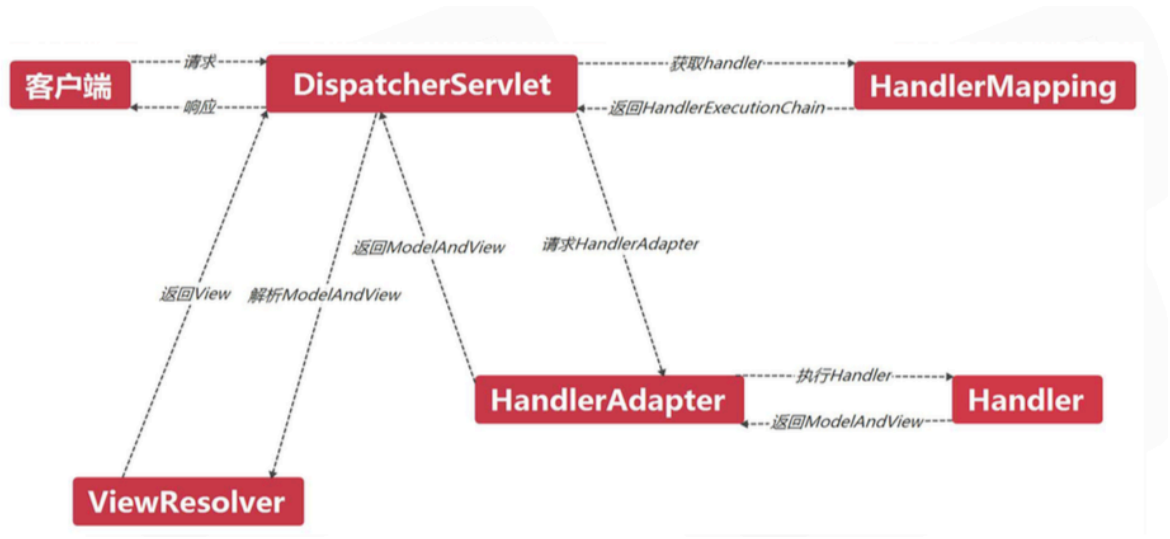
```xml
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>4.3.1.RELEASE</version>
</dependency>
```

在web.xml中配置。DispatcherServlet

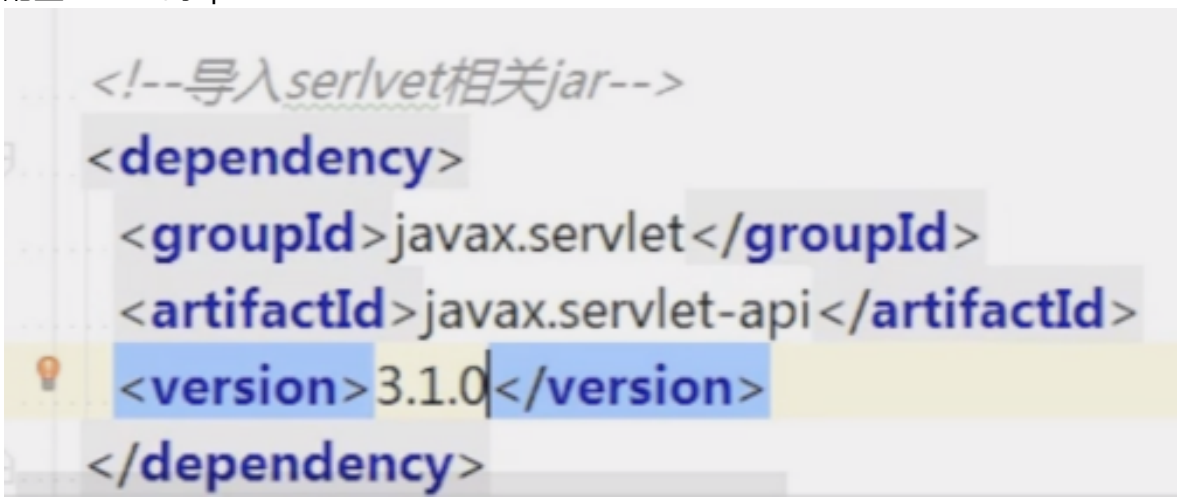
```xml
<servlet>
  <servlet-name>SpringMVC</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <!--配置springmvc.xml的路径-->
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:springmvc.xml</param-value>
  </init-param>
</servlet>
```

配置对应的servlet-mapping

```xml
<servlet-mapping>
  <servlet-name>SpringMVC</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
```



