville cherchée

### *D. Architecture détaillée*




Voici le schéma de l'architecture de notre projet, les modules sont en réalité des projets Java à part entière qui sont indépendant entre eux, les modules sont la plupart du temps des classes Java. Dans les sources, le module Voyages Favoris correspond au dossier GoogleAppVoyage, Interface Client à InterfaceWeb et Recherche Voyage à SOAPVoyage. Le dossier Map contient les sources de ce qui devait devenir le sous module Javascript de l'interface.

# II. Les services

## *A. Base de donnée voyages*

### 1) Description

Nous avons crée un service SOAP permettant de communiquer entre la base de données et l'interface utilisateur. Pour nous aider dans sa mise en place nous avons repris exemple sur ce qui avait été déjà fait en cours.
Nous recevons dans ce service une chaîne de caractères contenant un nom de pays ou de ville via une requête post, protocole http, puis nous créons une simple requête SQL pour aller rechercher les informations voulues sous la forme d'une paire ville/pays. Enfin chez le client nous créons un tableau à partir des résultats obtenus.

### 2) Problèmes rencontrés

Des problèmes de connexion entre le service SOAP et la base de données MySQL ont été rencontrés, pour les résoudre on a cherché plusieurs solutions sur Internet. Après avoir appliqué plusieurs solutions différentes : changer l'appel du driver jdbc pour mysql car nous trouvé plusieurs façons différentes de l'écrire, changer l'écriture de l'adresse de la base de données plusieurs façon ici aussi, recréer un nouveau projet car notre projet étant sur un repository en ligne il y eu quelque petits soucis au niveau des fichiers lors des différents push et pull, etc… Finalement nous avons réussi à faire marcher la connexion mais nous ne savons pas si c'est seulement une des ces solutions où l'ensemble qui a résolu le problème.
Problème : SOAP ne renvoi pas d'interface.

## *B. Flickr*

### 1) Description

Le service Flickr permet de récupérer les photos que ce site propose, à travers entre autres, un web service SOAP. Pour ce service, nous avons simplement utilisé la documentation fournie par flickr et utilisé la méthode search décrite dans cette dernière.
Nous avons donc envoyé des informations au format XML avec les balises définies dans la documentation. La balise <tags> contenait le nom de la ville recherchée par le moteur de recherche de voyages.
Ensuite, nous avons récupéré la réponse que avons parsé avec JDOM pour reformer les URL des photos.

### 2) Problèmes rencontrés

Le premier problème rencontré ne fut pas réellement lié à l'utilisation de flickr, ni à SOAP, mais au parseur de la réponse XML : En effet, en java, le message de retour d'une requête SOAP se fait avec un objet SOAPMessage, qui possède une méthode SOAPBody() renvoyant un objet de type DOM (Document Object Modeler) qui est le parseur XML par défaut de java.
Cependant, à l'affichage, nous obtenions toujours :

```
Body null : null
```

Nous pensions au départ qu'il s'agissait d'une erreur dans la requête, ce qui la rendait très difficile à débugger, car nous n'avions pas d'erreur SOAP à proprement parler.
En réalité, il ne s'agissait uniquement du fait que le parseur DOM avait un fonctionnement totalement différent de ce que nous pensions et n'affichais pas du tout le contenu du Body.
Nous sommes donc passés par la bibliothèque JDOM qui est beaucoup plus intuitive et qui

nous permit de gagner du temps plutôt que d'essayer d'utiliser directement DOM. Nous avons utilisé des stream et la methode WriteTo de la classe SOAPMessage pour parser le document au lieu de la méthode SOAPBody.

