Université de Technologie de Compiègne

Génie Informatique

Rapport de TD - Projet Emploi du temps

SR03

Steve LAGACHE & Romain PELLERIN

Chargé de TD : Cédric Martinet

Printemps 2016 (P16)

Dernière mise à jour : 30 mars 2016

Table des matières

1 Projet : emploi du temps

Le but initial du projet "Emploi du temps" était de recréer un emploi du temps visuel pour n'importe quel étudiant (ou personnel) de l'UTC, à la manière de celui disponible sur l'ENT [1]. Cet emploi du temps se devait d'être un minimum responsive (affichable correctement sur mobile). Par ailleurs, ce projet reposait sur l'utilisation d'une API fournie par l'UTC pour récupérer les emplois du temps au format JSON (voir explications plus bas).

De plus, d'autres objectifs supplémentaires nous ont été donnés :

— Pouvoir afficher simultanément plusieurs emplois du temps

— Vérifier la disponibilité de l'API de l'UTC

En plus de ces objectifs secondaires, nous avons rajouté quelques fonctionnalités, détaillées plus bas.

1.1 Choix préliminaires

Avant de rentrer dans le vif du sujet, il est important de préciser comment nous avons pensé notre projet. Nous avons décidé de créer notre API, qui viendra se positionner entre celle de l'UTC et notre *front-end*. À cela trois raisons :

— Éviter le problème des requêtes *cross-domain*, refusées par l'API de l'UTC.

— Pouvoir rajouter des services supplémentaires si besoin à notre API (codes HTTP personnalisés par exemple, selon l'erreur).

— Pour le plaisir de faire deux languages, PHP et JavaScript.

1.2 Choix technologiques

Nous avons décidé de faire notre *back-end* en PHP, avec un *front-end* en HTML, CSS et JavaScript. Au départ nous voulions utiliser des bibliothèques ou *frameworks* (pour le *back-end* ou le *front-end*), dans le but de nous faciliter certaines tâches puis finalement nous avons décidé de tout faire "à la main", autant pour le challenge que pour avoir plus de contrôle. De plus, cela nous a permis d'approfondir nos connaissances dans ces langages. Bien trop de personnes savent utiliser un *framework* et non le langage associé. Pour toutes ces raisons, nous avons voulu coder notre projet sans bibliothèque.

Concernant les versions, notre projet repose sur une version de PHP $>= 5.6$ (il a tout du moins été développé et testé sur PHP 5.6, mais il est probable qu'il fonctionne tout de même sur une version inférieure). Quant aux navigateurs web, le projet requiert des fonctions récentes de JavaScript. Le projet a été testé sur Firefox 45.0 et Google Chrome 49. Il est également probable que le projet fonctionne sur des versions antérieures.

1.3 Architecture du projet

Le code HTML est servi par un script PHP. Ce code HTML appelle ensuite une feuille de style CSS pour embellir l'affichage. Le code HTML appelle également un script JavaScript contenant

1. `https://webapplis.utc.fr/edt/index.html`

quelques fonctions. Il y a également deux *assets* graphiques (deux images). Tout est dans le même dossier racine du fait du nombre peu élevé de fichiers.

