

Java EE

Olivier Gutierrez

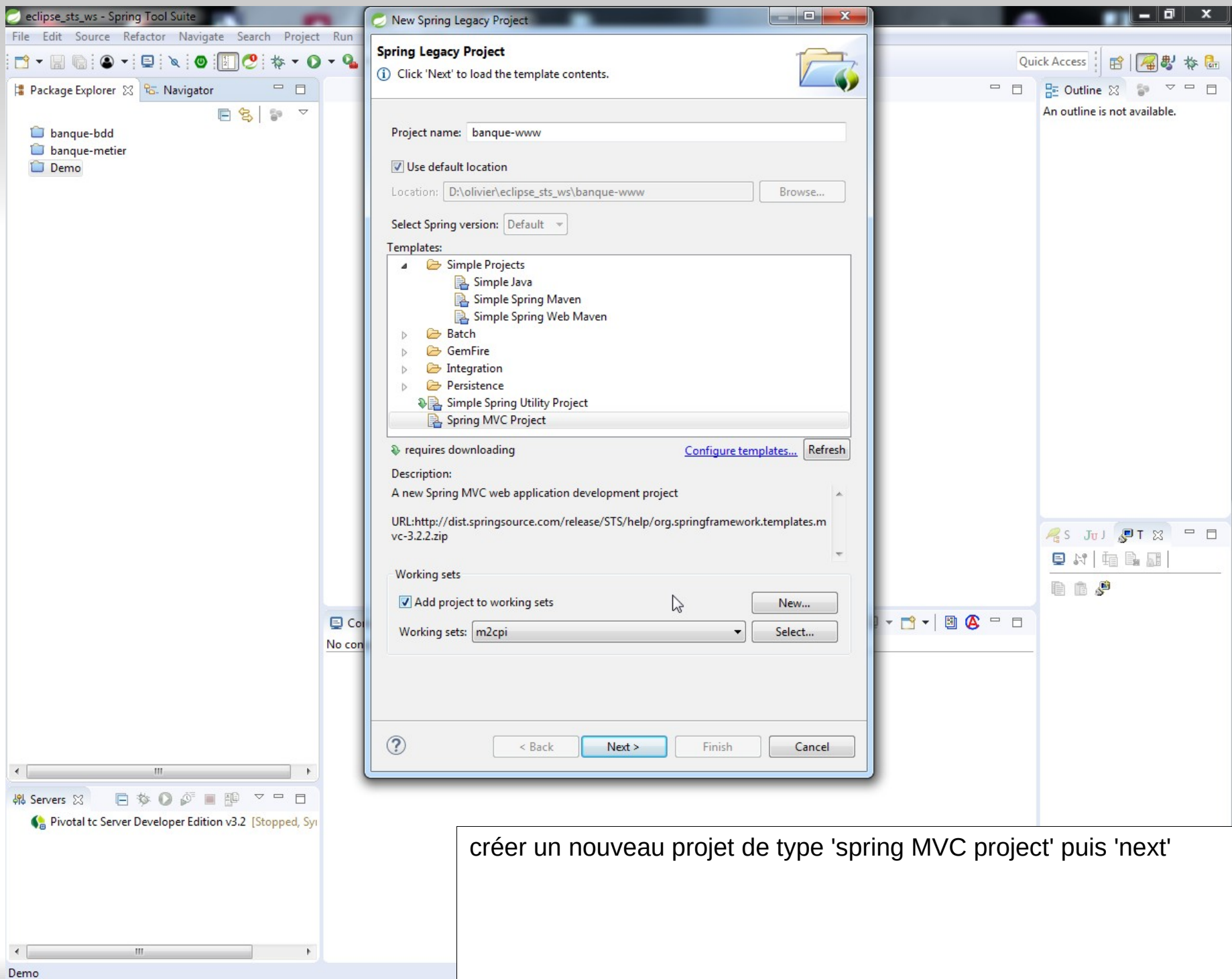
Le framework Spring

Spring MVC : premiers pas

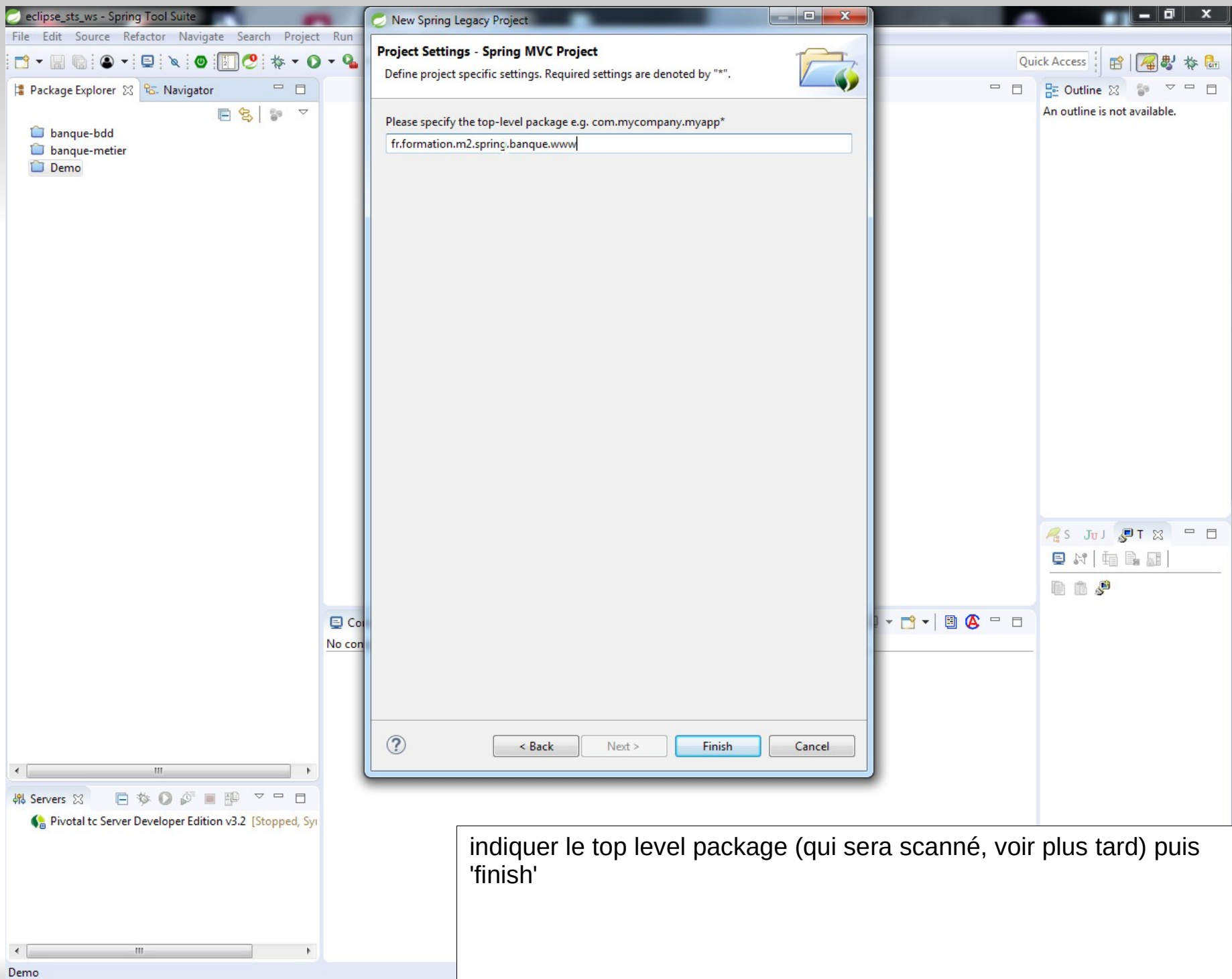


Sommaire

