

MODULE Spring Exercices / Spring MVC

Spring - Exercices

Exercices sur Spring MVC

- Mise en oeuvre de Spring Initializr et Spring Boot
- Exercice 01 : une application Spring MVC basique
- Exercice 02 : une application Spring MVC dynamique
- Exercice 03 : une application Spring MVC riche
- Exercice 04 : une application Spring MVC avec gestion des erreurs

Spring - Exercices

Préliminaire 1 : mise en oeuvre de Spring Initializr

Mise en œuvre de Spring Initializr

Spring Initializr est un service d'initialisation de projet Spring.

Il est disponible au travers d'une page web (<https://start.spring.io>) ainsi que de librairies qui peuvent être intégrées dans un IDE, voire utilisées en direct dans un programme utilisateur.

Spring Tool Suite 4 intègre Spring Initializr et il est recommandé d'utiliser ses wizards mettant en œuvre Initializr.

Mise en œuvre de Spring Initializr

Néanmoins, dans le cas où l'IDE ne supporterait pas Initializr, il est possible de récupérer depuis le site une archive représentant le projet, et de déplier cette archive

Ouvrir un onglet de navigateur pour l'URL :

```
https://start.spring.io
```

⇒ la page suivante apparaît.

Mise en œuvre de Spring Initializr

The screenshot shows the Spring Initializr web application. The interface is clean and modern, with a light gray background and green accents. The top navigation bar includes the Spring Initializr logo, the text "Bootstrap your application", and links for "Light UI", "Github", "Twitter", and "Help". The main content area is divided into several sections: "Project" (Maven Project, Gradle Project), "Language" (Java, Kotlin, Groovy), "Spring Boot" (2.2.1 (SNAPSHOT), 2.2.0, 2.1.10 (SNAPSHOT), 2.1.9), "Project Metadata" (Group: com.example, Artifact: demo, Options), and "Dependencies" (Search dependencies to add, Selected dependencies). The "Dependencies" section shows a search bar with the text "Web, Security, JPA, Actuator, Devtools..." and a message "No dependency selected". At the bottom, there are three buttons: "Generate - Ctrl + G", "Explore - Ctrl + Space", and "Share...".

Spring Initializr
Bootstrap your application

Light UI Github Twitter Help

Project Maven Project Gradle Project

Language Java Kotlin Groovy

Spring Boot 2.2.1 (SNAPSHOT) 2.2.0 2.1.10 (SNAPSHOT) 2.1.9

Project Metadata

Group
com.example

Artifact
demo

> Options

Dependencies

Search dependencies to add Selected dependencies

Web, Security, JPA, Actuator, Devtools... No dependency selected

Generate - Ctrl + G Explore - Ctrl + Space Share...

© 2013-2019 Pivotal Software
start.spring.io is powered by
[Spring Initializr](#) and [Pivotal Web Services](#)

Mise en œuvre de Spring Initializr

Choisir un nom de package, un nom d'application, et entrer la description du projet.

Entrer ensuite le mot-clé « MVC » dans le champ de recherche de la partie **Dependencies**.

Dans la liste de propositions qui apparaît, cliquer sur **Spring Web**.

Cliquer ensuite sur **Generate**.

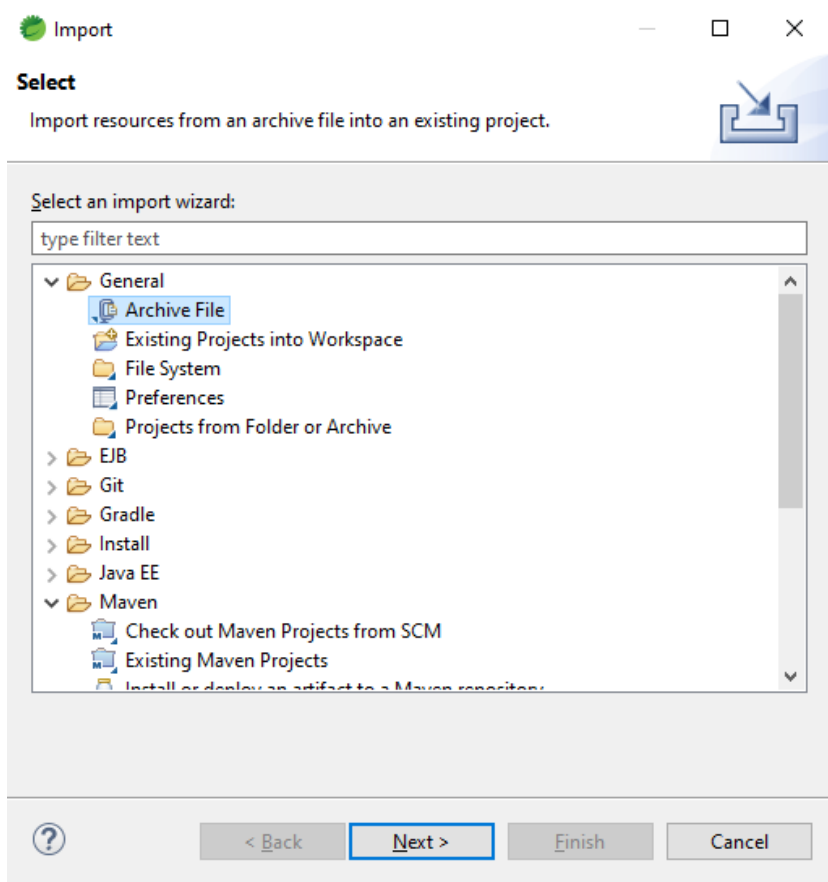
⇒ un fichier ZIP est téléchargé.

