

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN KHOA CÔNG NGHỆ THÔNG TIN

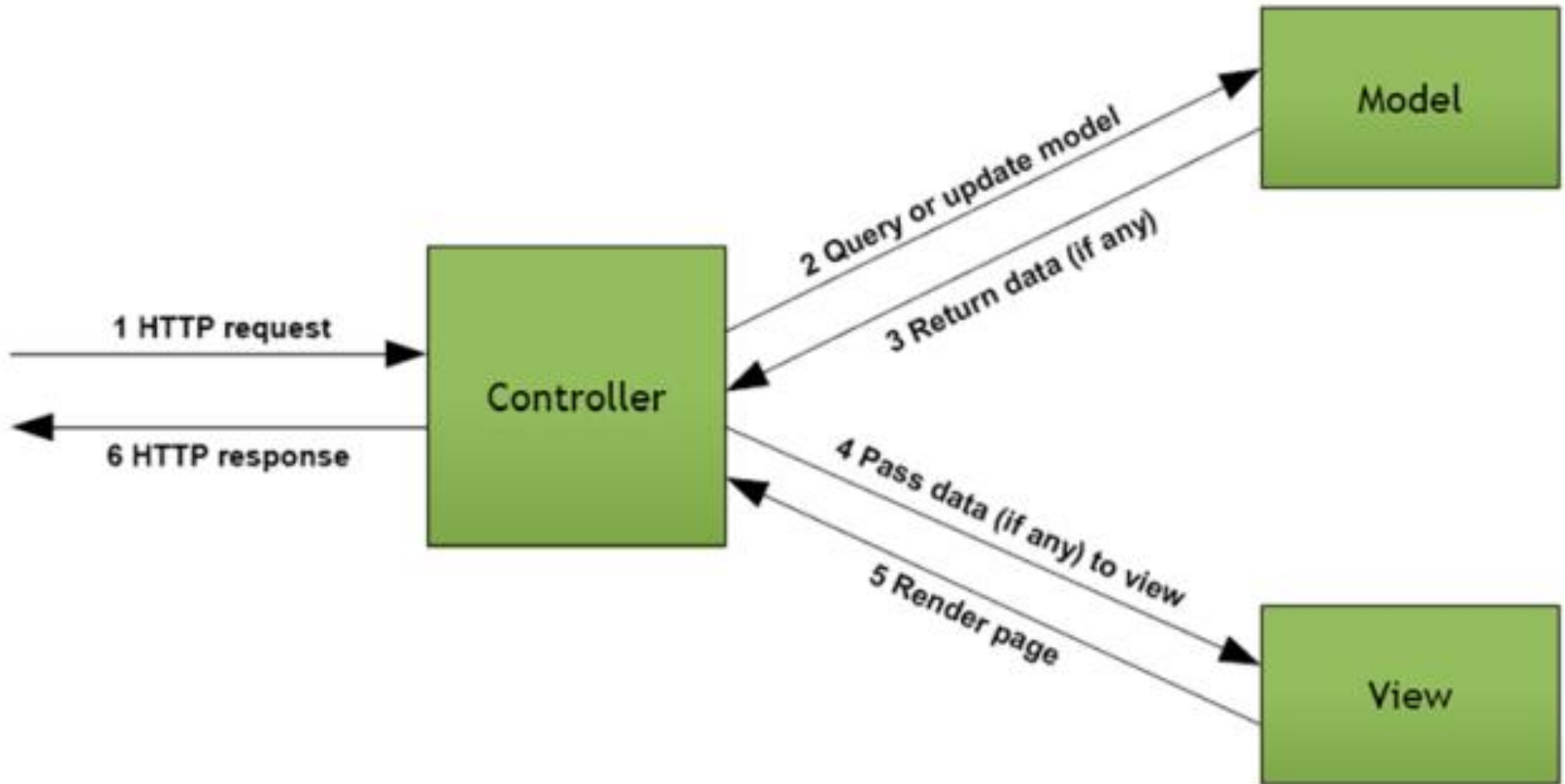
SPRING WEB MVC

Nguyễn Hoàng Anh
nhanh@fit.hcmus.edu.vn

