Partie 1

Mai 2020

Ce TD s’appuie sur le cours intitul´e Framework Play! d’ Alexandre Collignon et Fr´ed´eric Drouhin. En cas de soucis (notamment les erreurs de compilation), une vid´eo sur Moodle peut apporter une aide pr´ecieuse.

1. **Projet 1 : Premiers pas**

**Objectifs :**

* Poser les bases de la programmation dynamique ;
* Premi`ere approche du mod`ele MVC ;
* Premi`ere approche d’une base de donn´ees.

# Installation de l’environnement

## Pr´e-requis :

* + - Version de Java 5 ;
    - Un ´editeur de texte.

## Mode d’emploi :

1. t´el´echarger Play ici ;

https://developer.lightbend.com/start/?group=play&project=play-java-seed

1. cr´eer un dossier ”play” ;
2. d´ecompresser l’archive dans ”play”.

**Analyse :** L’archive d´ecompress´ee est compos´ee de nombreux r´epertoires. Il est n´ecessaire de comprendre leur utilit´e pour maˆıtriser la programmation dy- namique. pour cela, vous vous cr´eez un memento de l’utilit´e de chacun de ces 5 r´epertoires (2.5points):

**Question 1**, App : Dossier application contient le contrôleur et la vue

**Question 2**, Conf : Dossier conf contient l’application et les routes

**Question 3**, lib : Dossier librairies

**Question 4**, public : Dossier public contient les médias (images) et scripts

**Question 5**, test : Dossier test sert à tester le fonctionnement

# Lancer l’application

Pour lancer l’application,

./sbt

* + - lancer le framework avec sbt avec
    - regarder les options possibles avec

help

* + - utiliser play avec

run

Pour v´erifier que tout fonctionne comme pr´evu, vous regardez sur votre mo- teur de recherche `a l’adresse http://localhost:9000 le r´esultat de votre script. Normalement, le serveur web de Play est en ´ecoute sur le port 9000.

Vous essayez de comprendre ce qu’il s’est pass´e `a travers une s´erie de questions (1point).

**Question 6** : Ou` sont stock´es les fichiers .class (les classes Java compil´ees)? : Dans le dossier target

**Question 7** : Citez 3 caract`eres non reconnus par l’encodage UTF8 utilis´e

par Play ? : les caractères chinois et arabes

Le fichier *conf/routes* regroupe les points d’entr´ees de l’application. Il d´efinit toutes les URL accessibles pour l’application. On peut voir la route

choisit dans : .

GET/ *RouteChoisie*

**Question 8** : Quelle est cette *RouteChoisie* ? (0.5 points): /assets/\*file

Pour une application Play, chaque URL est consid´er´ee comme un point d’entr´ee sp´ecifique. Play appelle les m´ethodes li´ees `a ces points d’entr´ees des **actions**. Ces actions sont d´efinies dans des classes Java appel´ees contrˆoleurs.

Le fichier *controllers.Application* dans app/controllers utilise des m´ethodes comme *render*() ou *index*() (m´ethode publique et statique).

**Question 9** :Que signifie l’aspect statique ? (0.5 points): si une méthode doit s’exécuter qu’une seule fois on peut la déclarer comme statique

**Question 10** :Que fait la m´ethode render() ? (0.5 points): envoie le contenu du serveur vers le client

On veut que le controleur (fichier *controllers.Application*) se pr´esente (1 point). Il faut donc ´ecrire ”Je suis le controleur.” en l’ins´erant dans comme suit

: render(”Je suis le controleur.”) . Normalement, la page a pris en compte ce message.

On veut que le viewer (fichier *main.html* dans *app/view/*) se pr´esente (1 point). Il faut donc ´ecrire ”Je suis le viewer et j’ob´eis.” . Je suis le viewer et j’ob´eis doit ˆetre ins´er´e au d´ebut du texte. Normalement, la page a pris en compte ce message.

# Am´eliorer le code

**Am´elioration :** R´ealiser un petit texte static avec css/html et des liens sur d’autres pages en utilisant l’environnement dynamique de play.

**Pour finir :** Il ne faut pas oublier de mettre sur GIT et remplir la feuille de r´eponses .txt.

# Auto-´evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| Travail r´ealis´e | Not´e sur | Ma note | Note du professeur |
| R´eponses ´ecrites | 5 |  |  |
| Correspondance controller/vue | 3 |  |  |
| Partie html (pratique) | 3 |  |  |
| Partie css (pratique) | 3 |  |  |
| Hyperliens propos´e(pratique) | 3 |  |  |
| Code bien ´ecrit (pratique) | 3 |  |  |
| Travail sur le Git (points bonus) | 2 |  |  |
| Auto-´evaluation correcte (point bonus) | 1 |  |  |

Table 1: Auto-´evaluation