

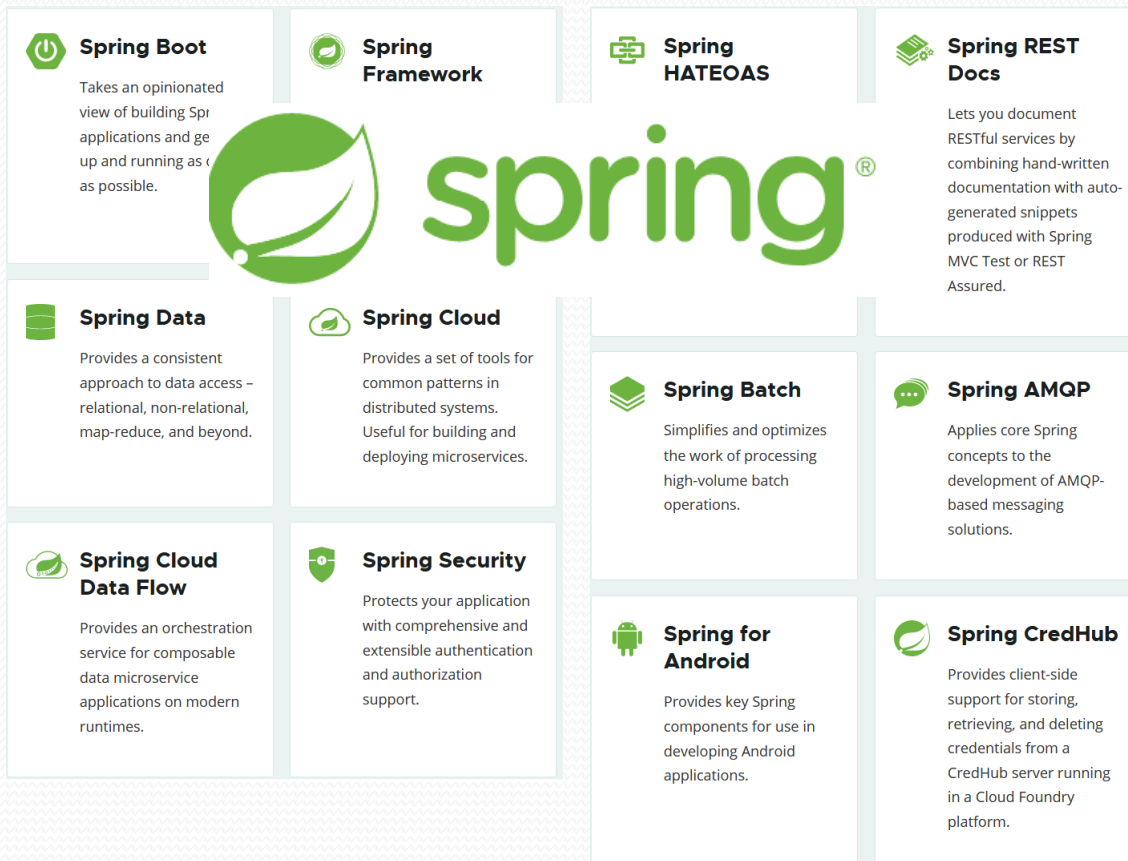
Spring

Cedric Dumoulin

Inscription sur Moodle

- Codes donnés en cours

Spring, Spring Boot, Spring Framework ... De quoi parle t-on ?



Lecture

Spring Framework Overview

- <https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/overview.html#overview>
- Quel est la base de tout ?
 - Dans la galaxie Spring, quel est le projet originel ?
- Comment se positionne Spring Boot
- Pourquoi Spring a-t-il été inventé, alors qu'il y avait JEE ?