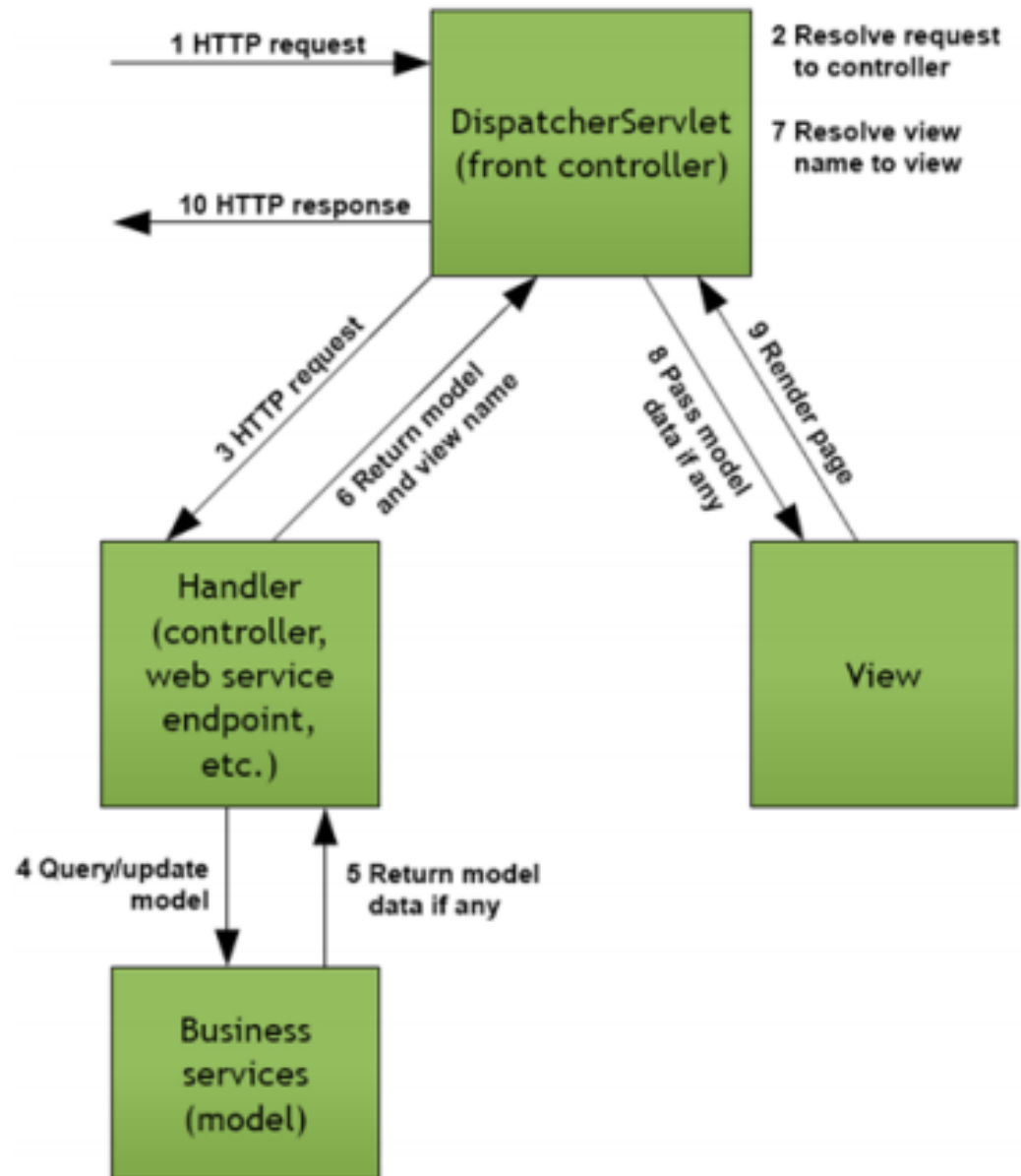
Nội dung

- **1- Spring Web MVC**
- **2- Spring Web Application 1**
- **3- Spring Web Application 2**
- **4- web.xml**
- **5- Application Context**