Mise en œuvre de Spring Initializr

Dans SpringToolSuite4, sélectionner `File => Import...`

⇒ la fenêtre suivante apparaît.

Mise en œuvre de Spring Initializr

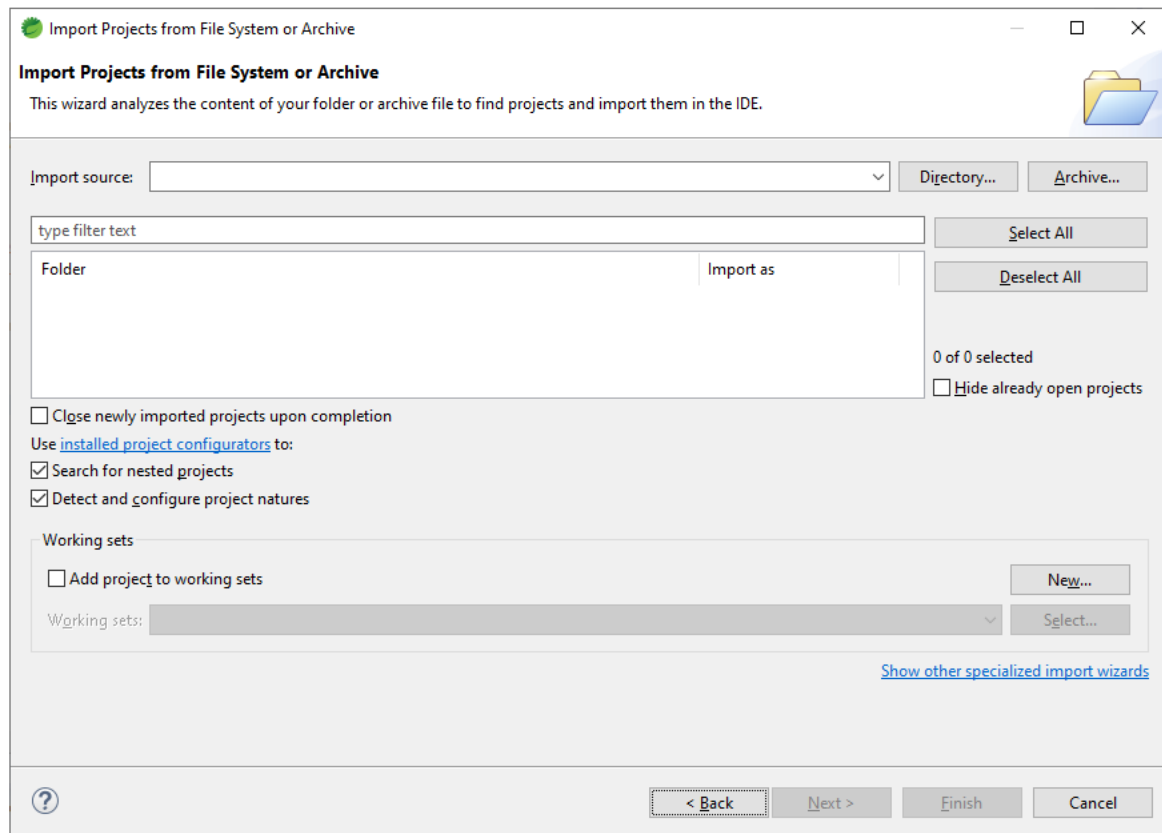


Mise en œuvre de Spring Initializr

Sélectionner `General => Projects from Folder or Archive` et cliquer sur `Next>`.

⇒ la fenêtre suivante apparaît.

Mise en œuvre de Spring Initializr



Mise en œuvre de Spring Initializr

Cliquer sur `Archive` et sélectionner le fichier ZIP téléchargé par Spring Initializr.

NB. Le fichier se trouve probablement dans le répertoire Téléchargements.

Cliquer ensuite sur `Finish` pour terminer l'import du projet.