Atelier

Spring Boot quickstart

- Spring Quickstart Guide
 - <https://spring.io/quickstart>
 - Utilisez gradle + Eclipse ou gradle+Intelij
- Questions :
 - Quelle est la différence entre Spring et Spring Boot ?
 - A quoi sert Spring Initializr ?
 - Quelle instruction dans le programme déclenche le lancement de Tomcat ?
 - Regardez le contenu de la page WEB reçu. Est-ce de l'HTML ou autre chose ?
 - Modifiez le programme pour qu'il fonctionne à l'URL :
`http://localhost:8080/bonjour`

Atelier

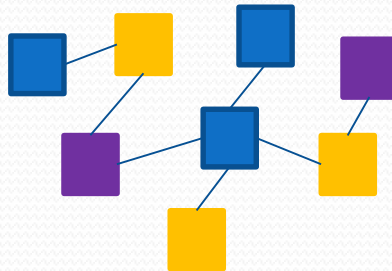
Building a RESTful Web Service

- Faites l'atelier :
 - <https://spring.io/guides/gs/rest-service/>
 - Utilisez gradle + Eclipse
- Questions
 - Qu'est-ce qu'un controller ?
 - A quoi servent et qu'elle est la différence entre @GetMapping @PostMapping et @RequestMapping ?
 - Regardez le contenu de la page WEB reçu. Est-ce de l'HTML ou autre chose ?
 - Quel mécanisme permet de transformer l'objet retourné par greeting(...) en du texte JSON ?
 - Pourquoi ce mécanisme est-il activé ?
 - Comment fait-on pour récupérer le paramètre 'name' de la requête http ?

Container et IoC

Le problème

- Une application == un **graphe d'objets** (instances de classes) **interagissant** entre eux
- Un objet dépend d'autres objets pour fonctionner



- Questions :
 - Qui crée (instancie) les objets ?
 - Qui connecte les objets ?
 - Qui gère les problèmes de dépendances (ex. : l'ordre de création des objets)

Avant

- Le programmeur gère lui-même les problèmes ☹
- C'est l'objet qui :
 - crée les objets dont il a besoin
 - `A a = new A();`
 - crée les dépendances
 - Gère l'ordre de création

Spring Framework

- Un conteneur 'léger'
- IoC
 - Le composant déclare les composants dont il a besoin
 - Le conteneur les instancie et les lui passe
- Le conteneur
 - 'scan' les composants,
 - établie la liste des dépendances et l'ordre de création
 - instancie les composants nécessaire
 - crée les connections
 - en fournissant le composant instancié au demandeur (dans le constructeur, ou un setter)

Spring Framework

IoC – Inversion of Control

- ➔ Le programmeur déclare son graphe d'objet
- Le conteneur se charge d'instancier le graphe d'objet

Comment déclarer les composants et leurs dépendances ?

- Plusieurs choix possibles :
 - dans des **fichiers xml** (historique)
 - par **programmation**
 - par **annotation** Java (@annotation)
- La doc contient beaucoup d'exemple XML
- Bootstrap est plutôt @annotation
- Dans les lectures suivantes :
 - ne pas s'attarder sur les exemples XML
 - regarder les exemples avec les @annotation

Lecture:

What is Spring Framework: Dependency Injection in Java

- <https://www.marcobehler.com/guides/spring-framework>
 - jusque « Spring's Aspect-Oriented Programming (AOP) » non compris
 - Si vous lisez vite, vous pouvez le lire
- Questions
 - A quoi sert @Configuration
 - Qu'est que l'ApplicationContext
 - Qu'est-ce qu'un « Bean Spring »
 - A quoi sert l'annotation @Bean sur une méthode ?
 - A quoi sert @ComponentScan ?
 - A quoi sert l'annotation @Component sur une classe ?
 - Quelle est la relation entre @Controller, @Service, @Repository et @Component ?
 - A quoi sert @Autowired ?
 - @Autowired est-il obligatoire ?
 - Comment Spring sait-il quel Bean doit être injecté ?

Lecture

Additional Modules

- https://www.marcobehler.com/guides/spring-framework#_additional_modules
- Questions
 - Qu'est-ce que « **Spring Web MVC** »?
 - Qu'est-ce que « Spring's Testing Framework »

Spring MVC

Lecture

What is Spring MVC: @Controllers & @RestController

- <https://www.marcobehler.com/guides/spring-mvc>
 - jusque « How to write a @Controller in Spring » non compris
- Questions
 - Spring MVC utilise t-il des Servlets ? Si oui, combien.
 - Quel est l'autre nom du DispatcherServlet ?
 - Quel est le rôle du DispatcherServlet ?

