

SR03

Rapport du Devoir 4

GILLET Guillaume

HERKENS Antoine

20/06/2020

Sommaire

Introduction	3
Cahier des charges	3
Contenu du rapport.....	3
Développement de l'application	4
Exemples et scénarios d'utilisation	5
Utilisation de l'application.....	6
Difficultés rencontrées	6
Conclusion	7

Introduction

Cahier des charges

Le projet à pour finalité de développer une application web pouvant afficher un ou plusieurs emplois du temps d'étudiants de l'UTC. L'application doit récupérer les résultats d'une API et traiter ces résultats pour afficher un emploi du temps, cette application est réalisée en javascript avec la bibliothèque REACT. Il faudra mettre en place une interface qui va pouvoir ajouter, supprimer l'affichage de vos emplois du temps. L'affichage sera sur un même calendrier. Il faut que votre interface soit la plus ergonomique, c'est à dire un outil facile à utiliser. Conseil : Les Uvs devront avoir une couleur différente pour mieux les différencier.

Contenu du rapport

Dans un premier temps, nous allons aborder la mise en place du développement de l'application. Ensuite, nous verrons des exemples et scénarios d'utilisation de l'application. Dans un troisième temps, nous verrons les difficultés que nous avons rencontrées lors de ce devoir.

Développement de l'application

Le développement commence par une prise en main de React. L'un d'entre nous n'a jamais utilisé Javascript avant ce devoir et donc il faut prendre en main javascript et React. Cela nous a pris beaucoup de temps, ne serait-ce seulement pour configurer l'environnement de travail.

S'ensuit l'utilisation de javascript pour exploiter le JSON récupéré de l'API. Cette partie était sûrement la plus simple car il fallait utiliser des fonctions natives comme `Json.parse`. Récupérer les données du JSON était donc simple car un code d'exemple était fourni et il était facile de l'adapter comme nous le souhaitions. Concernant l'affichage graphique, nous n'avions pas de connaissances particulières en front end. Nous nous sommes orientés vers la librairie `Timetables-events`, conseillée par notre chargée de TD. La librairie nous permet donc de faire l'affichage d'un calendrier simplement à partir de données basiques comme l'horaire de début et de fin, le titre, des informations complémentaires. Cette librairie s'est donc montrée très utile, pratique et puisque bien documentée, facile à utiliser.

Une fois la librairie prise en main et les données récupérées via l'API, nous avons dû faire en sorte de formater correctement les données afin qu'elles puissent être au même format que la librairie `Timetables-event`. Nous avons donc tout d'abord réalisé une preuve de concept en affichant les informations d'une seule uv pour un seul jour. Cela a pris un peu de temps car le format du JSON de l'API n'est pas le même que celui de la librairie. Une fois que nous avons trouvé le format adapté à la librairie, nous avons fait une itération sur toutes les Uvs récupérées dans le JSON. Cette dernière étape n'a pas pris trop de temps et nous n'avons pas rencontré de difficultés particulières.

Les couleurs des UVs sont différentes d'une UV à l'autre et choisies aléatoirement par un générateur de couleurs ce qui évite d'avoir les mêmes couleurs pour chaque uvs. Chaque UV a donc une couleur différente, mais les cours (TP, TD et CM par exemple) d'une même UV sont bien de la même couleur.

Nous savons comment mettre en place un affichage avec plusieurs emplois du temps et plusieurs logins (le code est en commentaires). Cependant, par manque de temps, nous n'avons pas réussi à l'implémenter complètement.

Exemples et scénarios d'utilisation

Si l'identifiant saisi est invalide, alors un message d'erreur s'affiche.

Identifiant UTC :

Veuillez renseigner un identifiant UTC valide

Voici l'emploi du temps de la personne dont le login est « aherkens ». Chaque UV a bien une couleur différente mais les enseignements d'une même UV sont de la même couleur. Toutes les données récupérées dans le JSON via l'API sont utilisées pour afficher les informations nécessaires : UV, salle, groupe, horaire. Si l'utilisateur souhaite consulter un autre emploi du temps, il n'a qu'à saisir un autre identifiant valide et le calendrier s'actualisera.