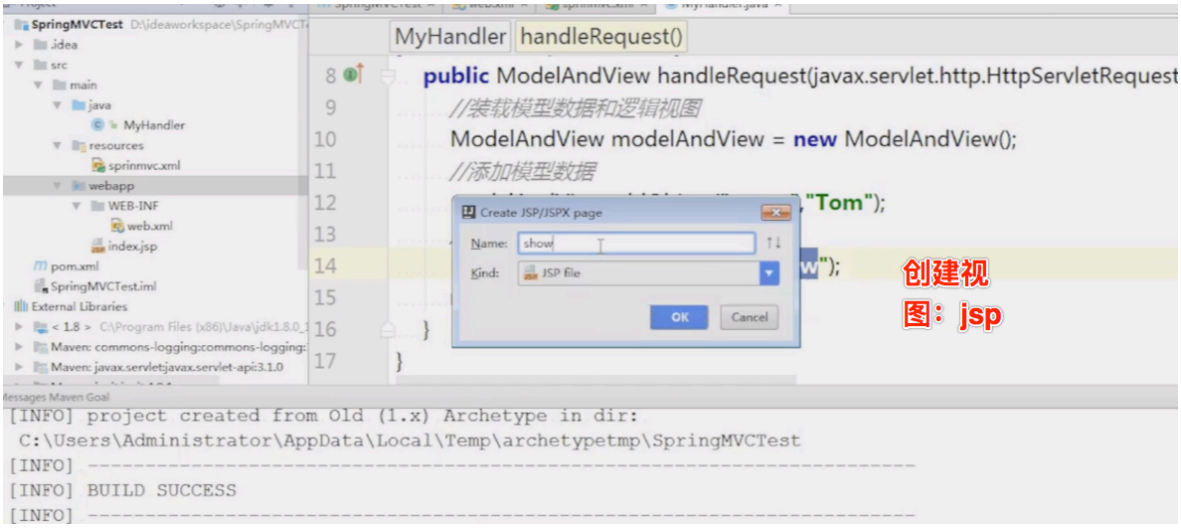
配置Servlet的api



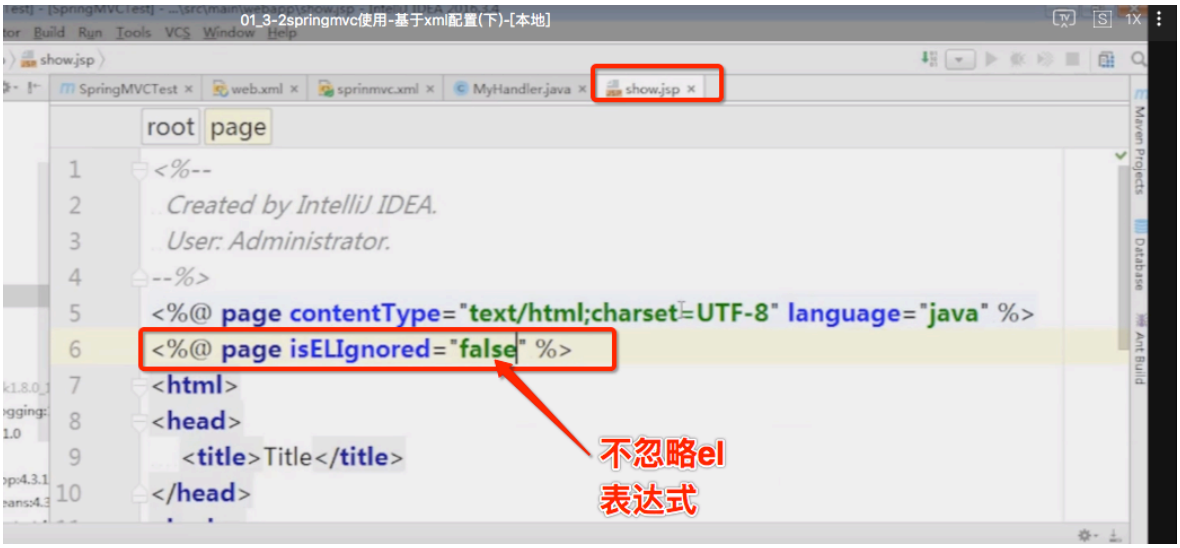
返回的是

然后spring的视图解析器将逻辑视图转换为物理视图，并且将数据（即model）填充进去　　为逻辑视图添加数据

创建物理视图、



不忽略el表达式



z

在springmvc.xml中进行配置，也就是说，在配置文件中进行设置，使得物理视图和逻辑视图进行关联

将testHandler映射到java类当中，同样是在springmvc.xml中进行配置

配置视图解析器，即ViewResolver

```
<!--配置视图解析器-->
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"></bean>
```

当前的逻辑视图是show这个单词

```
MyHandler  handleRequest()
        ModelAndView modelAndView = new ModelAndView();
        //添加模型数据
        modelAndView.addObject("name","Tom");
        //添加逻辑视图
        modelAndView.setViewName("show");
        return modelAndView;
    }
}

<!--配置视图解析器-->
<bean class="org.springframework.web.servlet.view.InternalResourceViewRes
  <!--配置前缀-->
  <property name="prefix" value="/"></property>
  <!--配置后缀-->
  <property name="suffix" value=".jsp"></property>
</bean>
```

配置前缀和后缀，针对show来说的

通过注解方式完成Handler的定义

```
 * Created by Administrator.
 */
public class AnnotationHandler {

    /**
     * 业务方法：ModelAndView完成数据的传递，视图的解析
     */
    public ModelAndView modelAndViewTest(){
        //创建ModelAndView对象
        ModelAndView modelAndView = new ModelAndView()
    }

<no parameters>
String viewName
```