Lecture

How to write a @Controller in Spring

- https://www.marcobehler.com/guides/spring-mvc#_how_to_write_html_with_controllers
 - Jusque “How to write XML/JSON with @RestController” non compris
- Questions
 - Comment déclare t-on un Controller
 - A quoi sert un Controller
 - Le nom d’une méthode a t-il une influence sur la requête reçu par cette méthode
 - Comment associe t-on une méthode de contrôleur à une URL ?
 - Comment récupère t-on les paramètres d’une requête ?
 - Quid de la conversion des types des paramètres
 - Suite slide suivant --→

Lecture (suite)

How to write a @Controller in Spring

- Questions (suite)
 - Qui construit la réponse à la requête : la méthode du contrôleur, ou le template ?
 - A quoi sert le « Model » dans une méthode annoté @RequestMapping (ou assimilé)
 - Quel est le type de retour de cette méthode ?
 - A quoi correspond le nom retourné par la méthode
 - Comment Spring fait-il pour mapper ce nom vers le nom du fichier (de la ressource) contenant le template ?
 - Comment faire simplement pour indiquer a Spring d'utiliser un système de template particulier, comme Thymleaf ?
 - Séparation des préoccupations (separation of concerns) :
 - Quelle est la préoccupation d'une méthode de contrôleur ?
 - Quelle est la préoccupation du template ?

Atelier

Serving Web Content with Spring MVC

- <https://spring.io/guides/gs/serving-web-content/>
- Modifiez l'application pour ajouter une page qui sera accessible a l'UrL :
`http://localhost:8080/bonjour?nom=Toto`
- Cette page affichera « Bonjour Toto » , ou Toto est la valeur passé dans l'argument 'nom' de l'URL

Atelier

Handling Form Submission

- <https://spring.io/guides/gs/handling-form-submission/>
- Questions
 - Pourquoi y a-t-il deux méthodes associées à l'url 'greeting' ?
 - Les deux méthodes retournent-elles le même nom de template ? Pourquoi ?
 - A l'aide de quoi récupère t-on les valeurs du formulaire ?
 - Comment est assuré le mapping entre les champs dans le formulaire et les propriété de l'objet ?

Validation

Bibliographie

- Validating Form Input
 - [2018] <https://spring.io/guides/gs/validating-form-input/>
- Validation
 - Spring MVC Form Handling Tutorial and Example
 - [2017] <http://www.codejava.net/frameworks/spring/spring-mvc-form-handling-tutorial-and-example>
- Doc
 - [2018] <https://docs.spring.io/spring/docs/5.1.2.RELEASE/spring-framework-reference/core.html#validation>
 - [2018] <https://docs.spring.io/spring/docs/5.1.2.RELEASE/spring-framework-reference/core.html#validation-beanvalidation>

Documentation

- Spring Validation Example – Spring MVC Form Validator
 - [2016] <http://www.journaldev.com/2668/spring-validation-example-mvc-validator>
 - Exemples avec 3 formes de validation (javax.validation, custom annotations, custom validator)
- Spring MVC Form Validation Example with Bean Validation API
 - [2018] <http://www.codejava.net/frameworks/spring/spring-mvc-form-validation-example-with-bean-validation-api>

Que valider ?

- Il faut valider pour se garantir :
 - De mauvaises saisies dans les formulaires
 - De données saisies non valables pour le métier
- Les mauvaises saisies peuvent être détectés par:
 - la conversion de la requête http → objet command
- Les contraintes métiers peuvent être détectés par:
 - des objets de validation

Validation

- Action de valider des données en fonction du métier
 - ex: $0 \leq \text{age} < 150$
- Plusieurs possibilités avec Spring
 - les technique se sont empilées avec le temps
- Les plus récentes:
 - Bean Validation 1.0 (JSR-303)
 - and Bean Validation 1.1 (JSR-349)

Atelier

Validating Form Input

- <https://spring.io/guides/gs/validating-form-input/>
- Questions
 - A quoi sert la classe PersonForm ?
 - Comment, dans le template, indique t-on le nom de l'objet qui sert a transferer les valeurs des champs ?
 - Qu'est ce qui declenche la validation ?
 - La validation se fait-elle du coté serveur ou du coté client ?
 - Que fait la methode addViewControleur() ?
 - Que fait 'redirect:/results' ?
 - A quoi sert l'objet de type BindingResult, et où faut-il le déclarer ?
 - Comment affiche t-on les erreurs dans la page html ?