1- Spring Web MVC



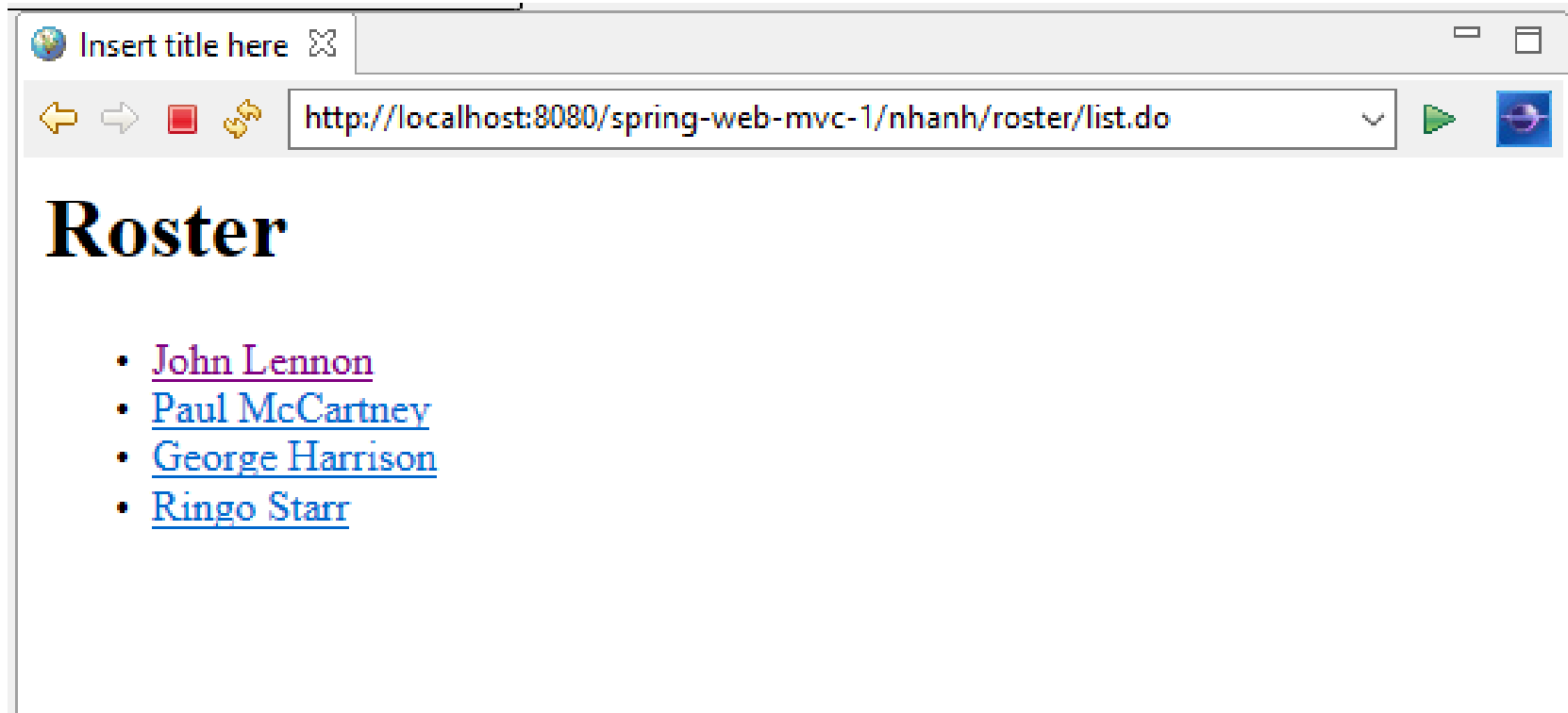
1- Spring Web MVC



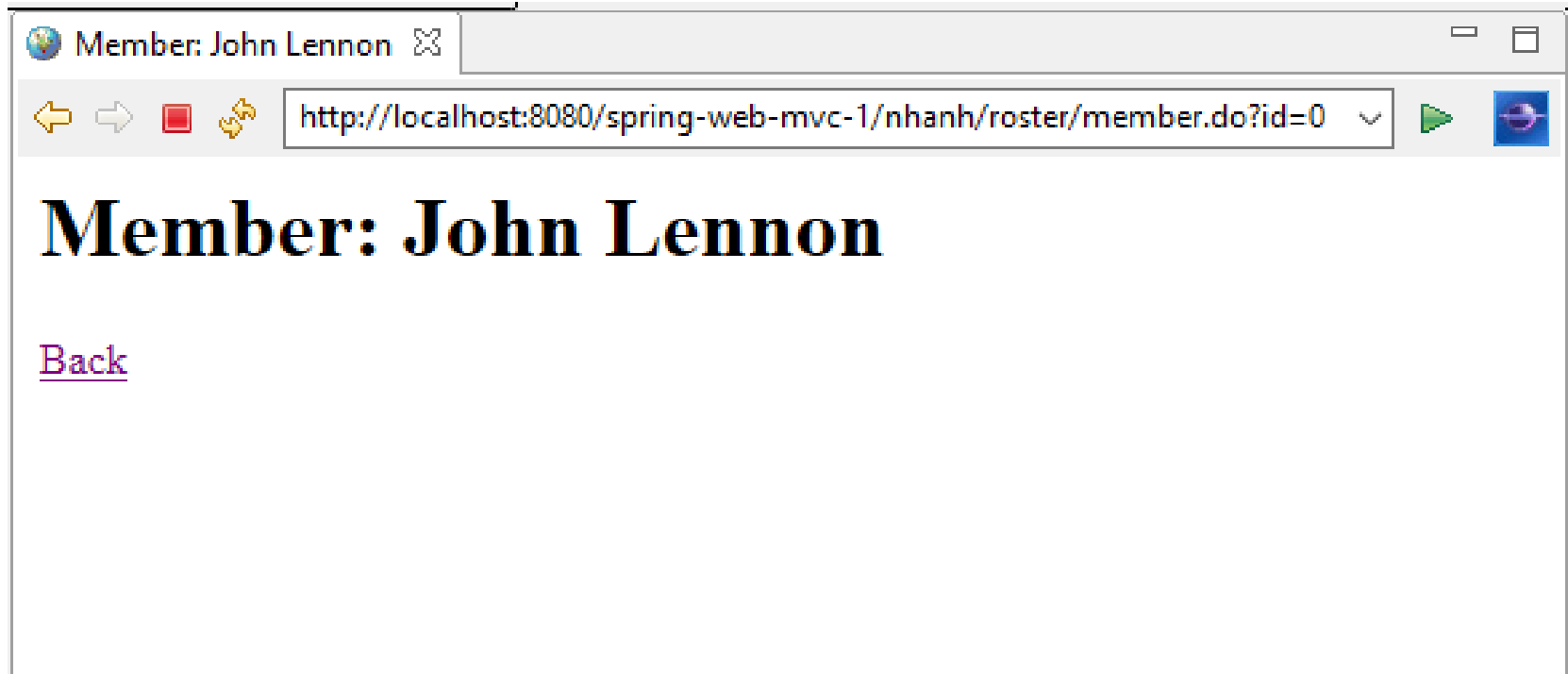
2- Spring Web Application 1

- Xây dựng chương trình đơn giản sau
 - Hiển thị danh sách người dùng
 - Hiển thị thông tin chi tiết người dùng

