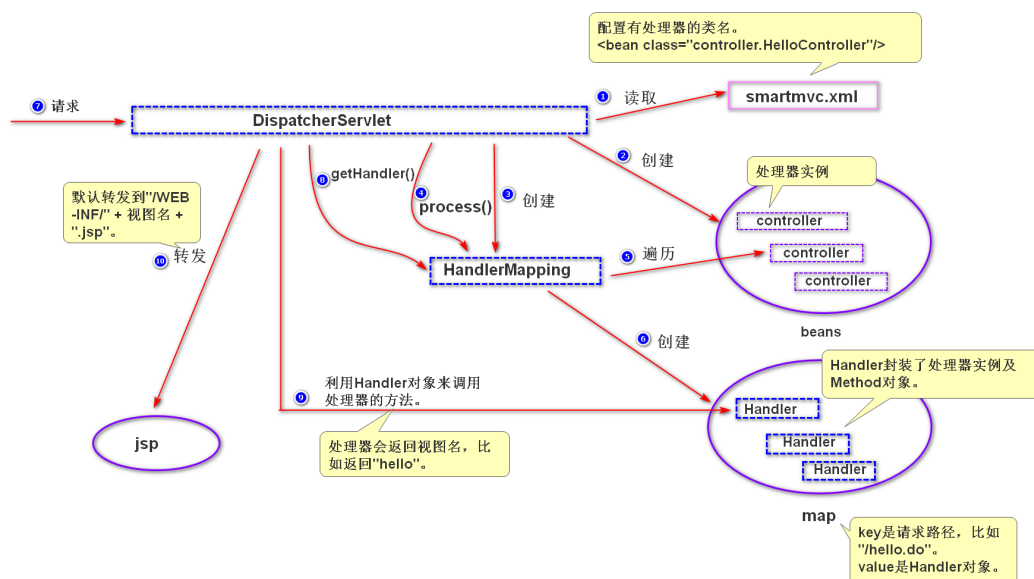
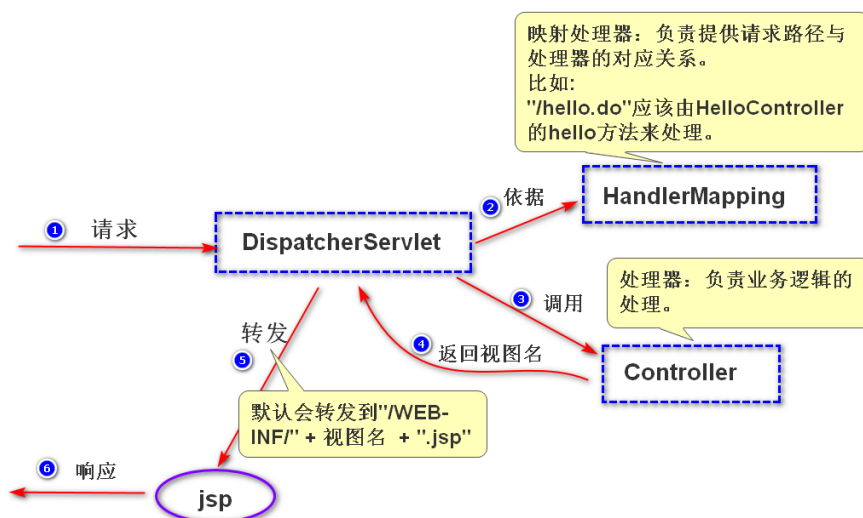


# SmartMVC

## 1. SmartMVC是什么？

是一个类似于SpringMVC的web mvc框架，其核心是一个可以通用的控制器。

## 2. 主要组件及其关系？



课后练习1：写一篇SmartMVC的使用文档，告诉开发人员如何使用SmartMVC来开发一个web应用。

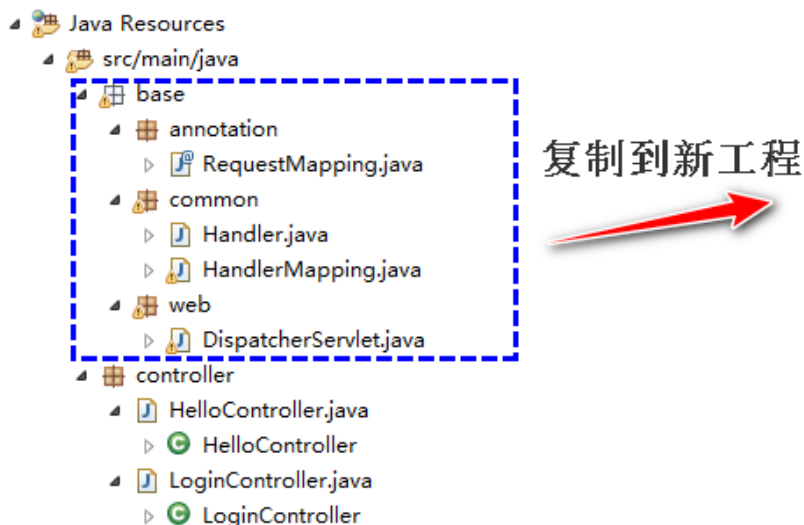
参考：

step1:创建maven工程。

step2.导包(dom4j)。

```
1 <dependency>
2     <groupId>dom4j</groupId>
3     <artifactId>dom4j</artifactId>
4     <version>1.6.1</version>
5 </dependency>
```