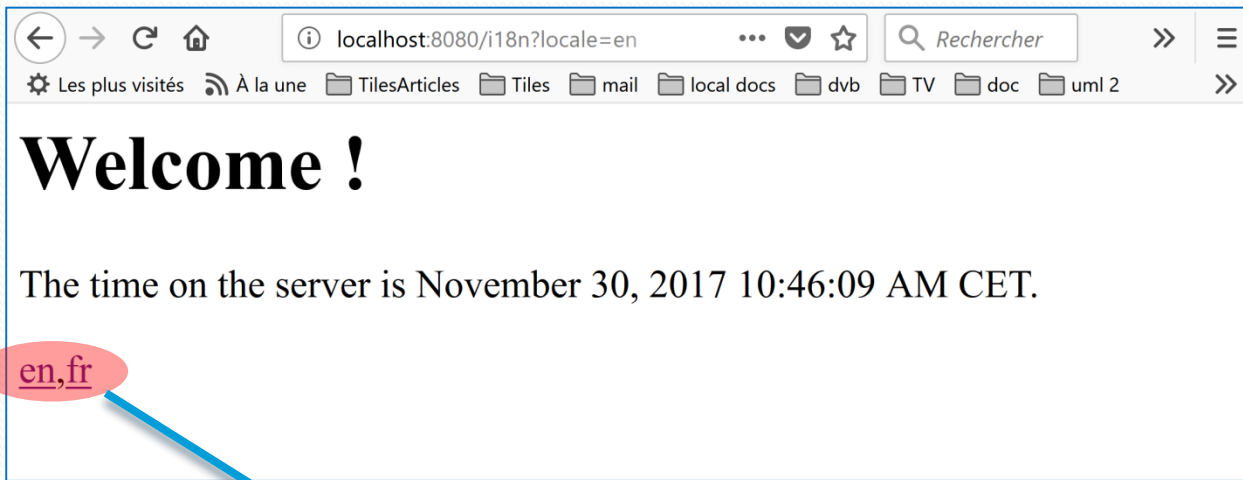
Autre méthode de validation possible

- Définir des objets « Validator »
 - Voir [Spring Validation Example – Spring MVC Form Validator]
 - <https://www.journaldev.com/2668/spring-validation-example-mvc-validator>

Internationalisation

i18n

Demo



Bibliographie

- [2018] Internationalization in Spring Boot
 - <http://www.baeldung.com/spring-boot-internationalization>
- [old] Spring MVC Internationalization (i18n) and Localization (L10n) Example
 - <https://www.journaldev.com/2610/spring-mvc-internationalization-i18n-and-localization-l10n-example>

Bibliographie

- <http://viralpatel.net/blogs/spring-3-mvc-internationalization-i18n-localization-tutorial-example/>
- I18n avec changement possible du local :
 - <https://www.journaldev.com/2610/spring-mvc-internationalization-i18n-and-localization-lion-example>
 - <http://www.baeldung.com/spring-boot-internationalization>

Principe

- Utilise la notion de « Locale » : Pays_langue
 - FR_fr, US_en
- Dans le code, on **utilise des constantes** à la place du texte
 - thymleaf `<h1 th:text="#{greeting}"></h1>`
 - JSP `<spring:message code="page.home.hello"/>`
- Les constantes sont définies dans des fichiers
 - Un fichier par langue

 messages_fr.properties

```
page.home.hello=Bonjour le Monde !  
page.home.timeOnServer=L'heure sur le serveur est
```

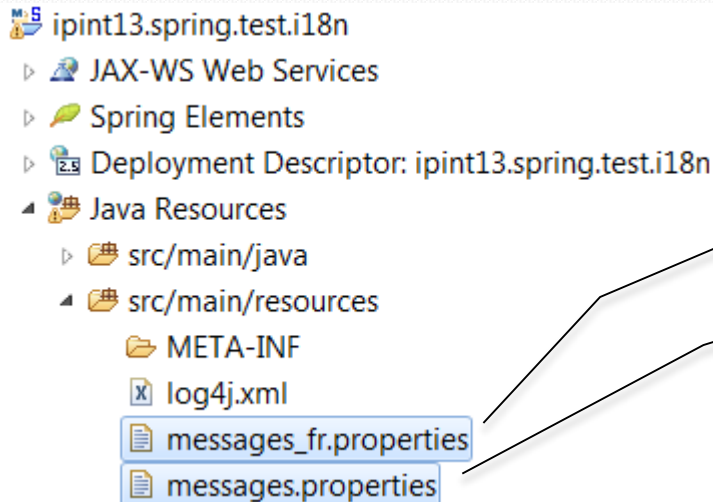

Fichiers de langues

- Contiennent la définition des constantes
 - Pairs `nom=valeur`
- A placer dans
 - `src/main/resources`
- Un fichier par langue
- Un fichier par défaut
- Tous les fichiers ont le même nom
 - Mais des extensions en fonction du locale:
 - `messages.properties`
 - `messages_FR_fr.properties`