2- Spring Web Application 1



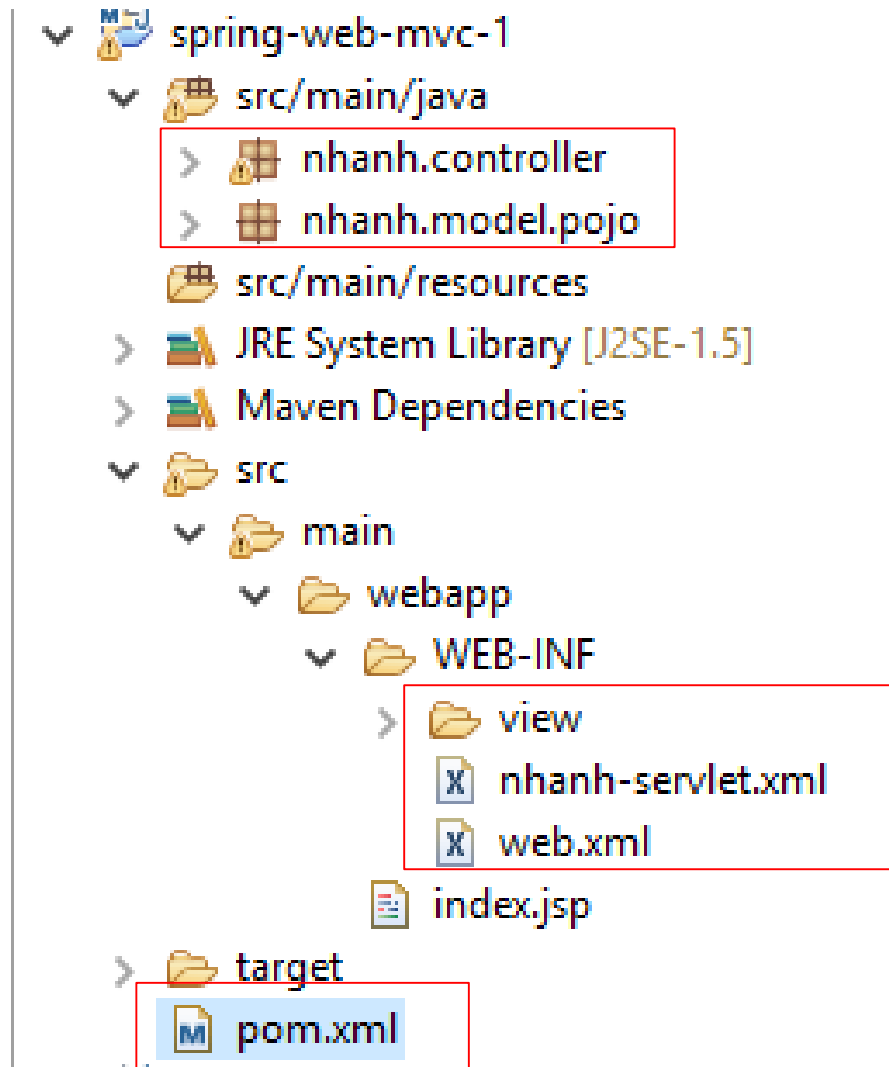
2- Spring Web Application 1



2- Spring Web Application 1

- Bước 1: Cấu hình **pom.xml**
- Bước 2: Xây dựng **Member POJO**
- Bước 3: Xây dựng **RosterController**
- Bước 4: Xây dựng **view list.jsp, member.jsp**
- Bước 5: Xây dựng **nhanh-servlet.xml**
- Bước 6: Cấu hình **web.xml**
- Bước 7: Thử nghiệm

2- Spring Web Application 1



2- Bước 1: pom.xml

```
1 <dependency>
2   <groupId>org.springframework</groupId>
3   <artifactId>spring-context</artifactId>
4   <version>4.1.6.RELEASE</version>
5 </dependency>
6 <dependency>
7   <groupId>org.springframework</groupId>
8   <artifactId>spring-webmvc</artifactId>
9   <version>4.1.6.RELEASE</version>
10 </dependency>
11 <dependency>
12   <groupId>jstl</groupId>
13   <artifactId>jstl</artifactId>
14   <version>1.2</version>
15 </dependency>
```

2- Bước 2: Member.java

```
1 package nhanh.model.pojo;
2 public class Member {
3     private String firstName;
4     private String lastName;
5     public Member() {...}
6     public Member(String firstName, String lastName) {...}
7     public String getFirstName() {...}
8     public void setFirstName(String firstName) {...}
9     public String getLastName() {...}
10    public void setLastName(String lastName) {...}
11    public String toString() {...}
12 }
```

2- Bước 3: RosterController.java

```
1  @Controller
2  public final class RosterController {
3
4      private List<Member> members = new ArrayList<>();
5
6      public RosterController() {
7          members.add(new Member("John", "Lennon"));
8          members.add(new Member("Paul", "McCartney"));
9          members.add(new Member("George", "Harrison"));
10         members.add(new Member("Ringo", "Starr"));
11     }
12     ...
13 }
```

2- Bước 3: RosterController.java

```
1  @Controller
2  public final class RosterController {
3      @RequestMapping
4      public void list(Model model) {
5          model.addAttribute(members);
6      }
7
8      @RequestMapping
9      public void member(@RequestParam("id") Integer id,
10                          Model model) {
11          model.addAttribute(members.get(id));
12      }
13 }
```

model.addAttribute(name, value)