Spring - Exercices

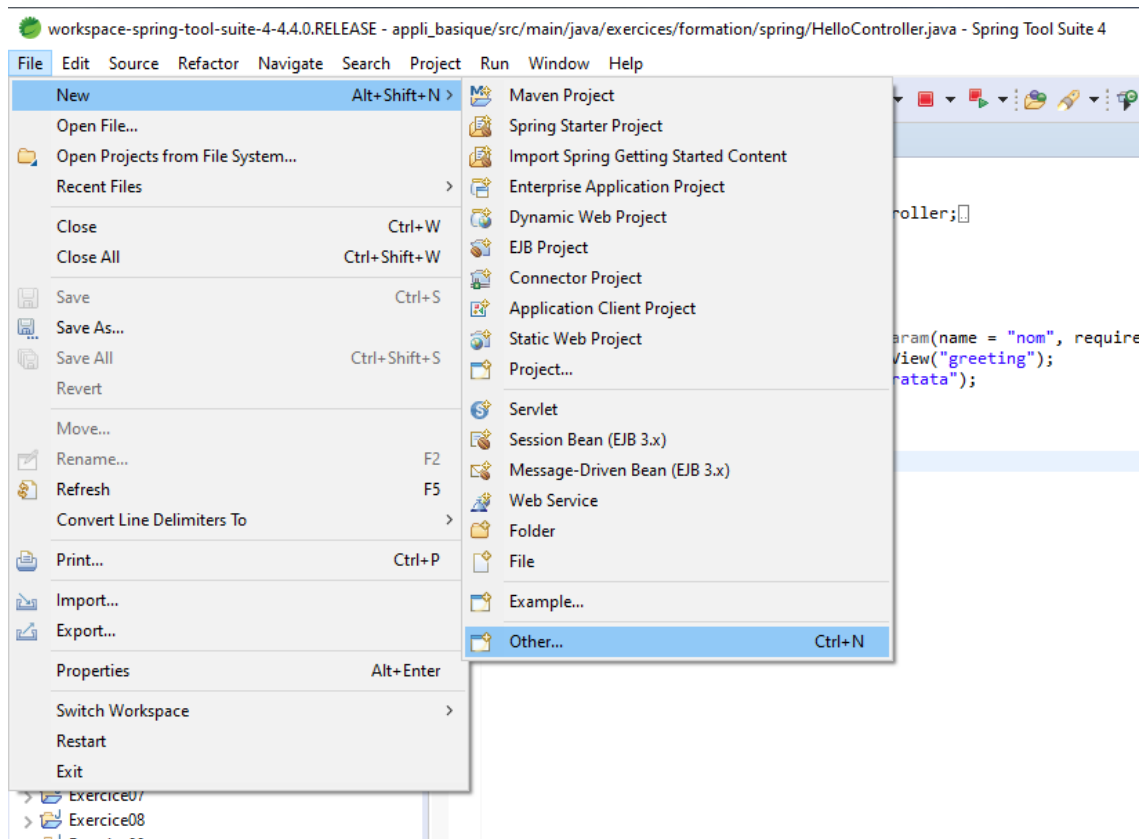
Préliminaire 2 : mise en oeuvre de Spring Initializr (bis)

Mise en œuvre de Spring Initializr

Vu que Spring Tool Suite 4 intègre Spring Initializr, une autre possibilité plus simple consiste à utiliser un de ses wizards de création de projet.

Pour cela, on sélectionne `File` \Rightarrow `New` \Rightarrow `Other...` dans Spring Tool Suite 4.