Spring MVC

Les fichiers properties

- Permet d'utiliser des constantes dans les pages
- Les définitions sont dans des fichiers *.properties
- Dans répertoire `src/main/resource/*`



fichier local = fr

fichier par défaut

```
messages.properties
page.home.title=Home
page.home.hello=Hello World !
page.home.timeOnServer=The time on the server is
```

pas défini = venant
du fichier par défaut

```
messages_fr.properties
page.home.hello=Bonjour le Monde !
page.home.timeOnServer=L'heure sur le serveur est
```

I18n avec Spring et JSP

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib uri="http://www.springframework.org/tags" prefix="spring"%>
<%@ page session="false" %>
<html>
<head>
<title><spring:message code="page.home.title"/></title>
</head>
<body>
<h1>
<spring:message code="page.home.hello"/>
</h1>

<P> <spring:message code="page.home.timeOnServer"/> ${serverTime}. </P>
</body>
</html>
```

La taglib utilisée

Le message

l18n avec Spring et thymeleaf

- Utiliser la syntaxe `{key}`

```
<h1 th:text="{greeting}"></h1>
```

I18n avec Spring et thymeleaf

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">

<head>
  <title th:text="#{home.title}" /></title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
  <h1 th:text="#{home.title}"></h1>

  <p th:text="#{home.welcome}" > ${serverTime}. </p>

</body>
</html>
```

Laisser l'utilisateur choisir sa langue

- Il faut ajouter un choix du local
- Il faut prendre en compte le changement
- ➔ voir tuto
 - [2018] <http://www.baeldung.com/spring-boot-internationalization>
 - [2016] <http://viralpatel.net/blogs/spring-3-mvc-internationalization-i18n-localization-tutorial-example/>

Changer le locale dynamiquement

- Ajouter un parametre locale=?? à la requête

```
http://monsite/monappli/page.html?locale=en
```

- Peut se faire grace à des liens dans les pages

```
<a href="/home?locale=en" >en</a>,  
<a href="/home?locale=fr" >fr</a>
```

- Prendre en compte le changement dynamique dans l'application
 - Configurer Spring !

Welcome !

The time on the server is November 30, 2017 10:46:09 AM CET.

[en,fr](#)

Prendre en compte le changement dynamique dans l'application

- Il faut aussi ajouter:
 - un *LocaleResolver*
 - un *LocaleChangeInterceptor*
 - Pour intercepter le paramètre locale
- Voir l'atelier

Atelier

Guide to Internationalization in Spring Boot

- <https://www.baeldung.com/spring-boot-internationalization>
- Questions
 - A quoi sert le `localResolver` ?
 - A quoi sert `LocaleChangeInterceptor` ?

Persistance avec JPA

Atelier

Accessing Data with JPA

- <https://spring.io/guides/gs/accessing-data-jpa/>
- Questions
 - A quoi sert un 'Repository' ?
 - Comment declare t-on un repository, et comment l'associe t-on à un type d'entité ?
 - Qui ecrit l'implementation du Repository ?
 - A quoi sert le Bean de type CommandLineRunner ?
 - Quel base de données est utilisées ?
 - Peut-on utiliser une autre BD ? Comment ?

Atelier

Accessing JPA Data with REST

- <https://spring.io/guides/gs/accessing-data-rest/>
 - A faire en complément du précédent atelier
- Questions

Atelier

Utiliser une autre BD, Type Derby server

- A l'aide du guide suivant :
 - Accessing data with MySQL
 - <https://spring.io/guides/gs/accessing-data-mysql/>
- Modifier votre projet précédent pour utiliser Derby server.
- Tips :
 - Spécifier la dépendance vers la library Derby
 - Paramétrer :
 - `src/main/resources/application.properties`
- Question
 - Indiquez les modifications principales à effectuer pour utiliser une autre DB.