2- Bước 4: list.jsp

```
1 <body>
2   <h1>Roster</h1>
3   <ul>
4     <c:forEach var="member" items="\${memberList} "
5               varStatus="status">
6       <li>
7         <a href="member.do?id=${status.index} ">
8           <c:out value="\${member} "></c:out>
9         </a>
10      </li>
11    </c:forEach>
12  </ul>
13 </body>
```

2- Bước 4: member.jsp

```
1 <body>
2     <h1>Member: ${member}</h1>
3     <p>
4         <a href="list.do">Back</a>
5     </p>
6 </body>
```

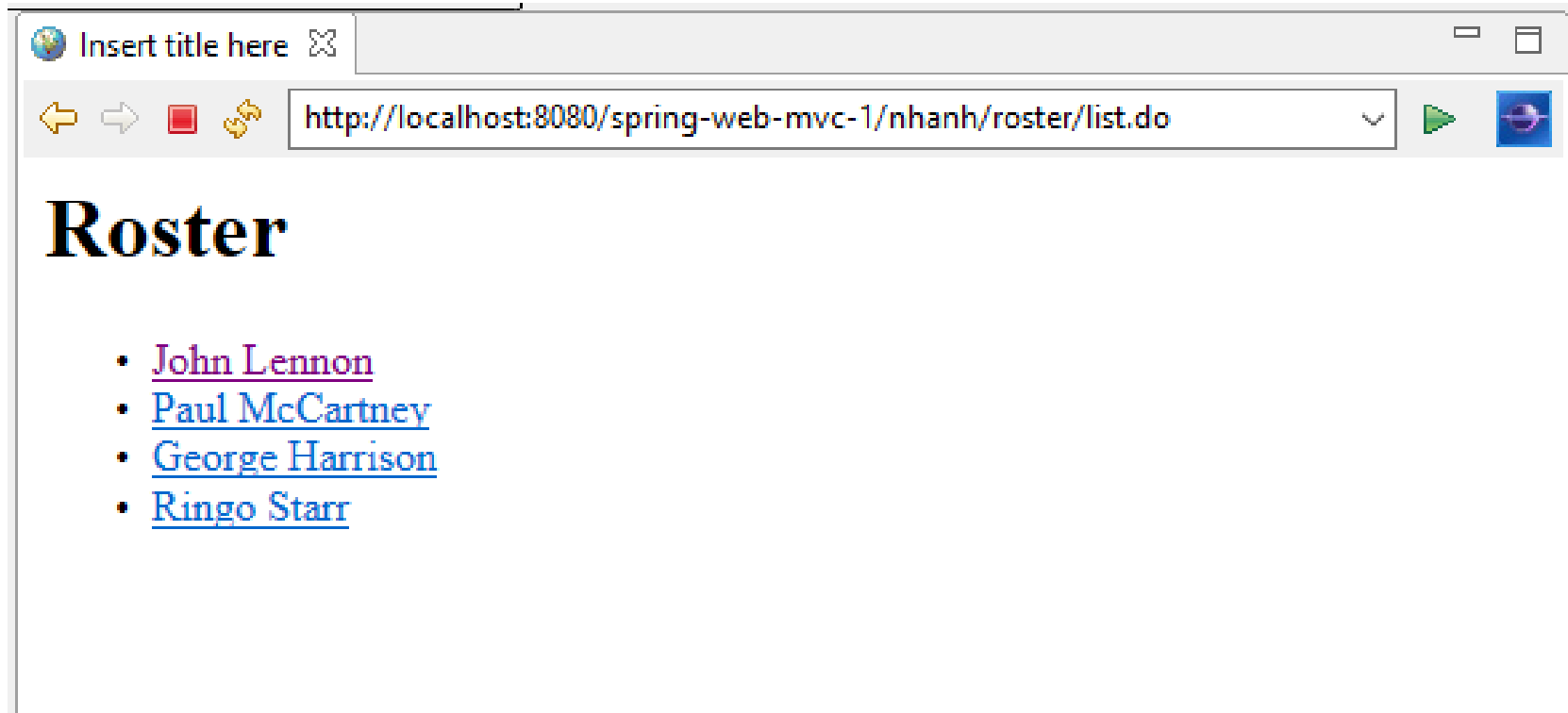
2- Bước 5: nhanh-servlet.xml

```
1 <beans ...>
2
3     <bean name="/roster/*"
4         class="nhanh.controller.RosterController"/>
5
6     <bean class="org.springframework.web.servlet.view.
7         InternalResourceViewResolver"
8         p:prefix="/WEB-INF/view/"
9         p:suffix=".jsp"/>
10 </beans>
```