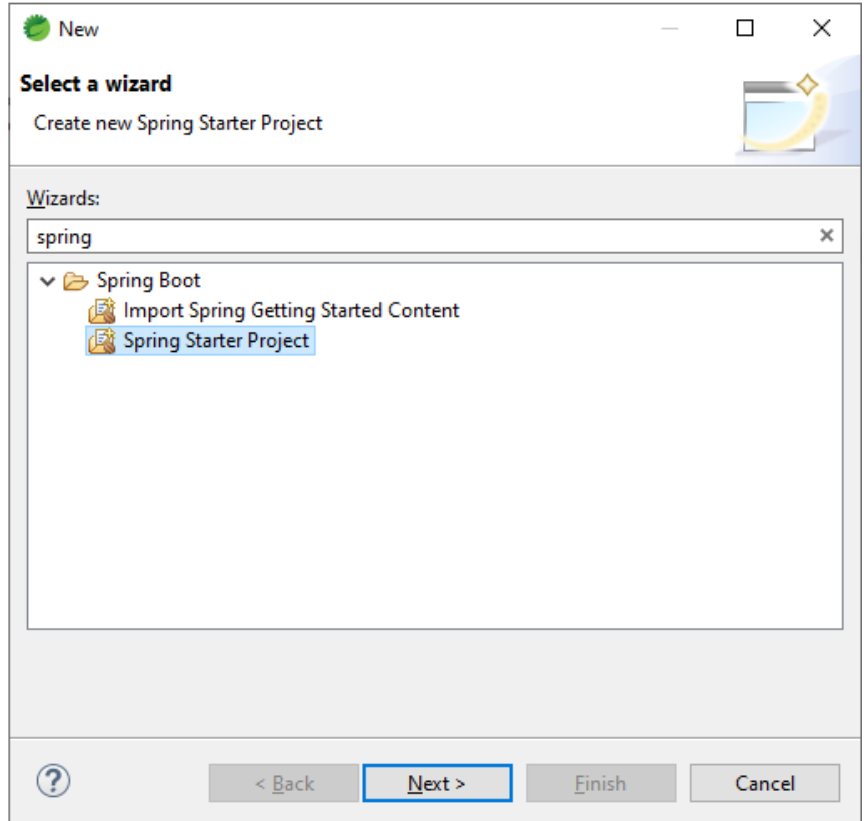
Mise en œuvre de Spring Initializr



Mise en œuvre de Spring Initializr

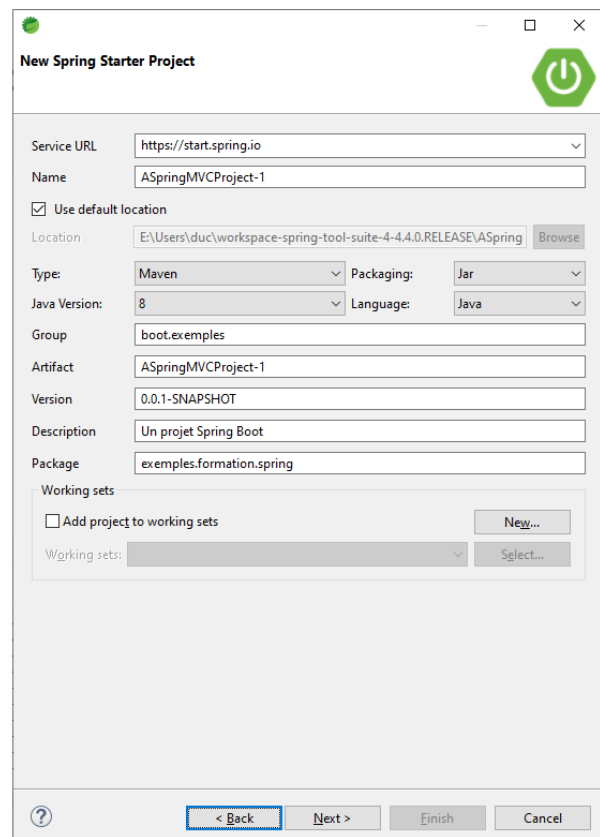
Sélection du wizard de création d'un projet Spring initial.

Il est recommandé d'utiliser le champ de recherche en entrant « Spring » comme texte à rechercher.



Mise en œuvre de Spring Initializr

Spécification des caractéristiques principales du projet : GAV (group / artifact / version) du projet Maven et nom de package.



The screenshot shows the 'New Spring Starter Project' dialog box. It contains the following fields and options:

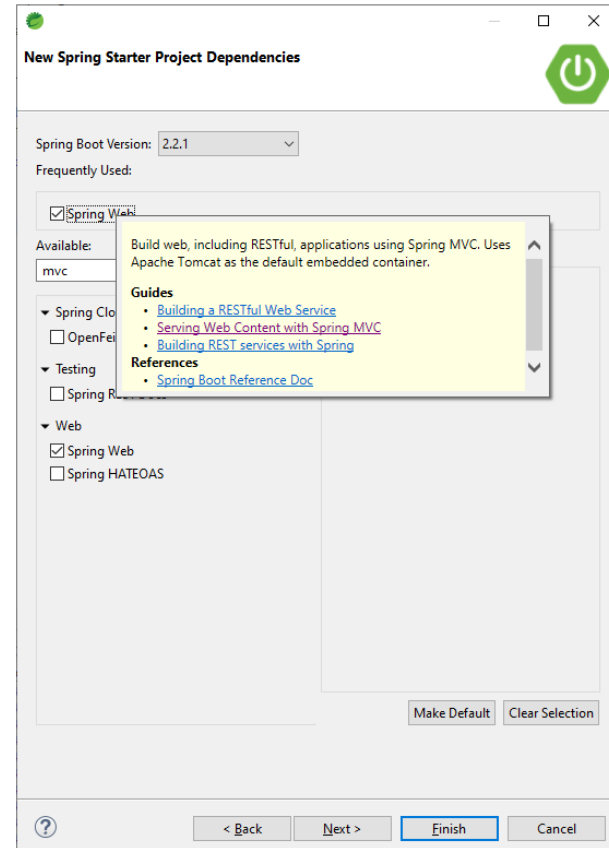
- Service URL:** `https://start.spring.io`
- Name:** `ASpringMVCProject-1`
- ☒ **Use default location**
- Location:** `E:\Users\duc\workspace-spring-tool-suite-4-4.0.RELEASE\ASpring` (with a 'Browse' button)
- Type:** `Maven` (dropdown)
- Packaging:** `Jar` (dropdown)
- Java Version:** `8` (dropdown)
- Language:** `Java` (dropdown)
- Group:** `boot.examples`
- Artifact:** `ASpringMVCProject-1`
- Version:** `0.0.1-SNAPSHOT`
- Description:** `Un projet Spring Boot`
- Package:** `exemples.formation.spring`
- Working sets:**
 - ☐ **Add project to working sets** (with a 'New...' button)
 - Working sets:** (dropdown menu) (with a 'Select...' button)

At the bottom, there are buttons: `< Back` (highlighted with a red box), `Next >`, `Finish`, and `Cancel`.

Mise en œuvre de Spring Initializr

Pour choisir Spring MVC comme dépendance, saisir « `mvc` » dans le champ de recherche des services Spring disponibles, puis cocher la case Spring Web.

Noter l'aide proposée si on approche le pointeur de la case Spring Web.



