UPMC
SORBONNE UNIVERSITÉS

Meet My Train

Rapport

Développement d'Applications Réticulaires

http://peaceful-sands-6919.herokuapp.com

Groupe :

Fatimata Coulibaly
Sarah Dahab
Thomas Aubry

Table des matières

I – Préambule

Ce projet que nous avons appelé « Meet my train » nous a été proposé dans le cadre de l'UE Développement d'Applications Réticulaires (DAR) par les professeurs Romain Demangeon et Vincent Simonet à l'Université Pierre et Marie Curie.

La SNCF a mis à disposition du public une interface de programmation (API) qui permet de connaître les prochains départs des différents trains pour chaque gare du réseau Transilien.

Le but de ce projet était de faire un site web à partir de cet outil tout en respectant certaines contraintes.

Notre application permettra non seulement de savoir l'heure des prochains trains à une station donnée, mais aussi les news géographiquement proche de la station, un plan des environs avec les différents types de commerce mais aussi et surtout la possibilité de rencontrer d'autres utilisateurs dans un train.

II – Technologies utilisés

1 – Les langages

Java : langage de programmation orienté objet. C'est ce langage que nous avons utilisé pour implémenté notre serveur notamment les servlet, les classes métiers et les JSP.

PHP: *Hypertext Preprocessor est* un langage de programmation libre permettant de produire des pages Web dynamiques côté serveur. Nous l'avons utilisé au début de notre projet pour les actualités.

JSP : technique basé sur Java pour développer des pages web qui respecte le standard XML. C'est une technique de développement web

HTML : (Hypertext Markup Language) et **CSS** (Cascading Style Sheets)

HTML est le format de données conçu pour représenter les pages web. Il est aujourd'hui utilisé en combinaison avec d'autres langages tels que Javascript et CSS (feuille de style) qui correspond au format de présentation d'une page html. Ces trois langages sont devenus complémentaires pour développer une application web dynamiques, interactives et esthétiques.

Javascript : Javascript est un langage de programmation de script principalement utilisé pour la réalisation de page web interactive. C'est un langage qui s'éxécute coté client et qui permet donc les interactions entre ce dernier et la page web/serveur.

2 – Les outils

JEE : Environnement de programmation d'application Eclipse. C'est une extension de la plateforme java standard JSE. C'est sous cet environnement que l'on a élaboré notre application Web.

TomCat : Est un conteneur web libre de servlets et JSP Java EE. Il est paramétrable par des fichiers XML et inclut des outils pour la configuration et la gestion. Et comporte également un serveur http.

Nous l'avons utilisé dans le cadre de ce projet afin de simuler un serveur web en local.

Flux RSS : *Really Simple Syndication,* Les flux RSS sont des fichiers XML qui sont souvent utilisés par les sites d'actualité et les blogs pour présenter les titres des dernières informations consultables, mise à jour en temps réel.

Framework : Un framework est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel. Un *framework* se distingue d'une simple bibliothèque principalement par :

Twitter Bootstrap est un framework développé pas Twitter et très utilisé sur une grande partie des sites et application web aujourd'hui. Il permet d'adapter sans difficulté l'affichage de l'application web à tous les écrans grâce à une librairie complète qui étend le langage CSS.

API - Interface de programmation : Une API est un ensemble normalisé de classes, de méthodes ou de fonctions qui permet à un logiciel souhaitant offrir des services à d'autres logiciels de le faire. Elle est souvent accompagnée d'une description détaillée pour faciliter son utilisation.Dans le cadre de ce projet nous avons utilisé les API GOOGLE Map et SNCF.

Base de donnnées :

-MySQL est un système de gestion de bases de données relationnelles (SGBDR). C'est un logiciel libre et il fait partie des logiciels de gestion de base de données les plus utilisés. C'est un système simple d'utilisation.

-JDBC est une librairie JAVA. Elle permet aux applications Java d'accéder par le biais d'une interface commune à des bases de données pour lesquelles il existe des pilotes JDBC. C'est cette librairie qui nous permet d'écrire et de lire dans notre base de données MySQL.

III – Le projet

1 – Les rencontres

Notre site offre la possibilité à un client de s'inscrire, en indiquant son identité et ses attentes amoureuses. Il peut avoir accès au profil des autres utilisateurs et son profil est accessible aux autres, et peuvent modifier leurs profils.

De plus les utilisateurs peuvent s'envoyer des messages.

Ils peuvent aussi faire savoir quel train ils vont prendre. Ils ont la possibilité pour un train donné de connaitre tous les utilisateurs qui ont indiqué qu'ils allaient prendre ce train. Ainsi leur permettre de faire connaissance par le biais d'un voyage dans un train SNCF.