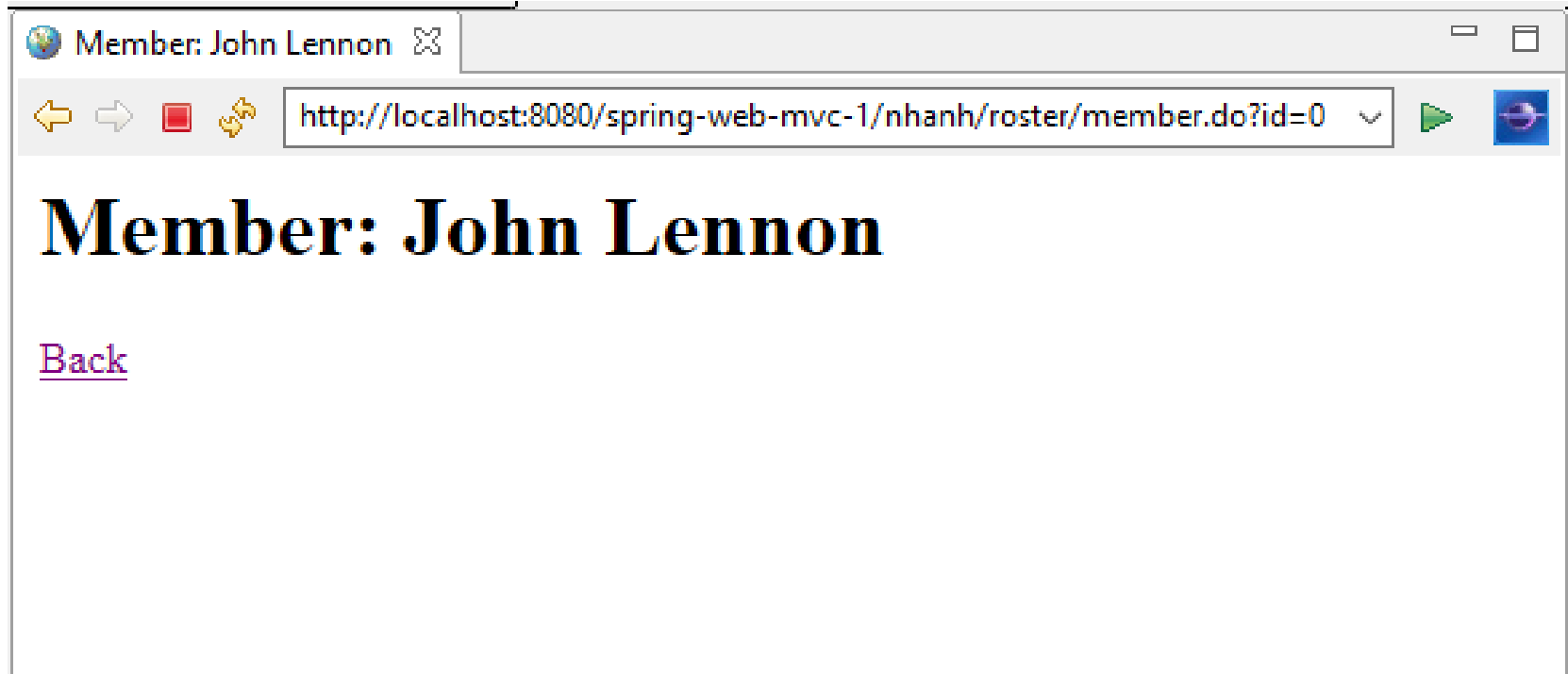

2- Bước 6: web.xml

```
1 <web-app>
2   <servlet>
3     <servlet-name>nhanh</servlet-name>
4     <servlet-class>
5       org.springframework.web.servlet.DispatcherServlet
6     </servlet-class>
7   </servlet>
8   <servlet-mapping>
9     <servlet-name>nhanh</servlet-name>
10    <url-pattern>/nhanh/*</url-pattern>
11  </servlet-mapping>
12 </web-app>
```

2- Bước 6: thử nghiệm



2- Bước 6: thử nghiệm



3- Spring Web Application 2

- Viết chương trình cho phép thêm một thành viên



A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/spring-web-mvc-1/nhanh/nominee/add`. The page title is "Nominate a member for the award". Below the title, there are two input fields: "First name:" and "Last name:". Below these fields is a "Submit" button.



A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/spring-web-mvc-1/nhanh/nominee/add`. The page title is "Thanks". Below the title, there is a message: "Thanks for nominating Mike David."

3- Spring Web Application 2

- Bước 1: Xây dựng **NomineeController**
- Bước 2: Xây dựng **form.jsp**, **thanks.jsp**
- Bước 3: Điều chỉnh **nhanh-servlet.xml**
- Bước 4: **Thử nghiệm**

3- Bước 1: NomineeController.java

```
1  @Controller
2  public final class NomineeController {
3
4      @RequestMapping(value="/add",
5                      method = RequestMethod.GET)
6      public String init(Model model) {
7          model.addAttribute("member", new Member());
8          return "/nominee/form";
9      }
10     @RequestMapping(method = RequestMethod.POST)
11     public String add(@ModelAttribute("member")
12                     Member member, Model model) {
13         Map map = model.asMap();
14         System.out.println(map.get("member"));
15         System.out.println(member);
16         return "nominee/thanks";
17     }
18 }
```

3- Bước 2: form.jsp

```
1 <%@ taglib prefix="form"
2     uri="http://www.springframework.org/tags/form"%>
3 <html>
4 <body>
5     <h1>Nominate a member for the award</h1>
6     <form:form modelAttribute="member">
7         <div>
8             First name: <form:input path="firstName" />
9         </div>
10        <div>
11            Last name: <form:input path="lastName" />
12        </div>
13        <div>
14            <input type="submit" value="Submit"></input>
15        </div>
16    </form:form>
17 </body>
18 </html>
```