Mise en œuvre de Spring Initializr

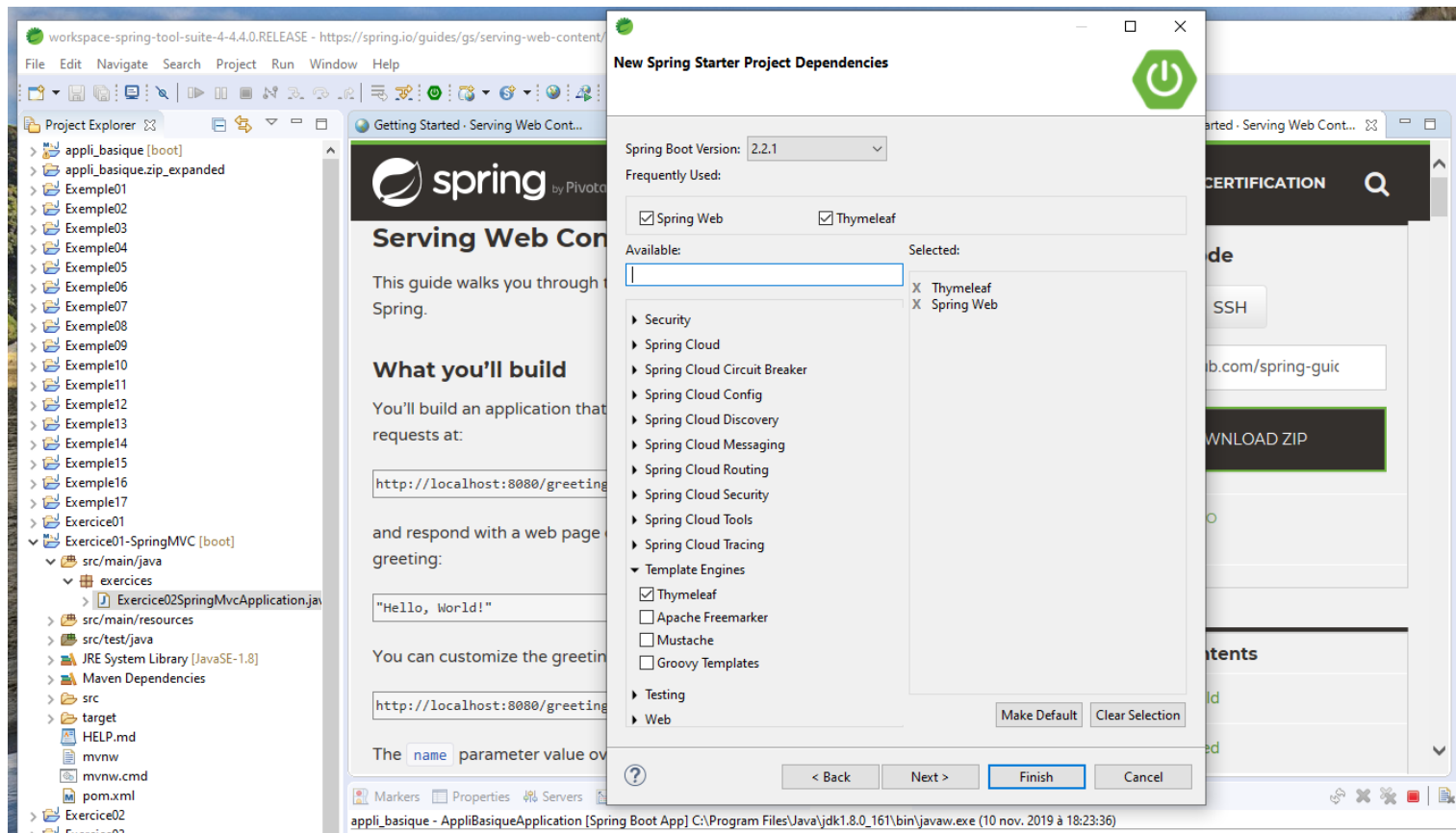
Sélectionner le lien « `Serving Web Content with Spring MVC` » et observer qu'une page d'aide apparaît dans Spring Tool Suite 4. Cette page d'aide est un court tutoriel sur la réalisation d'un projet Spring MVC basique.

Dans la fenêtre de sélection de dépendances Spring, sélectionner également `Template Engines` ⇒ `Thymeleaf`.

Cliquer enfin sur `Finish` pour terminer de créer le projet.

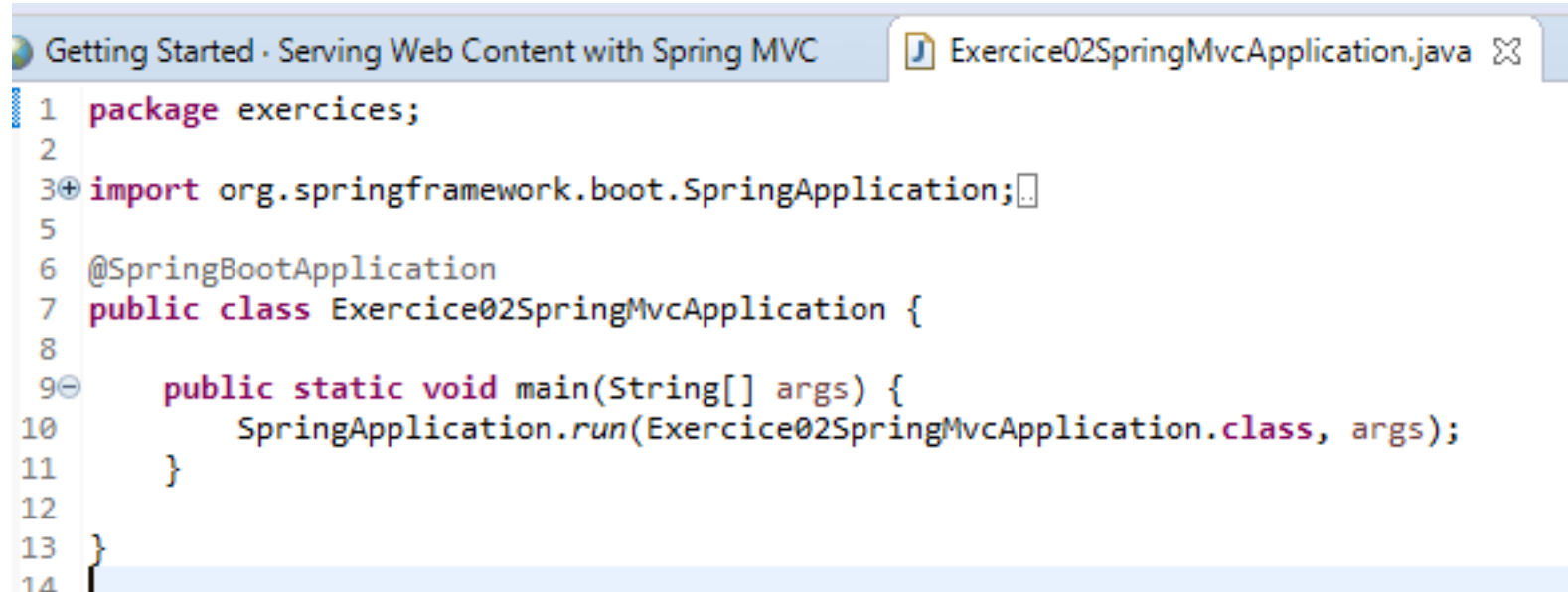
NB. La création complète du projet prend du temps car de nombreuses librairies sont récupérées par Internet.

