PlageConnect

Itération 2

Hugo LECLER

Code source et adresses de déploiement

Le code source du front-end :

https://github.com/Hlecler/PlageDashboardFront

Le code source du back-end:

https://github.com/Hlecler/PlageDashboard/

Front-end déployé à cette adresse :

http://plagedashboard.igpolytech.fr/

Back-end déployé à cette adresse :

http://plagedashboardapi.igpolytech.fr/

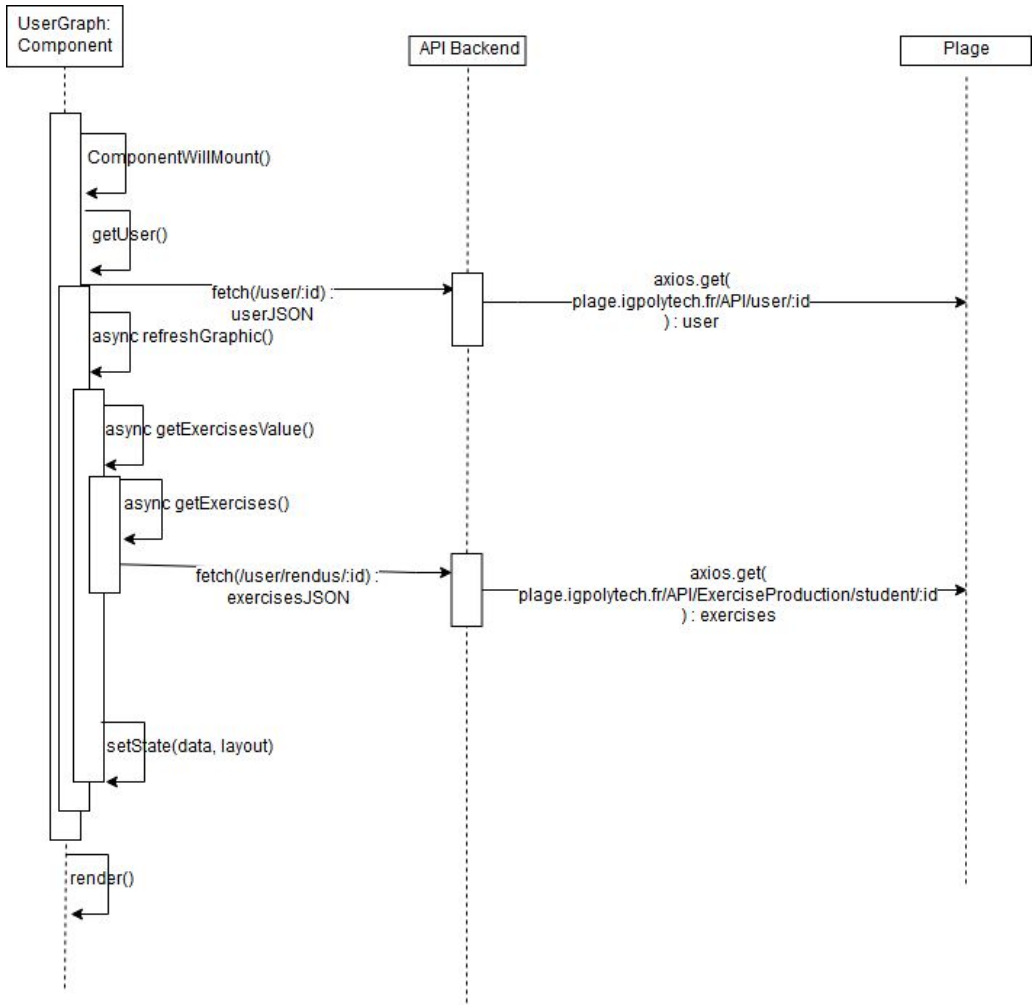
Interopérabilité

L'application utilise et offre des fonctionnalités interopérables. Tout d'abord, le front-end et back-end communiquent entre eux afin d'échanger des données, le front-end appelle le back-end qui lui répond. De plus, le back-end communique bien sûr avec l'application Plage réalisée par Stéphane Combes lors de la création et affichage des graphiques utilisateurs.

Enfin, l'application permet de télécharger les images des graphiques, qui pourront alors être utilisés par les utilisateurs. L'application propose également un certain nombre de routes indépendantes, affichant les graphiques des utilisateurs. Le but de ces derniers et de les solliciter depuis l'application Plage, pour que l'utilisateur final ne s'aperçoive pas du changement d'adresse.