Le second problème est plus une faute d'inattention qu'autre chose, était le fait que quel que soit les requêtes envoyées, les images renvoyées étaient les mêmes, sauf si on omettait le paramètre optionnel geoContext, auquel cas, rien n'étais renvoyé. Il s'est avéré qu'il s'agissait juste d'une erreur dans l'utilisation d'une balise XML qui s'appelais <tags> (pour chercher la ville) et que nous avions appelé <tag>. Nous ne nous attendions plus à ce qu'il renvoie une erreur si l'un des paramètres n'existait pas et cela nous a aussi pris du temps à comprendre.

Enfin, le dernier problème, étais simplement que même lorsque nous cherchions les bonnes choses, il renvoyait parfois des photos sans rapport (exemple : pour Chambéry l'équipe de handball) alors que sur le site avec la même ville nous obtenions de bon résultats. Nous avons tenté de rajouter des paramètres optionnels dans la requête SOAP pour corriger cela mais sans succès.

## *C. Google Map*

### 1) Description

Pour implémenter le système de carte développé par Google nous nous sommes aidés des différents tutoriels que l'on peut trouver sur leur site et de leur documentation très complète. Nous avons donc créé des fonctions qui peuvent être appelées par méthode HTTP GET et qui renvoient des chaînes de caractères, ces chaînes contiennent l'écriture de fonctions javascript qui une fois mise en ligne sur la page HTML initialise et instancie une carte centré sur la ville choisie et avec un petit marqueur sur la localisation.

### 2) Problèmes rencontrés

Nous n'avons pas eu assez de temps pour l'implémenter et surtout tester le code écrit. EN effet nous avons créé du code qui devrait fonctionner puisque le code est grande partie inchangée par rapport à l'exemple fourni par Google.

## *D. AppEngine*

### 1) Description

L'App Engine devait nous permettre d'enregistrer une liste de voyages sélectionnés dans l'interface principale du web service. L'interrogation de l'application sur le cloud Google devait se faire par l'utilisation d'un Web Service REST créé à partir d'un servlet Java. Les requêtes http POST et GET ont étés implémentés, mais malheureusement, l'application ne fonctionnait pas, pour une raison encore inconnue.
Le Web Service utilise Objectify pour enregistrer facilement des objets java dans une base de données (ici des simples id de voyages donc un tableau d'entier)

### 2) Problèmes rencontrés

Le principal problème rencontré fut un problème de conception : en effet, il fallait à la fois un service Rest et une application AppEngine. Nous avions vu en TD comment générer un Web Service Reste facilement et comment utiliser AppEngine mais pas fusionner les deux. Nous avons longtemps hésité à la façon de nous y prendre et au final, nous avons décidé de créer l'interface REST à la main.

Le second problème que nous n'avons pas résolut car nous n'avions pas eu le temps de chercher longtemps, est une erreur inconnue indiquée par Eclipse et qui empêchais aussi le fonctionnement sur le google App Engine. Eclipse indiquait :

| |
|---|
| "There was a problem generating the API metadata for your cloud endpoints classes : unknown protocol: d" |

Je n'ai pourtant détecté aucune différence majeure avec la classe du TD, leur fonctionnement étais quasiment la même avec un Servlet et une classe pour Objectify

## Conclusion

Pour conclure à propos du projet, Nous avons ateint plus de la moitié de l'objectif du projet, deux parties yant été complétées et deux autres parties étant en cours.
Nous aurions pu peut être avancer plus vite si nous n'avions pas perdu de temps sur certains point comme essayer de récupérer de bonnes photos sur flickr.
De plus, maintenant que nous sommes au courant de certains problème comme préferer l'utilisation de JDOM, et qu'une requête SOAP ne renvoie pas forcément d'erreur quand on envoie un mauvais argument, nous seront probablement plus attentif à ce genre d'erreur qui fait perdre du temps alors qu'elle sont simples à corriger.
Avec un peu plus de temps, il nous aurait été facile de Rajouter la possibilité de pouvoir chercher des voyages par durée en rajoutant un champ de durée dans la base de donnée et modifiant légèrement rapidement l’interface sans avoir a faire de modification sur le reste des modules.