Mise en œuvre de Spring Initializr



Mise en œuvre de Spring Initializr

Ouvrir l'unique fichier Java créé, dans `src/main/java`, et observer son contenu.

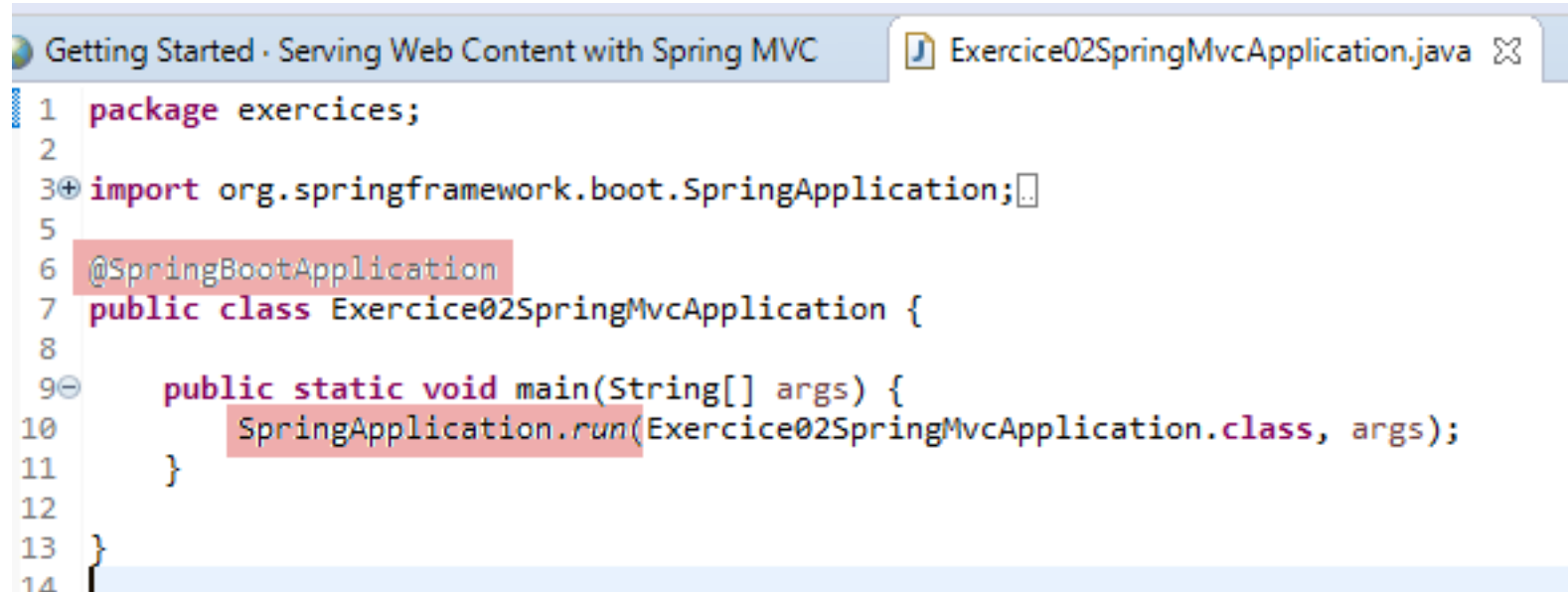


The screenshot shows an IDE window titled "Getting Started · Serving Web Content with Spring MVC". The active file is "Exercice02SpringMvcApplication.java". The code is as follows:

```
1 package exercises;
2
3 import org.springframework.boot.SpringApplication;
4
5
6 @SpringBootApplication
7 public class Exercice02SpringMvcApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(Exercice02SpringMvcApplication.class, args);
11     }
12
13 }
14
```

Mise en œuvre de Spring Initializr

Il s'agit d'une application **Spring Boot**. Il s'agit d'un outil Spring pour faciliter la définition d'un projet Spring.



```
Getting Started · Serving Web Content with Spring MVC  Exercice02SpringMvcApplication.java ✕
1 package exercices;
2
3+ import org.springframework.boot.SpringApplication;
4
5
6 @SpringBootApplication
7 public class Exercice02SpringMvcApplication {
8
9-     public static void main(String[] args) {
10         SpringApplication.run(Exercice02SpringMvcApplication.class, args);
11     }
12
13 }
14
```

Mise en œuvre de Spring Initializr

L'annotation `@SpringBootApplication` est une méta-annotation, qui regroupe plusieurs annotations Spring :

- `@SpringBootConfiguration`
- `@ComponentScan`
- `@EnableAutoConfiguration`

En synthèse, elle permet de créer une application Spring exécutable et standalone (i.e. qui n'a pas besoin d'un serveur d'application Java ni même d'un container de servlets).