Choix techniques

L'application a été réalisée avec React, et avec un back-end Node.JS. Voici ci-dessous le diagramme de séquence pour afficher le graphique répertoriant les résultats à tous les exercices d'un utilisateur :



Avant que le composant ne soit monté, on commence par demander au backend d'identifier l'utilisateur. Si aucun n'est trouvé, alors le processus s'arrête là et un message d'erreur s'affiche. Une fois celui-ci obtenu, on entame la procédure pour rafraîchir les informations du graphique. Il est important de noter que cette procédure est lancée périodiquement de façon dynamique afin de faire apparaître automatiquement les nouvelles données venant d'être créées dans Plage. Lors de l'appel à la méthode getExercises(), on envoie une demande au backend qui obtient alors les données de Plage et les renvoient. Il suffit ensuite de traiter chaque donnée correspondant à un exercice et de les mettre en forme dans la variable "data", qui est utilisée pour créer le graphe. Finalement, le composant appelle automatiquement la méthode render() après la modification de son état. Il pourra s'actualiser à chaque nouvelle modification.

Partie Graphique

La librairie utilisée afin de créer des graphiques est Plotly.js. Ce choix à été fait car il s'agit d'une librairie connue et disponible dans de multiples langages (Python, R, ...) et permettant de créer un nombre très important de graphiques différents, simples comme complexes. Cette librairie fonctionne de manière simple, puisqu'elle se base sur deux paramètres afin de créer des graphes, data et layout. Layout correspond au titre et la mise en forme, data étant un tableau de données JSON, correspondant chacune à un graphe différent. Pour notre application, nous n'avons eu besoin que d'avoir un graphe par composant.

Nous avons choisi un style assez épuré afin que ces graphiques puissent être incorporés sur le site de Plage.

Actions de chaque rôle

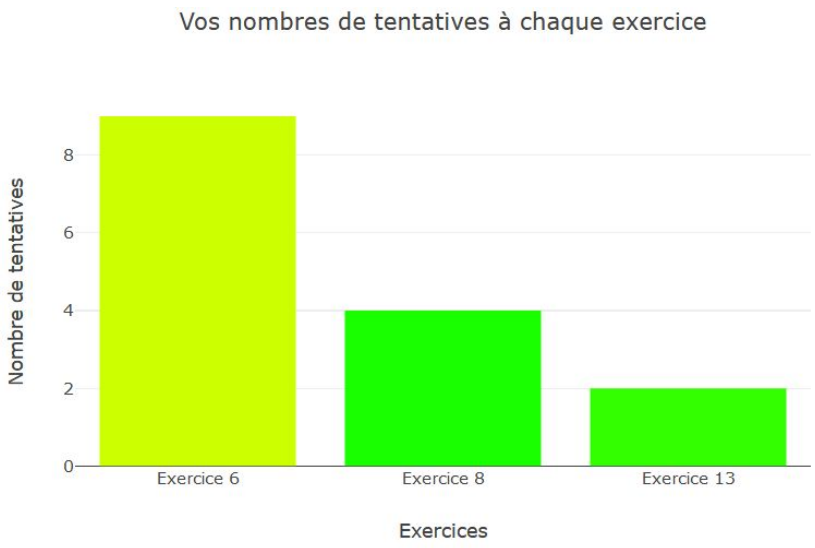
N'ayant pas accès à la base de données de Plage directement, il n'y a pas de vérification de connexion afin d'accéder aux graphiques des informations des résultats.

Elève :

Un élève connecté et possédant un identifiant moodle peut une fois connecté arriver au graphique récapitulant ses résultats d'exercices. La couleur de chaque barre dépend du résultat, avec rouge pour 0, jaune pour 50 et vert pour 100.