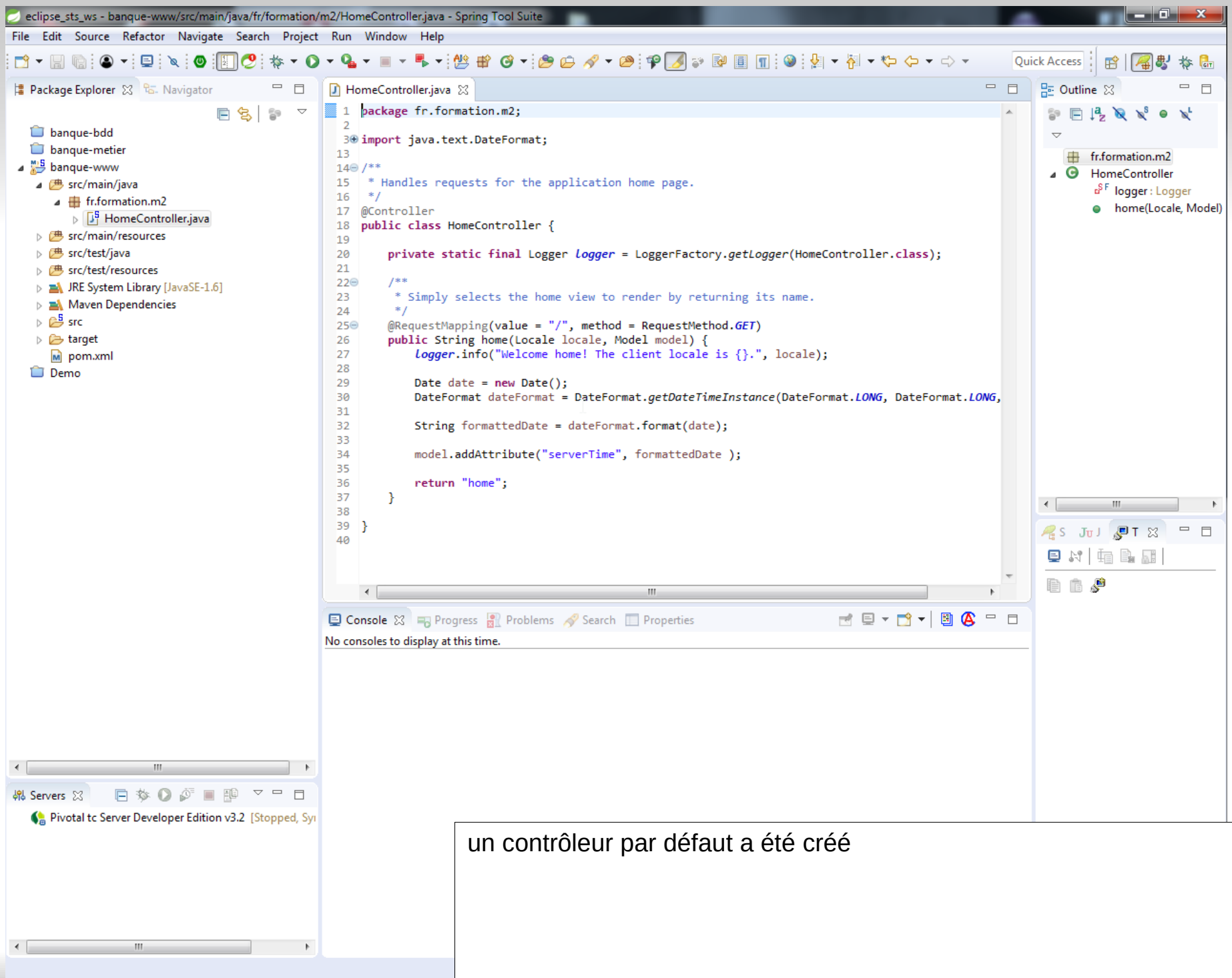
- Création d'un projet web MVC



créer un nouveau projet de type 'spring MVC project' puis 'next'



indiquer le top level package (qui sera scanné, voir plus tard) puis 'finish'



un contrôleur par défaut a été créé

The screenshot shows the Eclipse IDE interface with the following components:

- Package Explorer:** Displays the project structure. The package `fr.formation.m2` is expanded, showing the `HomeController.java` file.
- Editor:** Displays the `HomeController.java` file. The code is as follows:

```
1 package fr.formation.m2;
2
3 import java.text.DateFormat;
4
5 /**
6  * Handles requests for the application home page.
7  */
8 @Controller
9 public class HomeController {
10
11     private static final Logger logger = LoggerFactory.getLogger(HomeController.class);
12
13     /**
14      * Simply selects the home view to render by returning its name.
15      */
16     @RequestMapping(value = "/", method = RequestMethod.GET)
17     public String home(Locale locale, Model model) {
18         logger.info("Welcome home! The client locale is {}.", locale);
19
20         Date date = new Date();
21         DateFormat dateFormat = DateFormat.getDateTimeInstance(DateFormat.LONG, DateFormat.LONG,
22             locale);
23
24         String formattedDate = dateFormat.format(date);
25
26         model.addAttribute("serverTime", formattedDate );
27
28         return "home";
29     }
30 }
31
32 }
```
- Outline:** Shows the class structure. The package `fr.formation.m2` is expanded, showing the `HomeController` class. The class has a `logger: Logger` field and a `home(Locale, Model)` method.
- Console:** Displays the message "No consoles to display at this time."
- Servers:** Shows the `Pivotal tc Server Developer Edition v3.2` server, which is stopped.

l'application a déjà un minimum de code opérationnel avec
@RequestMapping "/" représentant l'url racine du projet web.
Logger.info permet des affichages dans la console
return "home" : home correspond à la jsp qui sera appelée à la
sortie de la méthode (voir plus loin) et la variable "serverTime" la date qui
sera passée à cette jsp

eclipse_sts_ws - banque-www/src/main/java/fr/formation/m2/HomeController.java - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer Navigator HomeController.java Outline

banque-bdd
banque-metier
banque-www

New
Go Into
Open in New Window
Open Type Hierarchy
Show In
Show in Local Terminal
Copy
Copy Qualified Name
Paste
Delete
Remove from Context
Build Path
Source
Refactor
Import...
Export...
Refresh
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Restore from Local History...
Java EE Tools
Maven
Team
Compare With
GitHub
Configure
Spring Tools
Properties

```
1 package fr.formation.m2;  
2  
3 import java.text.DateFormat;  
13  
14 /**  
 * requests for the application home page.  
 */  
15  
16 class HomeController {  
17  
18     static final Logger logger = LoggerFactory.getLogger(HomeController.class);  
19  
20     /**  
21      * Simply selects the home view to render by returning its name.  
22      */  
23     @RequestMapping(value = "/", method = RequestMethod.GET)  
24     String home(Locale locale, Model model) {  
25         logger.info("Welcome home! The client locale is {}.", locale);  
26  
27         Date date = new Date();  
28         DateFormat dateFormat = DateFormat.getDateTimeInstance(DateFormat.LONG, DateFormat.LONG,  
29             Locale.getDefault());  
30         String formattedDate = dateFormat.format(date);  
31         model.addAttribute("serverTime", formattedDate );  
32  
33         return "home";  
34     }  
35 }  
36
```

fr.formation.m2
HomeController
logger: Logger
home(Locale, Model)

1 Run on Server
2 Java Application
3 JUnit Test
4 Maven build
5 Maven build...
6 Maven clean
7 Maven generate-sources
8 Maven install
9 Maven test

on peut l'exécuter : run as > run on server

The screenshot shows the Eclipse IDE with the 'Run On Server' dialog box open. The dialog has the following sections:

- Select which server to use**
- How do you want to select the server?**
 - ☒ Choose an existing server
 - ☐ Manually define a new server
- Select the server that you want to use:**
 - type filter text
 - Table with columns: Server, State
 - localhost
 - Pivotal tc Server Developer Edition v3.2 (Stopped)
- Pivotal tc Server 3.0 support J2EE 1.2, 1.3, 1.4, Java EE 5, 6 and 7 Web modules.**
- ☐ Always use this server when running this project
- Buttons: ? , < Back , Next > , **Finish** (highlighted), Cancel