Elle embarque en effet un container Tomcat.

Mise en œuvre de Spring Initializr

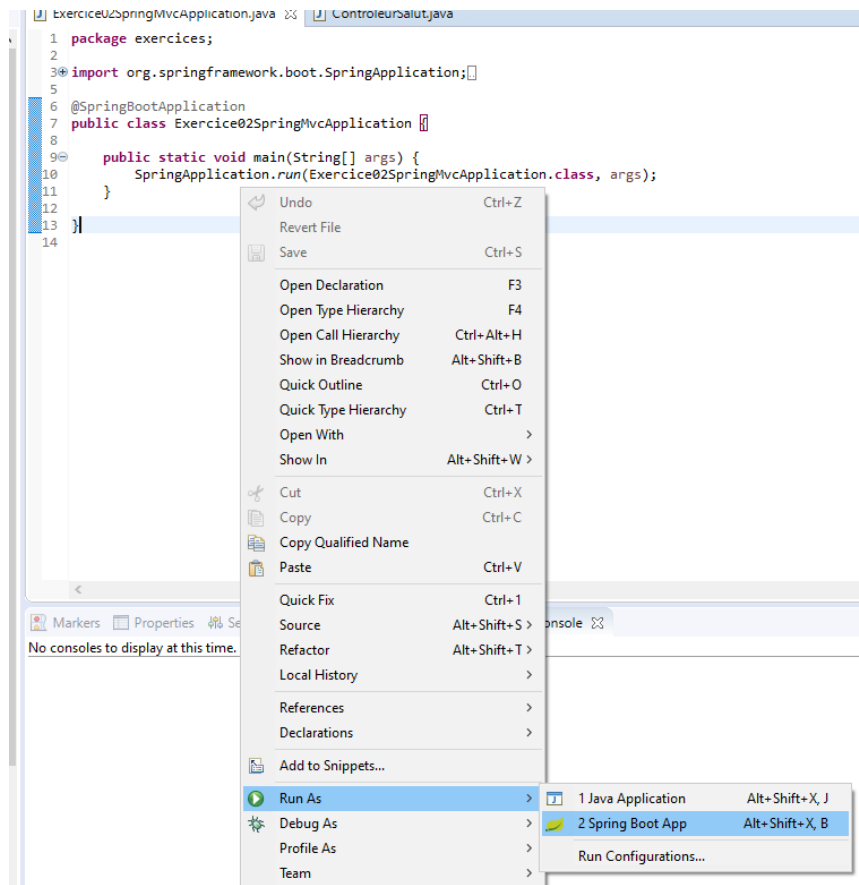
De plus, avec Spring Boot, il n'est pas nécessaire de créer un fichier de définition de l'application web (`web.xml`) ni de fichier de définition des beans du projet : tout est fait par code Java.

Mise en œuvre de Spring Initializr

L'instruction `SpringApplication.run` permet de lancer l'application.

Pour ce qui est du lancement depuis Spring Tool Suite 4, il suffit de cliquer avec le bouton droit dans le fichier Java contenant l'application Spring Boot et de sélectionner l'entrée `Run As ⇒ Spring Boot App`.

