非注解的映射器适配器配置

前端控制器

前端控制器用来分配浏览器对资源的请求，web.xml是一切B/S系统中交互的第一站，所以服务器的过滤器（字符集的过滤等）和springMVC的统一路口在web.xml下进行配置。

强调！配置就是一种映射关系，自己去理解一下

1. 浏览器请求过来后先去寻找mapper。
2. 映射会根据里面的映射关系去寻找相对应得实体（springMVC.XML）。
3. 找到后，跳进实体类中进行下一步操作。

前端控制器的映射配置

```
<servlet>
    <servlet-name>springmvc</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <!-- contextConfigLocation配置springmvc加载的配置文件（配置处理器映射器、适配器等等）
     如果不配置contextConfigLocation，默认加载的是/WEB-INF/servlet名称-servlet.xml（springmvc-servlet.xml）
     -->
    <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:springmvc.xml</param-value>
    </init-param>
    </servlet>
    <servlet-mapping>
    <servlet-name>springmvc</servlet-name>
    <!--
    第一种：*.action访问以.action结尾由DispatcherServlet进行解析。
    第二种：/所有访问的地址都由DispatcherServlet进行解析，对于静态文件的解析需要配置不让DispatcherServlet进行解析
       使用此种方法，可以实现RESTful风格。
    第三种：  这样配置不对。使用这种配置，最终要转发到一个jsp页面时，仍然会由DispatcherServlet解析jsp页面，
       不能根据jsp页面找到handler，会报错。-->
    <url-pattern>*.action</url-pattern>
    </servlet-mapping>
```

注释：所有后缀为.action的请求 都会需找springMVC容器去进行下一步的处理。所以可以说web.xml是前端请求通向后端的入口。

书写前端控制器

里面需要配置所需要的组件映射关系

1. handlerMapper（映射器）
2. handlerAdaptor（适配器）
3. View Resolver（视图解析器）

非注解版

1. **映射器**

handler的配置

- BeanNameUrlHandlerMapping方式
  BeanNameUrl处理器映射器，根据请求的url与spring容器中定义的bean的name进行匹配，从而从spring容器中找到bean实例。

```
    <!-- 配置Handler -->
    <bean id="itemsController1" name="/queryItems_test.action" class="com.yxd.ssm.controller.ItemsController1" />
```

```
  <!--将bean的name作为url进行查找，需要在配置handler时指定statement（就是url）
```

```
    所有的映射器都实现了HandlerMapping接口-->
 <bean class="org.springframework.web.servlet.handler.BeanNameUrlHandlerMapping" />
```

- SimpleUrlHandlerMapping方式
  simpleUrlHandlerMapping是BeanNameUrlHandlerMapping的增强版本，它可以将url和处理器bean的id进行统一映射配置。

```
<!-- 配置另外一个Handler -->
    <bean id="itemsController2" class="com.yxd.ssm.controller.ItemsController2" />
```

```
<!—简单url映射 -->
 <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
  <property name="mappings">
   <props>
    <prop key="/itemsN.action">controllerN的bean id</prop>
    <prop key="/items2.action">itemsController2</prop>
   </props>
  </property>
 </bean>
```

2. **适配器**

- SimpleControllerHandlerAdapter简单控制器处理器适配器，所有实现了org.springframework.web.servlet.mvc.Controller 接口的Bean通过此适配器进行适配、执行。

```
   <bean class="org.springframework.web.servlet.mvc.SimpleControllerHandlerAdapter" />
```

- HttpRequestHandlerAdapter，http请求处理器适配器，所有实现了org.springframework.web.HttpRequestHandler 接口的Bean通过此适配器进行适配、执行。

```
 <!-- 另一个非注解的适配器 -->
    <bean class="org.springframework.web.servlet.mvc.HttpRequestHandlerAdapter" />
```

视图解析器

```
<!-- ViewResolver -->
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
<property name="viewClass" value="org.springframework.web.servlet.view.JstlView"/>
<property name="prefix" value="/WEB-INF/jsp/"/>
<property name="suffix" value=".jsp"/>
</bean>
```

1. InternalResourceViewResolver：支持JSP视图解析
2. viewClass：JstlView表示JSP模板页面需要使用JSTL标签库，所以classpath中必须包含jstl的相关jar 包（如果不使用jstl，可以不用，开发会前后端的分离，所以我们会使用纯html去代替传统的jsp）；
3. prefix 和suffix：查找视图页面的前缀和后缀，最终视图的址为： 前缀+逻辑视图名+后缀，逻辑视图名需要在controller中返回ModelAndView指定，比如逻辑视图名为hello，则最终返回的jsp视图地址 “WEB-INF/jsp/hello.jsp”

完结