The background shows the Eclipse IDE with the Package Explorer on the left, the Navigator in the center, and the Outline view on the right. The Project Explorer shows the project structure: banque-bdd, banque-metier, and banque-www. The banque-www project contains src/main/java, src/main/resources, src/test/java, src/test/resources, JRE System Library [JavaSE-1.6], Maven Dependencies, src, target, pom.xml, and Demo.

on utilise le serveur embarqué (pour nos tests)

eclipse_sts_ws - http://localhost:8080/m2/ - Spring Tool Suite

File Edit Navigate Search Project Run Window Help

Package Explorer Navigator

- banque-bdd
- banque-metier
- banque-www
 - src/main/java
 - fr.formation.m2
 - HomeController.java
 - src/main/resources
 - src/test/java
 - src/test/resources
 - JRE System Library [JavaSE-1.6]
 - Maven Dependencies
 - src
 - target
 - pom.xml
 - Demo

HomeController.java Home

http://localhost:8080/m2/

Hello world!

The time on the server is 6 février 2018 18:21:04 CET.

Outline

An outline is not available.

Console Progress Problems Search Properties

Pivotal tc Server Developer Edition v3.2 [Pivotal tc Server] C:\Program Files\Java\jdk1.8.0_92\bin\javaw.exe (6 févr. 2018 à 18:20:49)

```
INFO : org.springframework.web.context.support.XmlWebApplicationContext - Refreshing Root WebApplicatio
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions fr
INFO : org.springframework.beans.factory.support.DefaultListableBeanFactory - Pre-instantiating singlet
INFO : org.springframework.web.context.ContextLoader - Root WebApplicationContext: initialization compl
INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'appServlet': initializati
INFO : org.springframework.web.context.support.XmlWebApplicationContext - Refreshing WebApplicationCont
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions fr
INFO : org.springframework.context.annotation.ClassPathBeanDefinitionScanner - JSR-250 'javax.annotatic
INFO : org.springframework.context.annotation.ClassPathBeanDefinitionScanner - JSR-330 'javax.inject.Na
INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330 'jav
INFO : org.springframework.beans.factory.support.DefaultListableBeanFactory - Pre-instantiating singlet
INFO : org.springframework
INFO : org.springframework
INFO : org.springframework
févr. 06, 2018 6:21
INFOS: Server start
INFO : fr.formation
```

Servers

- Pivotal tc Server Developer Edition v3.2 [Started, Syn
- banque-www [Started, Synchronized]

Terminé

l'application en cours d'exécution

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure, with 'servlet-context.xml' and 'root-context.xml' highlighted under the 'webapp' directory. The central editor shows the code for 'HomeController.java', which includes package declarations, imports, a logger, and a 'home' method that formats the current date. The Outline on the right shows the class structure. The bottom console area is empty, displaying 'No consoles to display at this time.'

```
1 package fr.formation.m2;
2
3 import java.text.DateFormat;
4
5 /**
6  * Handles requests for the application home page.
7  */
8 @Controller
9 public class HomeController {
10
11     private static final Logger logger = LoggerFactory.getLogger(HomeController.class);
12
13     /**
14      * Simply selects the home view to render by returning its name.
15      */
16     @RequestMapping(value = "/", method = RequestMethod.GET)
17     public String home(Locale locale, Model model) {
18         logger.info("Welcome home! The client locale is {}.", locale);
19
20         Date date = new Date();
21         DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.LONG, DateFormat.LONG,
22
23             String formattedDate = dateFormat.format(date);
24
25         model.addAttribute("serverTime", formattedDate );
26
27         return "home";
28     }
29 }
30
31 }
```

les trois fichiers de configuration principaux :

- 'servlet-context.xml'
- 'root-context.xml'
- 'web.xml'