2 – Les prochains départs

L'un des principaux objectifs de notre application est de fournir en un clin d'œil les prochains départs d'une gare SNCF. Pour cela il suffit de lancer une recherche à partir de la page d'accueil pour une gare donnée. La recherche nous redirige vers la page de la station. Une fois sur la page, la première information qui est affichée à l'utilisateur correspond aux dix prochains départs. Pour chaque départ il est possible de cliquer sur la gare de destination ce qui permet d'afficher la page de cette gare. Un utilisateur pourra voir pour chaque train les utilisateurs qui ont l'intention de prendre ce train et si il est connecté il pourra lui aussi signaler quel train il a l'intention de prendre.
C'est donc un moyen pour les utilisateurs qui sont à une même gare et qui attendent un train de se contacter via leurs profils. L'autre manière d'accéder à la page d'une gare est via la recherche d'une ligne qui dessert cette gare. Pour rechercher une ligne SNCF il suffit de saisir le nom de la ligne dans le champ de recherche qui se trouve sur la page principale de l'application. Ainsi sur la page d'une ligne on pourra trouver toutes les gares desservies par cette dernière.

3 – La minimap des alentours

Dans le cadre de rencontres il semble utile de proposer une recherche des services de consommation à proximité de la gare où l'on se trouve. L'application propose donc un plan GOOGLE avec la possibilité d'afficher l'ensemble des restaurants, hôtels, magasins à proximité de la gare. Ainsi

les utilisateurs pourront discuter dans un café ou restaurant proche du lieu où ils doivent prendre leurs trains.

4 – Les news

Offrir au client la possibilité de voir les actualités en temps réel, de la ville où se trouve la station qu'il a recherché. Ainsi apparait une liste de liens accompagnés d'un résumé d'article vers des sites proposants l'article concernant la ville et le département de la station.

IV – Déroulement

1 – Analyse / Conception

Le but du projet était de créer une application web qui fournirait un service innovant tout en faisant appel à l'API SNCF. En analysant ce qui existait déjà, notamment l'application RATP qui fournit un service assez complet pour tout ce qui est lié aux horaires des trains SNCF et en questionnant notre entourage, les pistes étaient limitées.

Notre première idée fut donc de donner un maximum d'informations sur une gare SNCF en plus des horaires, comme le plan des alentours ou les dernières actualités qui touchent ses environs. Pour chacune de ces fonctions il a fallu faire appel à des ressources externes. Pour le plan nous avons trouvé GOOGLE Map une API de GOOGLE.inc qui permet d'afficher aisément un endroit en fonction de ses coordonnées géographiques. Pour l'affichage de l'actualité d'une gare cela a été plus compliqué à mettre en ?uvre car il n'existait pas d'API fournissant ce service. Nous nous sommes donc tournés vers les flux RSS fournis par GOOGLE.

Il fallait rendre le site interactif d'où l'idée des comptes utilisateurs. Il a donc fallu créer une base de données d'où l'intégration de la base de données mySQL hébergée sur DB4FREE.NET. Ensuite nous avons eu l'idée de mettre en contact nos utilisateurs grâce à la possibilité d'échanger des messages via leurs comptes. Par ailleurs, pour pouvoir utiliser l'API SNCF il nous a fallu créer une table pour l'ensemble des gares SNCF. Toutes ces informations ont permis de définir l'ensemble des tables nécessaires pour le fonctionnement de notre application.

2 – Réalisation

La réalisation de l'application s'est faite en deux étapes.

Première étape :

Pour ce faire nous avons d'abord commencé par la création de la base de données en ligne, la mise en place du serveur web en ligne, l'affichage des actualités, les services concernant les trains et la minimap.

La base de données :

Nous avons logé notre base de données mySQL sur db4free.fr.

Tout d'abord nous avons créé une table station avec toutes les stations qui sont à la charge de la SNCF.

La table station contient un identifiant, l'identifiant SNCF, un nom et la ligne à laquelle elle appartient.

Serveur web :

Le serveur web a été logé sur Heroku.

Actualités :

Nous avons utilisé le flux rss de Google actualités pour les afficher. Ce flux nous permet d'avoir les actualités du département d'une ville donnée.

Pour ce faire, nous faisons une requête GET à l'adresse du flux, puis nous récupérons le format XML du flux et nous retournons uniquement les liens et les résumés d'articles. Pour ce faire

nous l'avons fait d'abord en PHP, puis en Ajax en utilisant une servlet RssServlet comme proxy, que l'on n'a pas pu aboutir jusqu'au bout en java.