step3.将base包的代码复制到新工程。



step4.在resources下添加smartmvc的配置文件。

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans>
3     <!--
4         配置处理器的相关信息。
5         其中，class属性用来指定处理器类名。
6     -->
7     <bean class="controller.HelloController"/>
8 </beans>
9
```

step5.在web.xml中配置DispatcherServlet。

```
1 <servlet>
2     <servlet-name>DispatcherServlet</servlet-name>
```

```

3      <servlet-class>base.web.DispatcherServlet</servlet-class>
4      <!--
5          提定配置文件名
6      -->
7      <init-param>
8          <param-name>configLocation</param-name>
9          <param-value>smartmvc.xml</param-value>
10     </init-param>
11     <!--
12         配置启动加载。
13         即容器启动之后，会立即将这个Servlet实例化和初始化。
14         参数值是一个大于等于0的整数，越小，优先级越高（即容器优先创建其实例）
15     -->
16     <load-on-startup>1</load-on-startup>
17 </servlet>
18 <servlet-mapping>
19     <servlet-name>DispatcherServlet</servlet-name>
20     <url-pattern>*.do</url-pattern>
21 </servlet-mapping>
22

```

step6.在WEB-INF下添加JSP文件。

step7.添加相应的处理器类(比如HelloController)。

```

1  @RequestMapping("/demo")
2  public class HelloController {
3
4      @RequestMapping("/hello.do")
5      public String hello() {
6          System.out.println("HelloController's hello()");
7          /*
8           * 返回视图名。
9           * DispatcherServlet默认会转发到"/WEB-INF/" + 视图名 + ".jsp"
10          */
11          return "hello";
12      }
13  }

```

step8.在SmartMVC配置文件中配置处理器。

step9.部署运行。

课后练习2: 对SmartMVC框架进行改进，允许将  
@RequestMapping注解加到处理器类前面，比如:

```

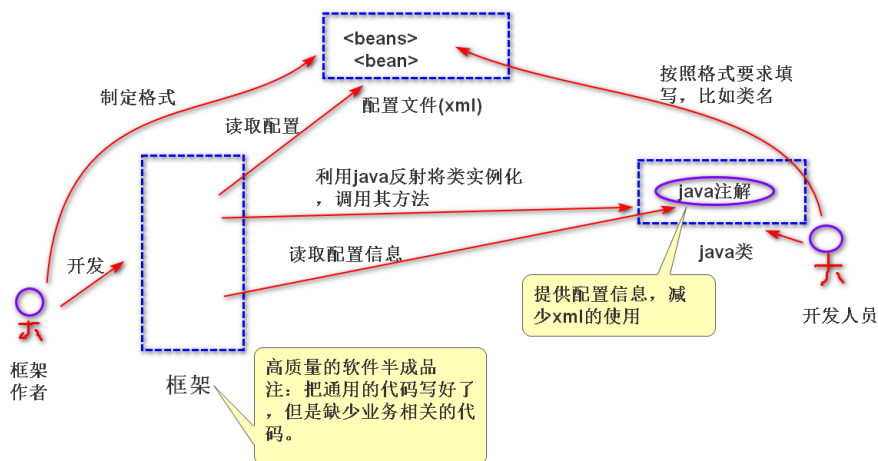
@RequestMapping("/demo")
public class HelloController{
    @RequestMapping("/hello.do")
    public String hello(){
        ....
    }
}

```

以上配置，对应的请求路径是”/demo/hello.do”。

参考：参见名为smartmvc-demo的工程。

其它：java反射与java注解在框架中的作用



其它:

路径问题:

- 1.以”/”开头的路径是绝对路径，不以”/”开头的路径是相对路径。在实际开发当中，应该优先使用绝对路径。
- 2.如何写绝对路径？

链接、表单提交、重定向从应用名开始写，而转发从应用名之后开始写。

练习：（3：25~3：55）

step1.创建一个maven工程(比如叫smartmvc)。

step2.导包(dom4j)。

step3.在WEB-INF下添加一个hello.jsp

step4.在controller包下面添加一个HelloController类。

step5.在base.annotation下添加一个@RequestMapping注解。

step6.在resources下添加smartmvc.xml。

step7.在base.web包下添加DispatcherServlet。