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
5
6   <!-- The definition of the Root Spring Container shared by all Servlets and Filters -->
7   <context-param>
8     <param-name>contextConfigLocation</param-name>
9     <param-value>/WEB-INF/spring/root-context.xml</param-value>
10  </context-param>
11
12  <!-- Creates the Spring Container shared by all Servlets and Filters -->
13  <listener>
14    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
15  </listener>
16
17  <!-- Processes application requests -->
18  <servlet>
19    <servlet-name>appServlet</servlet-name>
20    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
21    <init-param>
22      <param-name>contextConfigLocation</param-name>
23      <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
24    </init-param>
25    <load-on-startup>1</load-on-startup>
26  </servlet>
27
28  <servlet-mapping>
29    <servlet-name>appServlet</servlet-name>
30    <url-pattern>/</url-pattern>
31  </servlet-mapping>
32
33 </web-app>
34
```

'web.xml' qui est le point de référence habituel d'une application Java EE avec :

- servlet-mapping qui indique la localisation du contexte de configuration
- le contextLoaderListener qui indique quand et comment charger le conteneur léger spring

<context-param> est le **contexte** racine de l'application avec :

- contextLoaderListener qui permet de charger le conteneur léger
- root-context.xml est le fichier de contexte principal de l'application

<servlet>

- DispatcherServlet est le contrôleur principal
- Servlet-context.xml est le fichier de configuration du contrôleur

The screenshot displays the Eclipse IDE interface for a Spring MVC application. The Package Explorer on the left shows the project structure 'banque-www' with packages 'src/main/java' and 'src/main/resources'. The central editor shows the XML file 'servlet-context.xml' with annotations for MVC, static resources, and view resolution. The right sidebar shows the Outline view with the 'beans:beans' element selected. The bottom status bar indicates 'No consoles to display at this time.'

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans:beans xmlns="http://www.springframework.org/schema/mvc"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:beans="http://www.springframework.org/schema/beans"
5   xmlns:context="http://www.springframework.org/schema/context"
6   xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc
7     http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/
8     http://www.springframework.org/schema/context http://www.springframework.org/schema/context"
9 >
10   <!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure -->
11
12   <!-- Enables the Spring MVC @Controller programming model -->
13   <annotation-driven />
14
15   <!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resources -->
16   <resources mapping="/resources/**" location="/resources/" />
17
18   <!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF -->
19   <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
20     <beans:property name="prefix" value="/WEB-INF/views/" />
21     <beans:property name="suffix" value=".jsp" />
22   </beans:bean>
23
24   <context:component-scan base-package="fr.formation.m2.spring.banque.www" />
25
26 </beans:beans>
29
```

'servlet-context.xml' qui est le fichier du contrôleur de l'application mvc avec :

- <annotation-driven pour la définition des contrôleurs
- <context:component-scan indiquant la localisation du package de base des éléments à scanner
- <beans:bean contenant la spécification des vues pour l'InternalResourceViewResolver

The screenshot shows the Eclipse IDE interface with the following components:

- Package Explorer:** Displays the project structure for 'banque-www'. It includes packages like 'src/main/java' (containing 'HomeController.java' and 'HomeController'), 'src/main/resources' (containing 'META-INF' and 'log4j.xml'), 'src/test/java' (containing 'fr.formation.m2' and 'log4j.xml'), and 'src/test/resources' (containing 'log4j.xml'). It also shows the 'JRE System Library [JavaSE-1.6]', 'Maven Dependencies', and 'src' directory (containing 'main' and 'test' subdirectories).
- Editor:** The 'root-context.xml' file is open, showing the following XML content:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans-3.0.xsd">
    <!-- Root Context: defines shared resources visible to all other web components -->
</beans>
```
- Outline:** Shows the 'beans' element with the namespace 'http://www.springframework.org/schema/beans'.
- Console:** Displays the message 'No consoles to display at this time.'
- Servers:** Shows the 'Pivotal tc Server Developer Edition v3.2' and 'banque-www' (Stopped, Republish) server.

'root-context.xml' est le **contexte** racine de l'application mvc qui permet de définir les beans métiers qui seraient utilisés par cette application