3- Bước 2: thanks.jsp

```
1 <html>
2 <head>
3 <title>Thanks</title>
4 </head>
5 <body>
6     <h1>Thanks</h1>
7     <p>Thanks for nominating ${member}.</p>
8 </body>
9 </html>
```


3- Bước 3: nhanh-servlet.xml

```
1 <beans>
2   <bean name="/nominee/*"
3       class="nhanh.controller.NomineeController"/>
4
5   <bean class="org.springframework.web.servlet.
6       view.InternalResourceViewResolver"
7       p:prefix="/WEB-INF/view/"
8       p:suffix=".jsp" />
9 </beans>
```

3- Bước 4: Thử nghiệm



A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/spring-web-mvc-1/nhanh/nominee/add`. The page title is "Nominate a member for the award". Below the title, there are two input fields: "First name:" with the value "Mike" and "Last name:" with the value "David". Below these fields is a "Submit" button.



A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/spring-web-mvc-1/nhanh/nominee/add`. The page title is "Thanks". Below the title, there is a message: "Thanks for nominating Mike David."

4- web.xml

```
1 <web-app>
2   <listener>
3     <listener-class>
4     org.springframework.web.context.ContextLoaderListener
5     </listener-class>
6   </listener>
7   <servlet>
8     <servlet-name>main</servlet-name>
9     <servlet-class>
10    org.springframework.web.servlet.DispatcherServlet
11    </servlet-class>
12  </servlet>
13  <servlet-mapping>
14    <servlet-name>main</servlet-name>
15    <url-pattern>/main/*</url-pattern>
16  </servlet-mapping>
17 </web-app>
```

WEB/INF/**main**-servlet.xml

4- web.xml

```
1 <servlet>
2   <servlet-name>dispatcher</servlet-name>
3   <servlet-class>
4       org.springframework.web.servlet.DispatcherServlet
5   </servlet-class>
6   <init-param>
7       <param-name>contextConfigLocation</param-name>
8       <param-value>
9           /WEB-INF/conf/main-servlet.xml
10      </param-value>
11   </init-param>
12 </servlet>
```

4- web.xml

```
1 <servlet>
2   <servlet-name>nhanh</servlet-name>
3   <servlet-class>
4     org.springframework.web.servlet.DispatcherServlet
5   </servlet-class>
6 </servlet>
7 <servlet>
8   <servlet-name>main</servlet-name>
9   <servlet-class>
10    org.springframework.web.servlet.DispatcherServlet
11  </servlet-class>
12 </servlet> <servlet-mapping>
13   <servlet-name>nhanh</servlet-name>
14   <url-pattern>/nhanh/*</url-pattern>
15 </servlet-mapping> <servlet-mapping>
16   <servlet-name>main</servlet-name>
17   <url-pattern>/main/*</url-pattern>
18 </servlet-mapping>
```

5- Application Context

- HandlerMappings
- HandlerAdapters
- ViewResolvers

5- HandlerMappings

- Khi DispatcherServlet nhận một request, nó cần tìm một controller để xử lý request.
- DispatcherServlet giữ một danh sách theo thứ tự HandlerMapping (1-n).
- Nếu tìm thấy handler cho request, DispatcherServlet gửi request đến handler đó, ngược lại nó phát sinh lỗi HTTP 404.

5- HandlerMappings

■ BeanNameUrlHandlerMapping

- Mapping request url với một handler bean name.
- Có thể sử dụng wildcard.

```
1 <beans ... >
2   <bean class="org.springframework.web.servlet.handler.
3       BeanNameUrlHandlerMapping"/>
4   <bean name="/roster/*"
5       class="nhanh.controller.RosterController" />
6   <bean name="/nominee/*"
7       class="nhanh.controller.NomineeController"
8       p:thanksViewName="redirect:/nominee/thanks" />
9   <bean name="/hello*.htm"
10      class="com.nhanh.HelloWorldController" />
11 </beans>
```


5- HandlerMappings

- **ControllerClassNameHandlerMapping**

- Sử dụng controller name để xử lý request url

```
1 <bean
2     class="org.springframework.web.servlet.mvc.support.
3         ControllerClassNameHandlerMapping" />
4
5 <bean class="nhanh.controller.RosterController" />
6
7 <bean class="nhanh.controller.NomineeController" />
```

5- HandlerMappings

- **DefaultAnnotationHandlerMapping**
 - @Controller, @RequestMapping (xem thêm)

```
1 @Controller
2 @RequestMapping("/home")
3 public class HomeController {
4
5 }
```

```
1 @RequestMapping(value={"/method1", "/method1/second"})
2 public void method1() {
3     ...
4 }
5
```

5- HandlerMappings

- **DefaultAnnotationHandlerMapping**
 - @Controller, @RequestMapping (xem thêm)

```
1  @RequestMapping(value="/method2",
2  method=RequestMethod.POST)
3  public void method2 () {
4      ...
5  }
6
7  @RequestMapping(value="/method3",
8                  method={RequestMethod.POST,
9                          RequestMethod.GET})
10 public void method3 () {
11     ...
12 }
```

5- HandlerMappings

- **DefaultAnnotationHandlerMapping**
 - @Controller, @RequestMapping (xem thêm)

```
1  @RequestMapping(value="/method4/{id}")
2  public void method4 (@PathVariable("id") int id) {
3      ...
4  }
5
6  @RequestMapping(value="/method5/{id:[\\d]+}/{name}")
7  public void method5 (@PathVariable("id") long id,
8                      @PathVariable("name") String name) {
9      ...
10 }
11
12
```

