入门配置文件

叙述:

入门案例所用配置采取惯用开发配置,目的是为了简单了解spring web MVC。具体的配置会在后面的文件中具体介绍。

- 1. 准备jar包(专门去讲)
- 2. web.xml文件

```
<!-- 前端控制器 -->
 <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>
```

3.springMVC.xml文件

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"</pre>
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:mvc="http://www.springframework.org/schema/mvc"
      xmlns:p="http://www.springframework.org/schema/p"
      xmlns:context="http://www.springframework.org/schema/context"
      xmlns:aop="http://www.springframework.org/schema/aop"
      xmlns:tx="http://www.springframework.org/schema/tx"
      xsi:schemaLocation="http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
           http://www.springframework.org/schema/context
           http://www.springframework.org/schema/context/spring-context-3.0.xsd
           http://www.springframework.org/schema/aop
           http://www.springframework.org/schema/aop/spring-aop-3.0.xsd
           http://www.springframework.org/schema/tx
           http://www.springframework.org/schema/tx/spring-tx-3.0.xsd
           http://www.springframework.org/schema/mvc
           http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd
           http://www.springframework.org/schema/context
           http://www.springframework.org/schema/context/spring-context-3.0.xsd">
   <!-- 配置Handler -->
   <bean id="itemsController1" name="/queryItems test.action" class="com.swxc.ssm.controller.ItemsController1" />
   <!-- 配置另外一个Handler -->
   <bean id="itemsController2" class="com.swxc.ssm.controller.ItemsController2" />
   <!-- 对于注解的Handler,可以进行单个配置
    实际开发中,建议使用主键扫描
```

```
<!--<bean class="com.swxc.ssm.controller.ItemsController3" />-->
   <!-- 可以扫描controller、service、......
    这里让扫描controller,指定controller的包
   <context:component-scan base-package="com.swxc.ssm.controller" />
   <!-- 简单url映射 -->
   <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
       cproperty name="mappings">
          ops>
              <!-- 对itemsController1进行url映射, url是/queryItems1.action -->
               key="/queryItems1.action">itemsController1
               key="/queryItems2.action">itemsController1
               key="/queryItems3.action">itemsController2
          </props>
       </property>
   </bean>
   <!-- 处理器映射器
    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" />
   <!-- 注解的映射器 -->
   <bean class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping" />
   <!-- 处理器适配器
    所有的处理器适配器都实现HandlerAdapter接口
   <bean class="org.springframework.web.servlet.mvc.SimpleControllerHandlerAdapter" />
   <!-- 注解的适配器 -->
   <bean class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter" />
   <!-- 另一个非注解的适配器 -->
   <bean class="org.springframework.web.servlet.mvc.HttpRequestHandlerAdapter" />
   <!-- 使用mvc:annotation-driven可以代替注解映射器和注解适配器的配置
    mvc:annotation-driven默认加载很多的参数绑定方法,比如json转换解析器就默认加载了,
    如果使用mvc:annotation-driven,不用配置RequestMappingHandlerMapping和RequestMappingHandlerMapping
    实际开发使用mvc:annotation-driven
    -->
   <!--<mvc:annotation-driven />-->
   <!-- 视图解析器
    解析jsp视图,默认使用jstl标签, classpath下必须有jstl包
   <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
       <!-- 配置jsp路径的前缀和后缀 -->
       cproperty name="prefix" value="/WEB-INF/jsp/" />
       cproperty name="suffix" value=".jsp" />
   </bean>
</beans>
```