Introspection

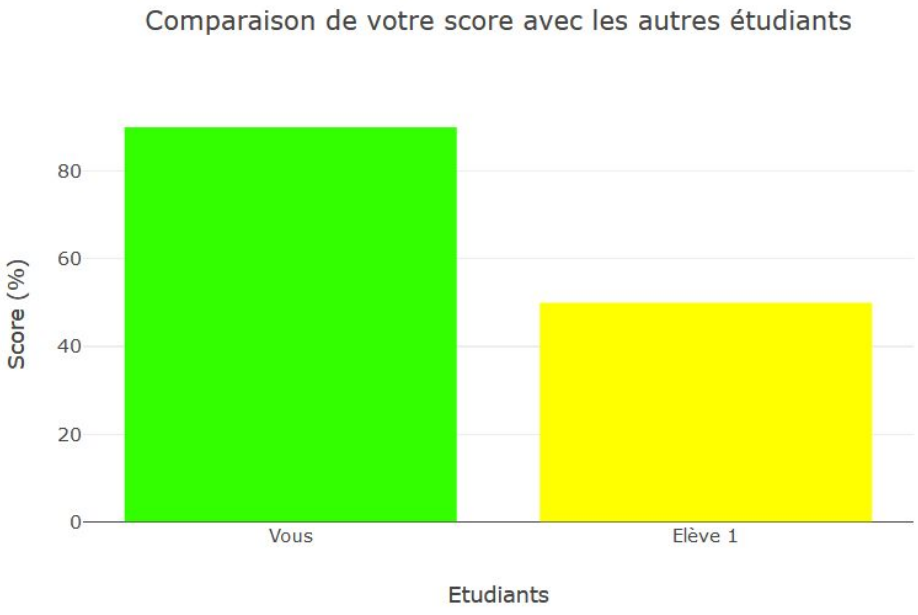
En cliquant sur le bouton Introspection, le graphique se recharge en faisant apparaître cette fois le nombre de tentatives pour chaque exercice.



Introspection

Il est également possible de comparer ses résultats à un exercice avec le résultat des autres élèves.

Comparaison sur l'exercice n°13

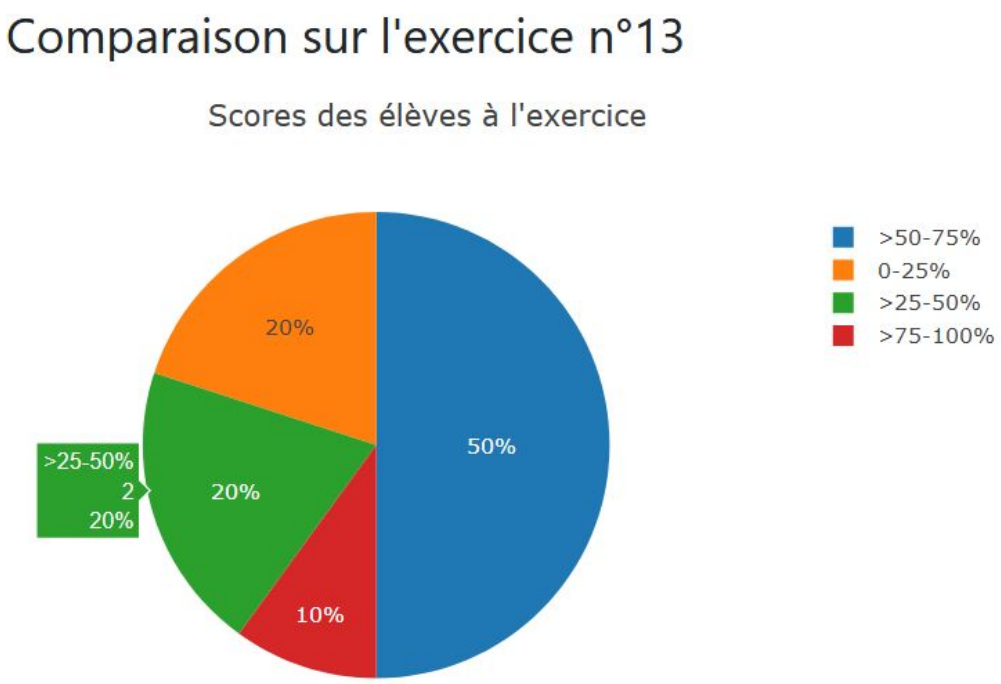


Si enfin l'utilisateur est connecté et possède un identifiant Plage, il peut accéder depuis sa page d'accueil aux graphiques qui le concernent.

En cliquant sur les liens d'exercices en bas de la page, il peut accéder au graphe de comparatif de ses notes à l'exercice par rapport aux autres étudiants.

Professeur :

Un professeur a également la possibilité d'obtenir un diagramme en secteur de la même vue, avec le résultat de tous les élèves par tranche de 25%. Ce graphique n'est pas accessible depuis une autre partie du site, pour y accéder, il faut accéder à la route directement, la route étant "/graph/teacher/:idExercice", on a par exemple http://plagedashboard.igpolytech.fr/graph/teacher/13.