Identifiant UTC : <input type="text" value="aherkens"/>		<input type="button" value="Afficher"/>			
TIME	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
7:00					
8:00					
9:00					[NF17] FA104 Cours Groupe 1 08:00 - 10:00
10:00		[GE37] RO128 TD Groupe 2 09:00 - 12:00			
11:00					[RO03] FA420 TD Groupe 6 10:15 - 12:15
12:00					
13:00			[PH01] FA108 Cours Groupe 1 13:00 - 14:00		
14:00		[NF17] FB100 TD Groupe 2 13:15 - 16:15			
15:00	[GE37] RB110 Cours Groupe 1 14:15 - 16:15		[SR03] FA104 Cours Groupe 1 14:15 - 16:15		
16:00				[SPJE] TD Groupe 1 14:15 - 18:15	
17:00	[RO03] RB110 Cours Groupe 1 16:45 - 18:45		[PH01] FA403 TD Groupe 2 16:30 - 18:30		
18:00		[SR03] FA506 TD Groupe 3 16:30 - 19:30			
19:00					
20:00					

Utilisation de l'application

Pour utiliser l'application, il suffit de télécharger le code source disponible à l'adresse suivante :

<https://gitlab.utc.fr/aherkens/sr03/-/tree/master/devoir4>

Ensuite, il suffit via un terminal, de se diriger dans devoir4/devoir4 puis d'exécuter la commande « npm start ». Cette commande lance le serveur NodeJs qui héberge l'application. Ensuite, il suffit d'utiliser un navigateur quelconque et de saisir cette adresse : <http://localhost:3000/>. Vous pouvez désormais utiliser l'application !

Difficultés rencontrées

Nous avons rencontré des difficultés assez importantes dans ce projet.

Tout d'abord, la mise en place technique de l'environnement de travail n'a certes pas été très complexe mais un peu laborieuse. Nous avons suivi les conseils du cours mais il a eu quelques problèmes (l'IDE a déplacé d'elle-même certains dossiers comme les modules nodes). Cela a donc pris du temps, mais grâce à l'aide d'une tierce personne, nous sommes parvenus à configurer l'environnement de travail.

Ensuite, un membre n'a jamais fait de Javascript. Cela prend donc forcément plus de temps pour développer l'application. De plus, les deux membres de notre binôme n'ont jamais utilisé REACT auparavant. Malgré le cours que nous avons suivi, la pratique est toujours un peu plus longue à mettre en place.

Lors du développement, nous avons aussi été confronté à des problèmes de syntaxe du langage, effectivement comme nous ne le connaissons pas nous avons du faire beaucoup de recherches pour trouver comment certaines fonctions de React fonctionnent. Le fonctionnement de l'application étant très simple, c'est réellement la syntaxe qui a mis du temps à être trouvée et assimilée.

Les difficultés sont plutôt pour stocker les uvs de plusieurs personnes en même temps car nous ne sommes pas expérimentés sur le passage des variables au travers du javascript même si nous avons réussi à comprendre comment il fallait faire pour en afficher plusieurs à la fois.

Malgré toute ces difficultés, nous avons pris React en main, avons pu consolider nos connaissances par la pratique et avons réussi à implémenter la majorité des besoins du cahier des charges.

Conclusion

Pour conclure, malgré les difficultés et les pertes de temps rencontrées, ainsi que le manque de temps pour finir le projet en période d'examens de fin de semestre, nous sommes parvenus à un résultat satisfaisant qui nous a bien fait prendre en main les outils et comprendre le fonctionnement de React. Nous avons réussi à développer la majorité des besoins du cahier des charges, et pourrions avec davantage de temps finir à 100% ce devoir et l'optimiser.

Nous sommes globalement satisfaits de notre travail, mais nous sommes tout de même déçus de ne pas avoir pu l'aboutir entièrement par manque de temps et d'expérience. Avec plus de temps nous pourrions réaliser une meilleure application, en ajoutant du CSS, plus d'options, des options de personnalisation etc. Malgré tout, nous avons acquis des connaissances et de la pratique dans un nouveau langage, ce qui est le plus important pour nous dans le futur.