1.4 Fonctionnement

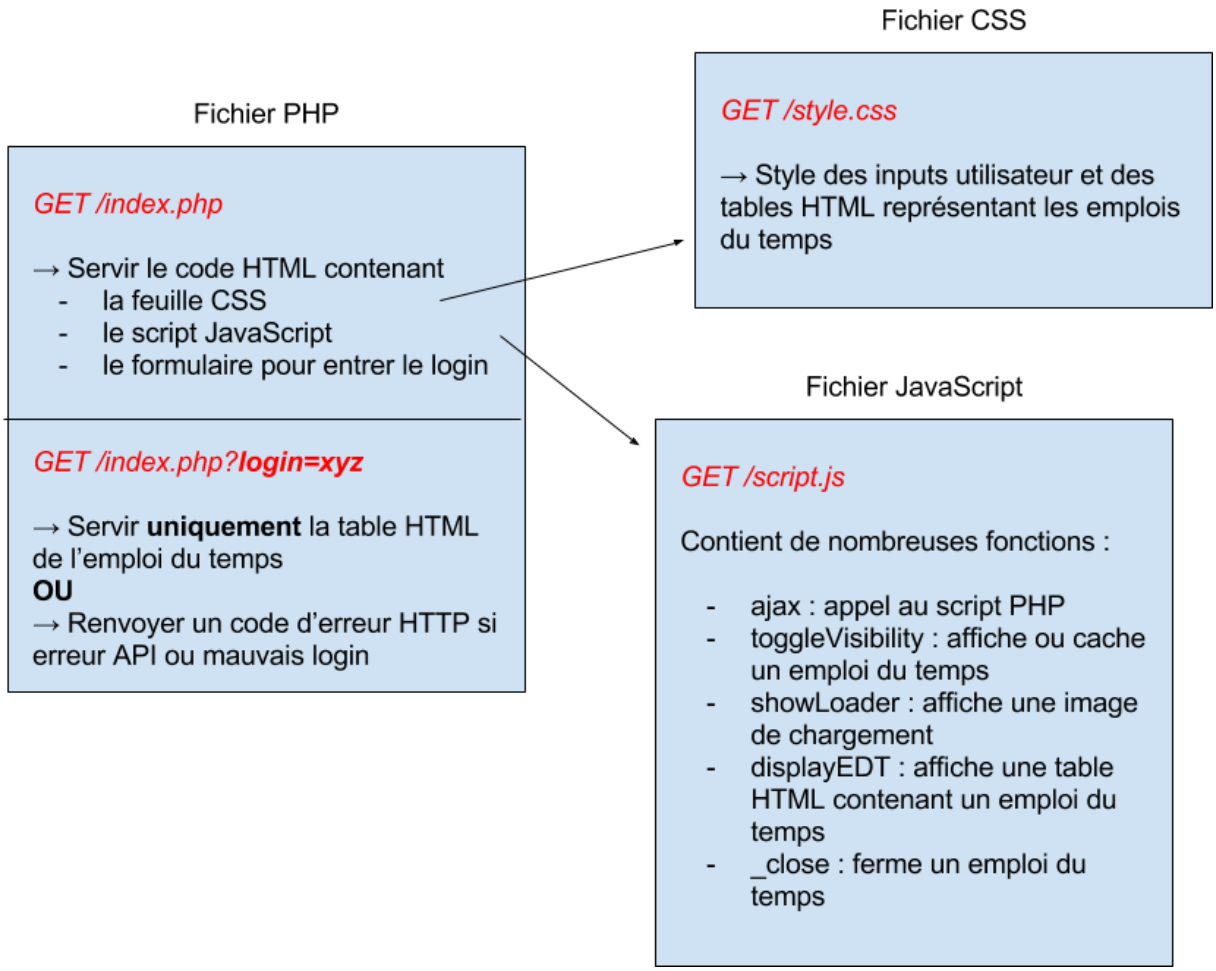

Figure 1.1 – Architecture du projet

Le JavaScript fera un appel à notre fichier PHP pour récupérer l'emploi du temps d'un étudiant, en transmettant dans la requête son *login*. La réponse contient directement du code HTML qui sera "inclu" dans notre page actuellement affichée grâce à JavaScript. Il y a quelques éléments visuels de *feedback* (mauvais *login*, API indisponible, etc).

Le code PHP, à la réception d'une requête HTTP contenant un paramètre `GET` nommé `login`, va aller faire une requête sur l'API de l'UTC [2] afin de récupérer l'emploi du temps correspondant au *login*. Cet emploi du temps, servi sous le format JSON, va être traité par notre code PHP afin d'obtenir en résultat final une `<table>` HTML. Le code PHP va envoyer ce code HTML final en réponse à la requête envoyée par le JavaScript.

Le JavaScript va inclure directement dans la page la table HTML reçue en réponse. Ainsi, il nous est très facile d'afficher plusieurs emplois du temps les uns à la suite des autres. Chaque requête avec un *login* nous permet de récuperer un table HTML qu'il nous suffit de rajouter à la suite, dans notre page couramment affichée.

