

Labo Spring 4

Attribut de session

Objectif : Définir et utiliser des attributs liés à la session de l'utilisateur ainsi qu'un service

Dans la suite de l'exercice, en fonction de l'âge et du hobby de l'enfant, un cadeau va lui être attribué. Le cadeau attribué sera affiché dans la page `gift.jsp`.

Le calcul du cadeau va se faire dans la méthode `get` du controller associé à cette page, soit la classe `GiftController`. La méthode `post` de `InscriptionController` devra faire une redirection vers `GiftController`.

Name

Age

☐ Boy ☐ Girl

What's your preferred hobby?

```
InscriptionController
{
    get
    {
    }
    post
    { }
}
```

```
GiftController
{
    get
    {
    }
    post
    { }
}
```

Congratulations John, you will receive a DVD about sport !

Etape 1 – L'objet de type *User* placé en session

Pour que la classe *GiftController* puisse calculer le cadeau de l'enfant, il faut pouvoir récupérer les informations encodées par l'enfant et récupérées dans l'objet de type *User* via la méthode *post* de *InscriptionController* (cet objet a été annoté *@ModelAttribute*). Cet objet n'existe que le temps de la requête ; il n'est donc pas accessible dans un autre controller.

Pour que cet objet persiste durant toute la session de l'utilisateur, il faut le déclarer attribut de session dès sa création dans *InscriptionController*.

Utilisez pour ce faire l'annotation *@SessionAttribute* en lui donnant un nom (ex : "*currentUser*"). Le mieux est de déclarer une constante pour ce nom, car ce nom va être utilisé à plusieurs endroits. Déclarez cet attribut via *@ModelAttribute*. ~~Prévoyez une méthode *initModel* qui ajoute cet attribut session au *Model*.~~

```
@Controller
@RequestMapping(value="/inscription")
@SessionAttributes({InscriptionController.CURRENTUSER})
public class InscriptionController {

    protected static final String CURRENTUSER = "currentUser";

    @ModelAttribute(CURRENTUSER)
    public User user() {
        return new User();
    }
}
```

Restez cohérent en utilisant le même nom pour cet attribut, y compris dans sa déclaration comme argument de la méthode *post*.

```
@RequestMapping(value="/send", method=RequestMethod.POST)
public String getFormData(Model model,
                          @Valid @ModelAttribute(value=CURRENTUSER) User user,
                          final BindingResult errors) {
```

Faites de même dans la page *userInscription.jsp*.

```
<form:form id="inscription"
           method="POST"
           action="/childgift/inscription/send"
           modelAttribute="currentUser">
```

Etape 2 – Service *GiftService*

Le cadeau à attribuer à l'enfant doit être calculé par une classe de type service sur base de l'âge de l'enfant et de son hobby.

Dans le package *service* dans *src/main/java/com.spring.henallux*, créez la classe *GiftService* qui contient une méthode public *chooseGift* qui retourne le cadeau (String) et qui reçoit en argument un hobby (String) et un âge (int). Cette méthode attribue un cadeau à l'enfant sur base des règles suivantes (par exemple) :

- Avant 5 ans : un puzzle concernant son hobby
- Entre 5 et 10 ans : un DVD concernant son hobby
- Après 10 ans : un livre concernant son hobby

Cette classe doit être annotée *@Service*. Ceci permettra d'y avoir accès par **injection de dépendance**.

Etape 3 – Classe *GiftController*

Créez la classe *GiftController* dans le package *controller*.

Récupérez dans cette classe la référence vers l'attribut *currentUser* placé dans la session via les annotations *@SessionAttribute* et *@ModelAttribute* sur base de son nom ("*currentUser*").

```
@Controller
@RequestMapping(value="/gift")
@SessionAttributes({"currentUser"})
public class GiftController {
```

```
    @RequestMapping(method=RequestMethod.GET)
    public String home(Model model,
        @ModelAttribute(value="currentUser") User user) {
```

Dans la classe *GiftController*, récupérez par injection de dépendance une référence vers un objet de la classe *GiftService*.

Dans la méthode *get*, appelez sur cet objet de type *GiftService* la méthode *chooseGift* en lui passant l'âge et le hobby de l'enfant. Ajoutez le cadeau ainsi calculé comme attribut dans le *Model*, afin de pouvoir l'afficher dans la page *gift.jsp*.

Etape 4 – Classe *InscriptionController*

N'oubliez pas de modifier la méthode *post* de *InscriptionController* afin de rediriger vers le controller *GiftController*.

