Serveur d'application PLM

---

1 15 janvier 2014

1.1 Données à conserver

- Énoncé de l'exercice ;
- code de l'élève ;
- nombre tentatives ;
- temps entre chaque tentatives ;
- changement d'exercice alors que l'actuel n'est pas résolu.

1.2 Structure stockage

- Un dépôt par utilisateur ;
- une branche par leçon ;
- une branche par exercice dans chaque branche de leçon.

- Message du commit : FAIL ou SUCCESS, l'ID de l'exercice, date, langage, indice utilisé, nombre de tests passés.
- Contenu du commit : code source, le message d'erreur s'il existe et l'énoncé.

Fichiers avec le code source, le template et le résultat.
Commit lors de la compilation. Quand il est en ligne, il push.

1.3 Questions

- Serveurs `http://www.loria.fr/~oster/plm-cloud/`, et `http://jlmserver-chmod0.appspot.com`
- Utilisation de Play ? Utilité de Google App Engine (a priori non compatible avec Play 2.x) ?

2 24 janvier 2014

Plutôt une branche par exercice. Google App Engine : première tentative. Play plus simple et plu portable. Ne pas dépendre d'une technique.
Voir ce que l'on attend de nous
Notre rapport doit permettre de faire gagner du temps à la prochaine équipe
Gestion de projet : ToDo list, planning des grandes étapes.
Regarder le code de la session de la PLM Git pourrait faire l'espion et la session
Mode professeur à reprendre. Fonctionnalités attendues (serveur d'utilisation) :

- explorer graphiquement l'évolution de chaque élèves ;
- mise en forme des données : représentation ;
-

Première version pour monter les gits. (premier tiers de notre travail) Commit en local. Monter automatiquement sur le serveur. Moulinette d'extraction des données.

Faire le git en local et le push sur un serveur

Trouver des étapes : établir un planning.

Faire un serveur qui centralise le code des élèves.

Gestionnaire de sessions en GIT qui commit plein de trucs Template, source, énoncé dans un seul commit. Chargeur qui récupère les données depuis le git.

ImportCloudSession.java SessionDB.java Exo, language, fichier : en mémoire ; le ISession-Kit : gère l'écriture sur le disque. Exercice.java -> SourceFile.java : template et le body (ce que l'élève à tapé) actuel Finir de mettre dans SessionDB le code de l'élève SourceFile : ce qui vient du jar SessionDB : ce qu'il faut écrire sur le disque Qui est en charge de getCompilableContent ?

Code session. Ramener code de l'élève dans SessionDB Les tests doivent passer Contenu du sessionDB dans le git et le contenu du template.

Bonus : Poster sur GitHub pour le feedback des problèmes : dans les issues PLM doit se loguer sur GitHub pour poster les messages d'erreurs.

Attentes du stages