2. `https://webapplis.utc.fr/Edt_ent_rest/myedt/result?login=<login>`

1.5 Résultat final

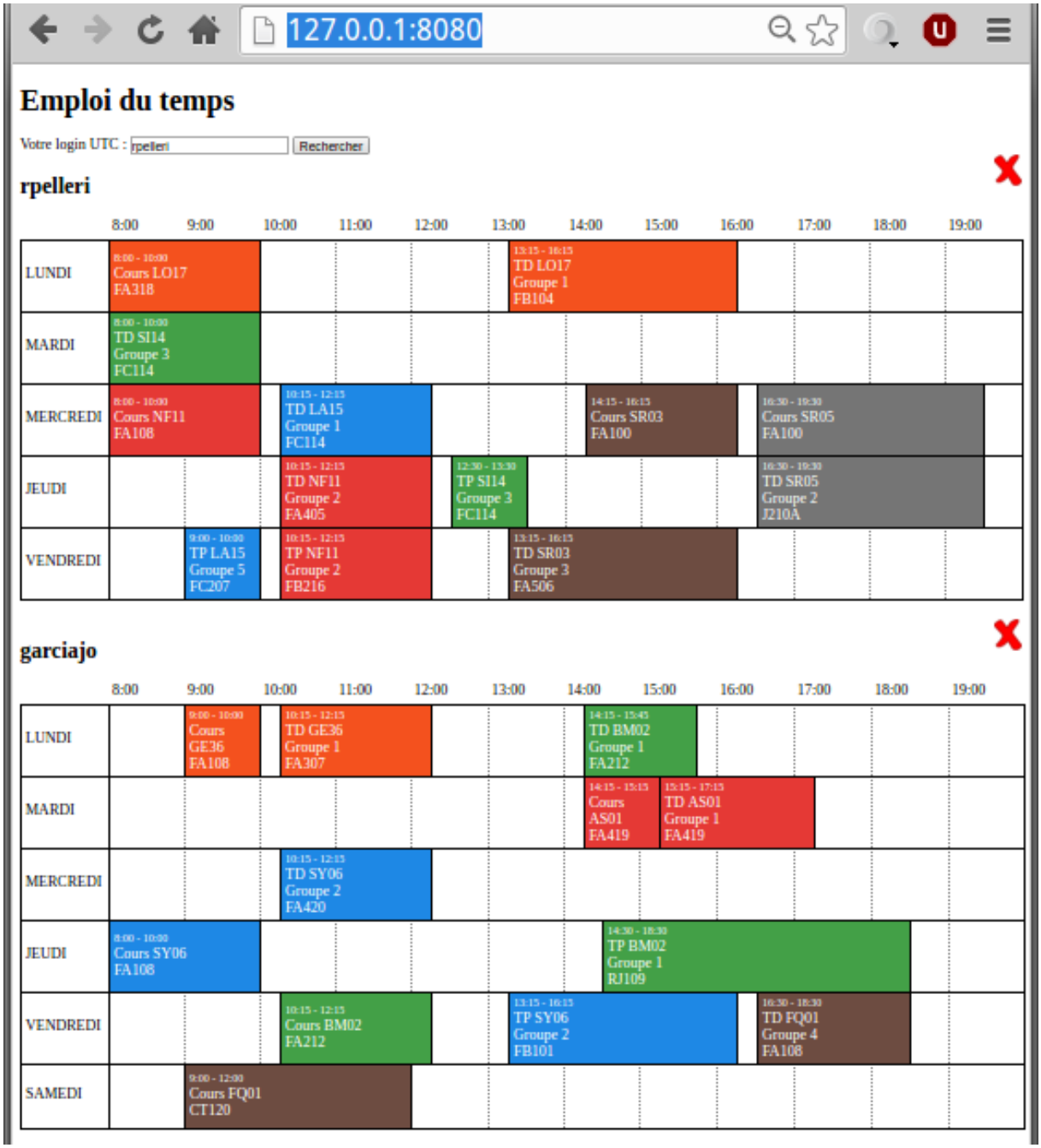

Figure 1.2 – Emplois du temps

1.6 Fonctionnalités ajoutées

- Message d'erreur affiché si l'API de l'UTC ne répond pas un code 200
- Message d'erreur si mauvais login entré (l'API de l'UTC rendra un tableau JSON vide)
- Possibilité d'afficher plusieurs emplois du temps
- Possibilité de cacher temporairement des emplois du temps en cliquant sur le login (accordéon)
- Possibilité de fermer un emploi du temps ouvert définitivement

— Vérification que l'utilisateur n'essaie pas d'afficher deux fois le même emploi du temps ou qu'il ne valide pas le formulaire sans avoir rempli le champ pour le *login*

— Affichage du samedi que s'il y a un créneau horaire occupé

— Affichage d'une image de chargement (*loader*) en attendant la réponse de notre script PHP suite à l'envoi d'un *login*

1.7 Difficultés rencontrées

La grosse et unique difficulté a été la création de la table HTML servant à afficher un emploi du temps. Pour construire une `<table></table>` il faut procéder ligne par ligne (`<tr>...</tr>`) d'où notre choix d'affichage des jours les uns au-dessus des autres. Un petit algorithme a pour cela été mis en place (un autre également pour trier par jour et heure les cours, à partir du JSON reçu par l'API de l'UTC).

1.8 Comment exécuter le projet

```
sudo apt-get install php5-cli # Ou installer php5-cli avec un autre
    package manager, en fonction de l'OS utilisé

cd code/ # Aller dans le dossier contenant le code source
php -S 127.0.0.1:80 -t .
```

Puis ouvrir un navigateur Web et aller à `http://127.0.0.1:8080`.