Mise en œuvre de Spring Initializr

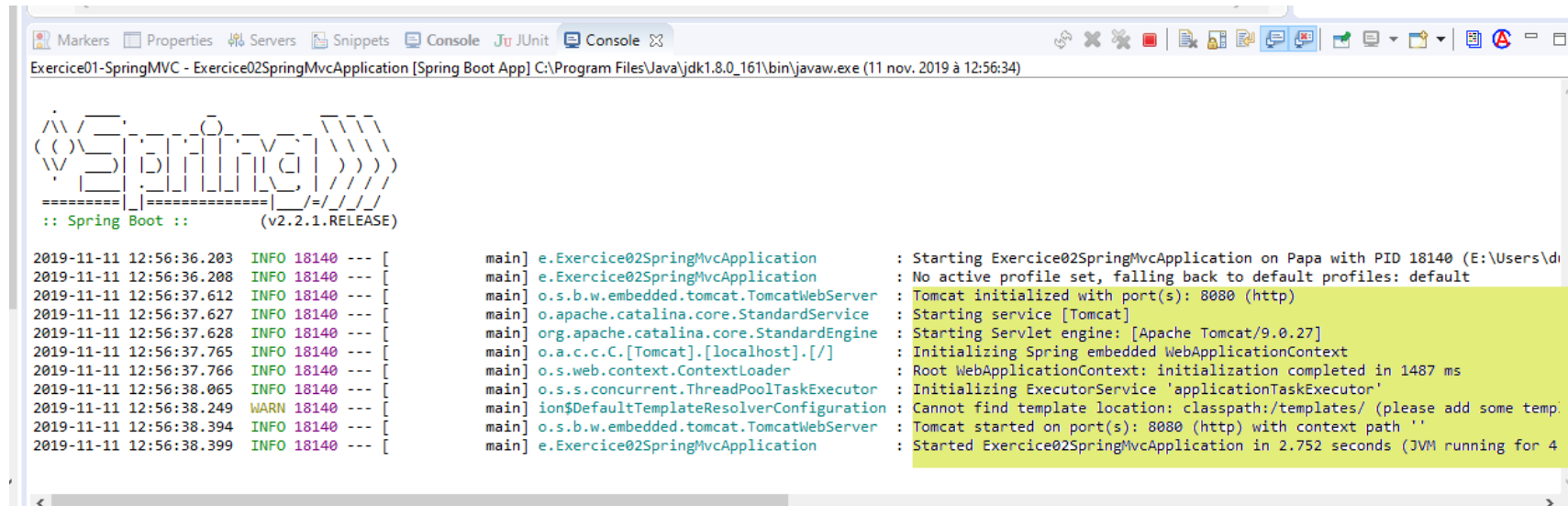


Mise en œuvre de Spring Initializr

Cela lance le container Tomcat et effectue toutes les opérations nécessaires (enregistrement des beans dans l'IOC container, création d'un contrôleur central, enregistrement comme servlet dans Tomcat, ...).

On voit dans la console STS4 que le démarrage de cette application BootStrap entraînent le lancement du container de servlets Tomcat, et que ce dernier écoute sur le port 8080.

Mise en œuvre de Spring Initializr



```
Exercise01-SpringMVC - Exercice02SpringMvcApplication [Spring Boot App] C:\Program Files\Java\jdk1.8.0_161\bin\javaw.exe (11 nov. 2019 à 12:56:34)

:: Spring Boot :: (v2.2.1.RELEASE)

2019-11-11 12:56:36.203 INFO 18140 --- [main] e.Exercice02SpringMvcApplication : Starting Exercice02SpringMvcApplication on Papa with PID 18140 (E:\Users\di
2019-11-11 12:56:36.208 INFO 18140 --- [main] e.Exercice02SpringMvcApplication : No active profile set, falling back to default profiles: default
2019-11-11 12:56:37.612 INFO 18140 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2019-11-11 12:56:37.627 INFO 18140 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2019-11-11 12:56:37.628 INFO 18140 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.27]
2019-11-11 12:56:37.765 INFO 18140 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2019-11-11 12:56:37.766 INFO 18140 --- [main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 1487 ms
2019-11-11 12:56:38.065 INFO 18140 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2019-11-11 12:56:38.249 WARN 18140 --- [main] ion$DefaultTemplateResolverConfiguration : Cannot find template location: classpath:/templates/ (please add some temp
2019-11-11 12:56:38.394 INFO 18140 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2019-11-11 12:56:38.399 INFO 18140 --- [main] e.Exercice02SpringMvcApplication : Started Exercice02SpringMvcApplication in 2.752 seconds (JVM running for 4
```

Spring - Exercices

Exercice 01 : une application Spring MVC basique

Une application Spring MVC basique

On veut réaliser une mini-application réalisant le fameux "Hello world !" mais respectant l'architecture Spring MVC.

En premier lieu, on veut juste afficher le fameux message sur invocation de l'URL

`http://localhost:8080/hello.`

La première chose à faire consiste à observer le contenu du projet et à déterminer ce qui manque pour réaliser une application fonctionnelle.

Une application Spring MVC basique

On notera que l'affichage consiste en une simple page statique (un message "Hello world !"), par conséquent on réalisera une implémentation très simple de cet exercice avec une architecture MVC.

A vous !

Spring - Exercices

Exercice 02 : une application Spring MVC dynamique

Une application Spring MVC dynamique

Cette fois, le but est qu'on puisse saluer une personne par son nom et prénom, sachant que :

- Le nom est passé sous la forme d'une partie de l'URL
- Le prénom est passé sous la forme d'un paramètre nommé `prénom` de l'URL.

Exemple :

`http://localhost:8080/hello/Duc?prénom=Patrick`

Une application Spring MVC dynamique

Bien sûr, on peut repartir du projet précédent et simplement ajouter le traitement de la nouvelle URL.

Le texte HTML doit être généré par un moteur de template, Thymeleaf ou pages JSP, au choix.

A vous !

Spring - Exercices

Exercice 03 : une application Spring MVC riche

Une application Spring MVC riche

On veut pouvoir gérer une bibliothèque numérique au moyen d'une application Spring MVC.

Il doit être possible d'effectuer les opérations suivantes sur la bibliothèque :

- Y ajouter des livres
- Lister les livres contenus
- Y rechercher un livre par son ISBN
- Y supprimer des livres.

Une application Spring MVC riche

Les informations suivantes de chaque livre doivent être gérées :

- Un nom d'auteur
- Un identifiant ISBN
- Un titre
- Une année d'édition.

Le texte HTML doit être généré par un moteur de template, Thymeleaf ou pages JSP, au choix.

A vous !

Spring - Exercices

Exercice 04 : une application Spring MVC avec gestion des erreurs

Gestion des erreurs

Reprendre l'exercice précédent (gestion d'une bibliothèque numérique au moyen d'une application Spring MVC), mais en ajoutant des pages d'erreur adaptées.

Les erreurs dans les objets métier doivent être exprimées sous forme d'exception :

- livre non trouvé
- remplacement d'un livre existant (interdit !)
- retrait d'un livre inconnu
- etc.

Gestion des erreurs

Une page d'erreur commune est demandée, affichant entre autres le code de retour HTTP émis par le serveur HTTP.

Conseil : ce serait bien de réaliser des tests unitaires (avec JUnit par exemple) avant de réaliser des tests fonctionnels (i.e. au moyen d'un navigateur)...

A vous !