Les trains :

Pour afficher les prochains trains d'une station ainsi que leurs horaires nous avons utilisé l'API SNCF qui nous a été fourni. Celle-ci fournis un identifiant et une mission. Pour faire appel à cette API avec Ajax nous avons utilisé comme proxy la servlet APITransilien qui se charge de faire l'appel avec l'identifiant du train et de renvoyer le résultat en JSON à la source.

Minimap :

Pour ce service nous avons utilisé l'API Google Map, qui est une sorte de bibliothèque et fourni des class permettant la manipulation de fonction en javascript afin de géo localiser les services aux alentours d'une gare.

Deuxième phase :

La deuxième phase consiste en l'ajout de comptes utilisateurs.

Profil :

Pour cela, nous avons créé une table Utilisateur avec pour clé l'identifiant («pseudo ») et comme autres champs : nom, prénom, âge, sexe, interesse_par, descriptif, mot de passe.

De plus nous avons mis en place un formulaire d'inscription et de connexion en JSP. Pour traiter les données soumis nous avons utilisé deux servlets, FormulaireServlet et ConnexionServlet gérant l'inscription et la connexion en déléguant à une class métier VerifInscription le traitement des données et à la class métier ConnexionTable la connexion à la base de données, sa mise à jour et l'ajout d'utilisateur.

Echange :

Pour les échanges entre les utilisateurs notamment les messageries, nous avons mis en place une table Messsagerie qui est défini par l'identifiant de l'émetteur, du récepteur, d'un champ texte pour le message et d'un champ date.

Un utilisateur peut envoyer un message à autre utilisateur en se connectant sur son profil. Pour ce faire nous avons utilisé la servlet ReadMessage a qui l'on transmet l'identifiant de l'utilisateur qui a les messages et il renvoie la liste des messages. Celui-ci se connecte à la base de données pour les mettres à jour. La vue des messages se fait avec JSP.

Prendre un train :

Les utilisateurs connectés peuvent enregistrer un train qu'ils vont prendre, et peuvent voir la listes des utilisateurs qui prennent un train données. Pour cela, de même on utilise une servlet ReadTrain qui lit dans la base de données et renvoie la réponse à travers d'une JSP. De plus on a créé une table prend_train qui est défini par le numéro du train, sa mission, l'id de la garde SNCF et l'identifiant de l'utilisateur.

V – Déploiement et intégration

1 – db4free

Nous avons choisis d'utiliser db4free.net pour d'héberger notre base de données. C'est un service gratuit et simple qui utilise MySQL. Il est facile avec la plupart des langages de programmation d'interagir via se gestionnaire de base de données. Remplir la base de données avec toutes les gares a constitué la première partie de la phase de réalisation. Nous ne savions pas encore que notre site serait sur un hébergeur proposant l'hébergement d'une base de données. Malgré cela, nous n'avons pas jugé utile de la déplacer car db4free nous convenait parfaitement.

Toutes les gares présentes dans le PDF donné par la SNCF (identifiants + nom) ainsi que la ou les lignes qui leurs correspondent y ont été copiés dans un premier temps. Par la suite nous y avons ajouté d'autres tables telles que celles qui représentent les utilisateurs, les messages échangés entre eux ainsi que les trains qu'ils prennent.

2 – Heroku

Nos servlets et notre site web sont hébergés par heroku.com. Nous avons fait ce choix car l'hébergeur, bien que difficile à prendre en main, proposait tous les services que nous recherchions. Par exemple il permet d'utiliser des servlets en java sans avoir à installer de serveur tomcat ou autre. Il est même possible d'y stocker une base de données. De plus il est possible d'avoir plusieurs sous-domaine sur ce sur ce site ce qui implique plusieurs applications. Cela aurait pu être utile.

3 – Les liens

Les liens vers les pages jsp :

http://peaceful-sands-6919.herokuapp.com/ C'est la racine du site web, qui correspond au fichier index.jsp
http://peaceful-sands-6919.herokuapp.com/Documents/Manuel_Utilisation.pdf Lien vers un PDF contenant un guide d'utilisation
http://peaceful-sands-6919.herokuapp.com/utilisateur.jsp Affiche la page d'un utilisateur
http://peaceful-sands-6919.herokuapp.com/message.jsp Affiche uniquement la liste des messages d'un utilisateur et que si il est connecté
http://peaceful-sands-6919.herokuapp.com/station.jsp Affiche les prochains horaires d'une station, les news et le plan des alentours.