Administrateur :

Un composant permettant d'afficher la fréquence des logs a été créé, malheureusement l'application Plage étant encore en développement, la route permettant d'obtenir tous les logs n'a pas été réalisée et il manque donc dans le back-end la récupération réussie des données de "/API/LMSLogging/".

La route néanmoins créée est "/graph/admin/logs", donc par exemple : http://plagedashboard.igpolytech.fr/graph/admin/logs

Installation

Pour déployer l'application :

Assurez vous d'avoir Node.JS et npm installés.

Les deux sont disponibles ensemble à cette adresse : https://nodejs.org/en/download/

Assurez vous également d'avoir git, ce dernier est disponible ici : https://git-scm.com/downloads

Placez vous en ligne de commande dans un dossier vide afin de télécharger séparement les deux projets PlageDashboard, entrez les commandes suivantes :

"git init" "git pull git@github.com:Hlecler/PlageDashboard.git"

"git init" "git pull git@github.com:Hlecler/PlageDashboardFront.git"

Puis, pour lancer les applications en local, il est nécessaire de changer dans le fichier config.json dans le dossier PlageDashboardFront/src la valeur de API_HOST, et la passer en "http://localhost:5000". Une fois cela fait, écrire "npm install" suivi de "npm start" dans la console et ce dans les deux projets. L'application sera alors disponible à l'adresse "http://localhost:3000". Pour déployer l'application en ligne après avoir réalisé des modifications, il faut écrire "git add ." "git commit -m "description de la modification"" puis enfin "git push dokku master". Après un certain temps, les modifications apparaîtront sur la version en ligne de l'application.

Un utilisateur créé afin de tester les fonctionnalitées est "test@plage.com", avec comme mot de passe "plage". Celui-ci est rattaché à l'utilisateur avec pour identifiant 2 dans l'application Plage.

Configuration

Routes du Front-end afin d'afficher les graphiques :

Route	Description	paramètres
/graph/user/:id	La route permettant d'obtenir les résultats d'un étudiant à tous ses exercices. Permet aussi d'obtenir son nombre de tentatives pour chaque exercice	id => identifiant de l'utilisateur sur Plage
/graph/exercice/:idUser/:idEx	La route permettant d'obtenir le comparatif des résultats d'un étudiant avec les autres sur un exercice	idUser => identifiant de l'utilisateur sur Plage idEx => identifiant de l'exercice comparé aux autres utilisateurs
/graph/teacher/:idEx	La route permettant d'obtenir les résultats des étudiants sur un exercice en diagramme en secteur	idEx => identifiant de l'exercice dont les résultats sont comparés

Routes de l'API :

Route	Description	paramètres	Retour	Verb HTTP
/login	La route de connexion, prend un id et un pwd dans le corps de la requête.	id = string pwd = string	application/json {"token": string, "userId": string}	POST
/logout	La route de déconnexion, prend un id et un token dans le corps de la requête.	Id = string Token = string	200	POST
/user/:id	Route permettant d'obtenir l'utilisateur Plage associé à l'id	Id = string, l'identifiant de l'utilisateur	application/json {"userid": string, "lastname": string, "lastname" : string, ...}	GET
/user/rendus/:id	Route permettant d'obtenir de Plage tous les exercices associés à un utilisateur	Id = string, l'identifiant de l'utilisateur	application/json { [{"ex_id" : string, "user_id" : string,	GET

			“mark” : string, ...}]}	
/exercises/rendus/:id	Route permettant d’obtenir de Plage tous les utilisateurs ayant réalisés un exercice	Id = string, l’identifiant de l’exercice	application/json { [{“ex_id” : string, “user_id” : string, “mark” : string, ...}]}	GET
/logs	Route permettant d’obtenir de Plage tous les logs de connexion, actuellement indisponibles.	/	application/json	GET

Variables d’environnement :

Back-End :

Nom	Valeur par défaut	Description
PORT	5000	Le port d’écoute du serveur
PLAGE_URL	"https://plage.igpolytech.fr"	L’adresse de l’application Plage

Front-end :

Nom	Valeur par défaut	Description
API_HOST	“http://plagedashboardapi.igpolytech.fr”	L’adresse de l’API.
TIMEOUT	2500	Le temps avant que le client ne timeout.