TD2

October 2019

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), ce document apporte une aide pr´ecieuse. Il est disponible sur le moodle.

# Param`etres dynamiques et Formulaires

|  |
| --- |
| **Objectifs :**   * G´erer les Param`etres dynamiques ; * Maˆıtriser les formulaires. * Application de bases de donn´ees. |

## Param`etres dynamiques

### Partie th´eorique

On s’int´eresse au code suivant :

|  |
| --- |
| public class HomeController extends Controller { public Result helloworld() { return ok(helloworld.render("Lilou")) ;  }  } |

1

2

3

4

5

Listing 1: MVC1

|  |
| --- |
| @(firstname: String) <html>  <body>  Hello @firstname  </body>  </html> |

1

2

3

4

5

6

Listing 2: MVC2

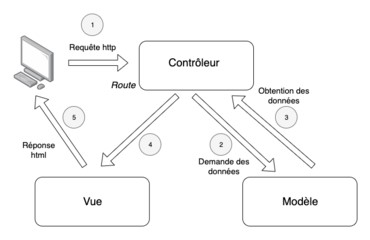


Figure 1: Rappel Mod`ele vue controleur

Dans l’exemple ci-dessus(1pt), la partie **mod`ele-ou-vue-ou-controleur** *(entourer l’un des 3)* de MVC1 passe un argument `a la partie **mod`ele-ouvue-ou-controleur***(entourer l’un des 3)* de MVC2.

MVC1 : Contrôleur

MCV2 : Vue

Dans cet exemple, on a :

* la variable contenant le pr´enom not´ee Lilou.
* et une vue dynamique qui change en fonction de cette variable.
* Par rapport `a la figure 1, quel est le num´ero du lien correspondant `a la relation MVC1 et MVC2 ? MVC1 = 4 et MVC2 = 5

### Partie pratique

Ecrivez le programme MVC3 dans la partie correspondante :

|  |
| --- |
| public class HomeController extends Controller  {  public Result helloworld(String fname)  { return ok(helloworld.render(fname)) ;  }  } |

1

2

3

4

5

6

7

Listing 3: MVC3

La route aussi se doit d’ˆetre modifi´ee :

|  |
| --- |
| GET /helloworld controllers.HomeController.helloworld(fname: String  ) |

1

Listing 4: route 1

Par contre, vous pouvez prendre la vue MVC2, en la nommant helloworld.scala.html. Que se passe-t-il si vous entrez les adresses suivantes ? 1,5 pt:

|  |  |
| --- | --- |
| localhost:9000 | Nous accédons au site de base |
| localhost:9000/helloworld | Nous accédons à une page : Bad request |
| localhost:9000/helloworld/?fname=lilou | Nous accédons à la page souhaitée : |

Enfin, si la route est *route*2, que se passe-t-il ?

|  |
| --- |
| GET /helloworld controllers.HomeController.helloworld (fname:  String ?="world") |

J’accède à la même page que précédemment mais depuis localhost:9000/helloworld ainsi que depuis localhost:9000/helloworld/?fname=lilou

Listing 5: route 2

**Pour aller plus loin :** Cette m´ethode est la m´ethode GET. Les donn´ees sont ins´er´ees dans l’URL. Il existe une autre m´ethode : la m´ethode POST. Dans ce cours, on la survole simplement. La m´ethode POST envoit un en-tˆete et un corps de message au serveur. Le corps est g´en´eralement constitu´e des donn´ees entr´ees dans le champ de formulaire par l’utilisateur.

Voici, un tableau comparatif des deux m´ethodes :

|  |  |
| --- | --- |
| **GET** | **POST** |
| Seule une quantit´e limit´ee de donn´ees | Une grande quantit´e de donn´ees |
| peut ˆetre envoy´ee car les donn´ees | peut ˆetre envoy´ee car les donn´ees |
| sont envoy´ees dans l’en-tˆete. | sont envoy´ees dans le corps. |
| Non s´ecuris´ee car les donn´ees | S´ecuris´ee car les donn´ees |
| sont expos´ees dans la barre d’URL. | ne sont pas expos´ees dans la barre d’URL. |
| Plus efficace et utilis´e plus que POST | POST est moins efficace et utilis´ee moins que GET. |
| Les param`etres restent dans | Les param`etres ne sont pas . |
| l’historique du navigateur car | enregistr´es dans l’historique |
| ils font partie de l’URL | du navigateur. |
| Les requˆetes GET sont r´e-ex´ecut´ees | Le navigateur pr´evient |
| mais ne peuvent pas ˆetre soumises | g´en´eralement l’utilisateur que |
| au serveur si le code HTML est stock´e | les donn´ees devront |
| dans la m´emoire cache du navigateur. | ˆetre soumises `a nouveau. |
| Seuls les caract`eres ASCII sont autoris´es. | Pas de restrictions. |
| Visible par tout le monde (affich´ee dans l’url) et limite le nombre d’informations `a envoyer. | Non visibles dans l’URL. |
| Peut ˆetre mis en cache | Ne peut ˆetre mis en cache. |

## Ajout d’un formulaire

Dans la partie vue, ´ecrivez le code Formulaire1.

|  |
| --- |
| @()  <html>  <form action = routes.HomeController.helloworld()>  {  <input name="fname"/>  <input type="submit" value="Say hello"/>  }  </form>  </html> |

1

2

3

4

5

6

7

8

9

Listing 6: Formulaire1

Que fait le code Formulaire1 (2.5 pts) ?

Le code du Formuaire1 permet de demander le prénom pour rediriger sur une potentiel page attitré pour le nom

**Un peu de pratique :** Maintenant, cr´eez un formulaire stylis´e qui entre les donn´ees suivantes :

prenom;

@()

<html>

<form action = routes.HomeController.helloworld()>

{

<input surname="fsurname"/>

<input type="submit" value="Say hello"/>

}

</form>

* </html>

age ;

@()

<html>

<form action = routes.HomeController.helloworld()>

{

<input age="fage"/>

<input type="submit" value="Say hello"/>

}

</form>

* </html>

Adresse ;

@()

<html>

<form action = routes.HomeController.helloworld()>

{

<input adresse="fadresse"/>

<input type="submit" value="Say hello"/>

}

</form>

* </html>

# Evaluation

## Auto-´evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| Travail r´ealis´e | Not´e sur | Ma note | Note du professeur |
| R´eponses ´ecrites | 5 | 5 |  |
| Petites exercices r´ealis´es | 3 | 3 |  |
| Partie vue du formulaire | 4 | 4 |  |
| Partie contrˆoleur du formulaire | 4 | 4 |  |
| Code bien ´ecrit (pratique) | 3 | 3 |  |
| Travail sur le Git | 1 | 1 |  |
| Auto-´evaluation correcte (point bonus) | 1 | 1 |  |