5- HandlerMappings

- **DefaultAnnotationUrlHandlerMapping**
 - @Controller, @RequestMapping (xem thêm)

```
1 @RequestMapping(value="/method9")
2 public void method5 (@RequestParam("id") int id) {
3     ...
4 }
```

5- HandlerMappings

- **SimpleUrlHandlerMapping**
 - Mapping nhiều patterns với một bean (controller)

```
1 <bean class="org.springframework.web.servlet.handler.  
2     SimpleUrlHandlerMapping">  
3     <property name="mappings">  
4         <props>  
5             <prop key="/index.html">coreController</prop>  
6             <prop key="/about.html">coreController</prop>  
7             <prop key="/contact/*">contactController</prop>  
8             <prop key="/forums/*">forumsController</prop>  
9         </props>  
10    </property>  
11 </bean>
```

5- HandlerMappings

■ Multiple HandlerMappings

```
1 <bean class="org.springframework.web.servlet.handler.  
2     SimpleUrlHandlerMapping"  
3     p:order="0">  
4     <property name="mappings">  
5         <props>  
6             <prop key="/index.html">coreController</prop>  
7             <prop key="/about.html">coreController</prop>  
8             <prop key="/contact/*">contactController</prop>  
9             <prop key="/forums/*">forumsController</prop>  
10        </props>  
11    </property>  
12 </bean>  
13 <bean class="org.springframework.web.servlet.mvc.  
14     support.ControllerClassNameHandlerMapping"  
15     p:order="1"/>
```

5- HandlerAdapter

- Developer có thể cài đặt các handler bằng cách implement lại các controller được định nghĩa sẵn hoặc sử dụng các POJO + @Controller
- DispatcherServlet có thể gọi được các handler tương ứng bằng cách sử dụng các HandlerAdapter.
- Mỗi HandlerAdapter cài đặt cách tương tác chi tiết với mỗi handler.

5- ViewResolvers

■ InternalResourceViewResolver

```
1 <bean class="org.springframework.web.servlet.  
2         view.InternalResourceViewResolver">  
3     <property name="viewClass"  
4         value="org.springframework.web.servlet.  
5             view.JstlView" />  
6     <property name="prefix" value="/WEB-INF/view/" />  
7     <property name="suffix" value=".jsp" />  
8 </bean>
```

1	Logical view name	Actual view file
2	home	/WEB-INF/view/home.jsp
3	roster/list	/WEB-INF/view/roster/list.jsp
4	service/credit	/WEB-INF/view/service/credit.jsp

5- ViewResolvers

- InternalResourceViewResolver – Best practice

```
1 <bean class="org.springframework.web.servlet.  
2     view.InternalResourceViewResolver">  
3     <property name="prefix" value="/WEB-INF/view/" />  
4     <property name="suffix" value=".jsp" />  
5 </bean>
```

```
1 <bean class="org.springframework.web.servlet.  
2     view.InternalResourceViewResolver">  
3     <property name="prefix" value="/view/" />  
4     <property name="suffix" value=".jsp" />  
5 </bean>
```

5- ViewResolvers

- **BeanNameViewResolver**

- Bước 1: Tạo **YourCustomViewer**

- Bước 2: Controller

```
ModelAndView mav=new ModelAndView();  
  
mav.setViewName("customViewer");  
  
return mav;
```

- Bước 3: Context

```
<bean class="org.springframework.web.  
        servlet.view.BeanNameViewResolver" />  
  
<bean id="customViewer"  
        class="nhanh.viewresolver.MyCustomViewer" />
```

5- ViewResolvers

■ XmlViewResolver

```
1  <!-- WEB-INF/views.xml -->
2  <beans>
3      <bean id="home" class="org.springframework.web.
4          servlet.view.JstlView">
5          <property name="url"
6              value="/WEB-INF/view/home.jsp" />
7      </bean>
8      <bean id="admin/home" class="org.springframework.
9          web.servlet.view.JstlView">
10         <property name="url"
11             value="/WEB-INF/view/admin.jsp" />
12     </bean>
13 </beans>
```

5- ViewResolvers

- XmlViewResolver

```
1  <!-- nhanh-servlet.xml -->
2
3  <beans>
4      <bean class="org.springframework.web.servlet.
5              view.XmlViewResolver">
6          <property name="location">
7              <value>/WEB-INF/views.xml</value>
8          </property>
9      </bean>
10 </beans>
```

5- ViewResolvers

■ XmlViewResolver

```
1 @Controller
2 @RequestMapping("/nhanh")
3 public class WelcomeController {
4     @RequestMapping("/home")
5     public String home(Model model) {
6         ...
7         return "home";
8     }
9
10    @RequestMapping("/admin/home")
11    public String admin() {
12        .
13        return "admin/home";
14    }
15 }
```

5- ViewResolvers

- MultipartResolver
- LocaleResolver
- ThemeResolver



HỎI VÀ ĐÁP

Reference

- [1] Willie Wheeler and Joshua White. 2013. *Spring in Practice* (1st ed.). Manning Publications Co., Greenwich, CT, USA.