

Tutorial interactif :

<http://itutorial.thymeleaf.org/>

Documentation Thymleaf 2.1

Formulaires

<http://www.thymeleaf.org/doc/tutorials/2.1/thymeleafspring.html#creating-a-form>

Itérations

<http://www.thymeleaf.org/doc/tutorials/2.1/usingthymeleaf.html#iteration>

Conditions

<http://www.thymeleaf.org/doc/tutorials/2.1/usingthymeleaf.html#conditional-evaluation>

Littéraux, chaînes et opérateurs

<http://www.thymeleaf.org/doc/tutorials/2.1/usingthymeleaf.html#literals>

Atelier Conf partie 2

1. Intégrer Bootstrap et JQuery dans le projet

Ajouter les dépendances suivantes dans pom.xml

```
<dependency>
    <groupId>org.webjars</groupId>
    <artifactId>jquery</artifactId>
    <version>3.2.1</version>
</dependency>
<dependency>
    <groupId>org.webjars</groupId>
    <artifactId>bootstrap</artifactId>
    <version>3.3.7-1</version>
</dependency>
```



2. Ajouter un modèle thymeleaf

- Dans le dossier templates/layout (le sous dossier layout doit être créé), ajouter le modèle default.html

```
<!DOCTYPE html>
<html lang="fr" xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout">

<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<title>Default title</title>
<link href="webjars/bootstrap/3.3.7-1/css/bootstrap.css"
      rel="stylesheet" />
</head>
<body>
  <section layout:fragment="content" class="container">
    <p>Contenu de la page</p>
  </section>
  <script src="/webjars/jquery/3.2.1/jquery.js"></script>
  <script src="webjars/bootstrap/3.3.7-1/js/bootstrap.js">
    </script>
</body>
</html>
```

3. Appliquer le modèle aux vues

Dans chaque vue attachée au modèle :

- Remplacer la balise ouvrante html par :

```
<html xmlns:th="http://thymeleaf.org"
```

```
xmlns:layout="http://www.ultraq.net.nz/thymeLeaf/Layout"  
layout:decorator="Layout/default">
```

- b. Le contenu de la vue doit être dans un élément ayant l'attribut `layout:fragment="nom_fragment"`
4. Créer la classe entité Session, attributs :
 - id
 - theme
 - description
 - date
 - intervenant
2. Créer les interfaces SessionRepository et SessionService et la classe SessionServiceImpl
3. Créer le contrôleur SessionController et les vues correspondantes pour réaliser les actions suivantes : ajout, liste, modification et suppression, sessions / intervenant