Les servlets utilisés :

http://peaceful-sands-6919.herokuapp.com/APITransilien Servlet qui renvoi les prochains passages de trains à une station.
http://peaceful-sands-6919.herokuapp.com/Connexion Affiche un formulaire pour se connecter
http://peaceful-sands-6919.herokuapp.com/Formulaire Affiche un formulaire d'inscription
http://peaceful-sands-6919.herokuapp.com/ReadUser Servlet qui renvoi un utilisateur ou une liste suivant les paramètres.
http://peaceful-sands-6919.herokuapp.com/ReadMessages Envoi les messages d'un utilisateur via la méthode post
http://peaceful-sands-6919.herokuapp.com/ReadStation Différents type d'utilisation : Retourne l'identifiant d'une station depuis son nom / Retourne les identifiants des stations contenant une partie de nom / Retourne le nom d'une station depuis son identifiant
http://peaceful-sands-6919.herokuapp.com/Insert Servlet qui permet d'insérer dans la base de données
http://peaceful-sands-6919.herokuapp.com/Deconnexion Déconnecte un utilisateur connecté et redirige vers la page d'accueil
http://peaceful-sands-6919.herokuapp.com/DeletePassager Supprime un passager parmi la liste de ceux qui prennent un train
http://peaceful-sands-6919.herokuapp.com/ListeStations Liste les stations depuis une ligne

VI – Problèmes rencontrés

1 – L'installation et l'utilisation de Heroku

Le nombre de service offert par cet hébergeur est très grand. Cette offre implique donc une difficulté d'utilisation relativement grande.

Dans un premier temps nous avons dû

- Installer heroku et d'autres programmes utiles à l'exécution d'une application en local.
- Créer des variables d'environnement Windows afin d'exécuter les applications en ligne de commandes
- Ce familiariser avec l'utilisation de git via un hébergeur
- Adapter les fichiers de configurations de base d'une application de linux vers Windows.

2 – Les identifiants Sncf

Pour se connecter à l'API prochains départs de la Sncf, il nous faut des identifiants. L'université a fait une demande pour nous auprès de la Sncf pour avoir que chacun puisse s'y connecter. Or aujourd'hui (le 01/11/2014) nous ne les avons toujours pas reçu.

Cette phase nous a considérablement ralenti dans le sens, ou nous aurions aimé faire des tests sur l'API assez rapidement et voir toutes les fonctionnalités quelle nous proposait. Au bout de trois semaines nous avons pris les devants et avons fait une demande privée à la Sncf pour avoir les identifiants. Ils nous ont été communiqués dans la journée. Cela nous a permis d'avancer.

3 – Une API limitée

Pour notre application web nous avons été limités par le fait que les trains ne sont pas identifiés par des noms de lignes connues du grand public mais par des identifiants. N'ayant pas trouvé de solution pour faire correspondre les numéros de train à une ligne, il a été décidé qu'un train serait identifié dans notre application par son identifiant SNCF, sa mission et sa direction.

4 – Flux RSS

Au début du développement de l'application, nous avons utilisé PHP pour afficher les actualités. Puis on a décidé de changer et de le faire en Ajax car le fichier n'était pas compatible avec le serveur. Cependant étant données qu'on doit faire appel à une URL d'un autre domaine, cela à poser des problèmes de Cross Domain. Il a donc fallu créer un proxy qui est la servlet RssServlet afin d'y palier Mais par manque de temps on n'a pu aboutir.

VII – Conclusion

Pour conclure, ce projet nous a énormément apporté sur différents points de vue.

En effet c'est un projet de taille conséquente faisant appel à de nombreuses technologies. Il nous a permis de connaître toutes les composantes d'un site internet relativement complexes. Que ce soit du côté client ou du côté serveur.

Chercher nous-mêmes les fonctionnalités du site c'est révélé être une tâche amusante et inhabituelle. Ça nous a confrontés à des problèmes que nous nous étions par encore poser jusqu'alors dans notre cursus, car répondre à un cahier des charges et le créer sont deux choses radicalement différentes.

D'un point de vue plus général, l'UE DAR a été très formatrice pour nous, elle nous a appris des technologies dont nous ignorions l'existence. Avoir pu les mettre en œuvre dans ce projet et pour nous une très bonne façon de clôturer cette première partie de semestre.

VIII – Meet my train dans le futur

- Un tchat pourrait être créé afin de faciliter les paroles entre les différents utilisateurs.
- Une photo pourrait également être associée à un utilisateur pour améliorer les rencontres.
- Des favoris pour les utilisateurs concernant les news, ou des stations.
- Permettre aux utilisateurs de noter une gare, une ligne, et même un utilisateur