

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.....	3
II – Technologies utilisés.....	3
1 – Les langages	3
2 – Les outils	3
III – Le projet.....	5
1 – Les rencontres.....	5
2 – Les prochains départs.....	5
3 – La minimap des alentours.....	5
4 – Les news.....	6
IV – Déroulement	7
1 – Analyse / Conception.....	7
2 – Réalisation	7
Première étape :	Erreur ! Signet non défini.
Deuxième phase :	Erreur ! Signet non défini.
V – Déploiement et intégration	8
1 – db4free	8
2 – Heroku	8
3 – Les liens.....	8
VI – Problèmes rencontrés	10
1 – L’installation et l’utilisation de Heroku.....	10
2 – Les identifiants Sncf	10
3 – Une API limitée	10
4 – Flux RSS.....	Erreur ! Signet non défini.
VII – Conclusion	11
VIII – Meet my train dans le futur	11

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 proches 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.

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 respectent 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'exécute coté client et qui permet donc les interactions entre ce dernier et la page web/serveur.

2 – Les outils

J2EE : 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. Il 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'actualités 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 d'un logiciel.

Twitter Bootstrap est un framework développé par 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 données :

-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 ce qu'il recherche. Il peut avoir accès au profil des autres utilisateurs et son profil est accessible par les autres. De plus les utilisateurs peuvent s'envoyer des messages. Ils peuvent savoirs quels utilisateurs vont prendre un train grâce à un système permettant de sélectionner leurs trajets.

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 sur une gare. La recherche nous redirige vers la page de la station. Une fois dessus, 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 le prendre. 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 actualités

Cette fonctionnalité permet d'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 apparaît une liste de liens accompagnés d'un résumé d'article vers des sites proposant cette actualité 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 œuvre 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, nous avons donc pensé aux comptes utilisateurs. Il a donc fallu créer une base de données que nous utilisons via MySQL, hébergée sur db4free.

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 Snf. Toutes ces informations ont permis de définir l'ensemble des tables nécessaires pour le fonctionnement de notre application.

2 – Réalisation

La première étape de la phase de réalisation a été d'instancier la base de données. Il nous a paru nécessaire dès le début que peu importe les fonctionnalités de notre site après, avoir une base contenant le nom des stations reliés à leurs identifiants Snf était primordial.

Ensuite nous nous sommes répartis le travail afin d'effectuer parallèlement, une interface pour le site, le déployer sur internet et commencer à implémenter quelques fonctionnalités basiques.

Pour l'interface nous avons utilisé la bibliothèque Bootsrap afin que le site soit utilisable depuis n'importe quel type de navigateur récent.

Le déploiement sur Heroku a été réalisé grâce aux outils que cet hébergeur propose.

Le mini-plan et les actualités ont été implémentés à ce moment-là.

Nous nous sommes rendu compte que le travail que nous avions prévu n'était pas suffisant pour ce projet. Nous avons donc réfléchi à une fonctionnalité principale pour notre application. L'idée des rencontres nous est alors apparue.

Des comptes utilisateurs ont été créés. Cela a nécessité des nouvelles tables dans la base de données. Cette étape nous a pris beaucoup de temps car nous voulions une interface fluide et fonctionnelle, que ce soit pour un utilisateur connecté ou un simple visiteur.

Plusieurs servlets (détaillés par la suite) ont été utilisés pour interagir avec la base de données. Que ce soit pour insérer des données, ou lire celle déjà écrites.

V – Déploiement et intégration

1 – db4free

Nous avons choisi 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 ce 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 copiées 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-domaines 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
- Se 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 reçus.

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 qu’elle 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.

VII – 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

VIII – 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 s'est révélé être une tâche amusante et inhabituelle. Ça nous a confrontés à des problèmes que nous nous étions par encore posé 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.