配置

web.xml

<servlet>  
 <servlet-name>dispatcher</servlet-name>  
 <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  
<!—load on startup必须放在最后-->  
 <load-on-startup>1</load-on-startup>  
</servlet>  
<servlet-mapping>  
 <servlet-name>dispatcher</servlet-name>

<!—指定拦截那些请求-->  
 <url-pattern>/</url-pattern>  
</servlet-mapping>

dispatcher-servlet.xml//servlet配置文件

<!-- 默认的注解映射的支持 -->

<mvc:annotation-driven/>

<!-- 如果当前请求为“/”时，则转发到“/helloworld/index” -->

<mvc:view-controller path="/" view-name="forward:/helloworld/index"/>

<!-- 静态资源映射 -->

<mvc:resources mapping="/js/\*\*" location="/WEB-INF/js/" />

<mvc:resources mapping="/css/\*\*" location="/WEB-INF/css/" />

<mvc:resources mapping="/fonts/\*\*" location="/WEB-INF/fonts/" />

<mvc:resources mapping="/plugins/\*\*" location="/WEB-INF/plugins/" />

<mvc:resources mapping="images/\*\*" location="/WEB-INF/images/" />

<!-- 当上面要访问的静态资源不包括在上面的配置中时，则根据此配置来访问 -->

<mvc:default-servlet-handler/>

<!-- 开启controller注解支持 -->

<!-- use-default-filters="false" 只扫描指定的注解 -->

<context:component-scan base-package="com.demo.web.controllers" use-default-filters="false">

<context:include-filter type="annotation" expression="org.springframework.stereotype.Controller"/>

</context:component-scan>

<!-- 视图解析器 -->

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="viewClass" value="org.springframework.web.servlet.view.JstlView"/>

<property name="contentType" value="text/html"/>

<property name="prefix" value="/WEB-INF/views/"/>

<property name="suffix" value=".jsp"/>

</bean>

@Controller

@RequestMapping(value = "/helloworld")

public class HelloWorldController {

@RequestMapping(value="/index", method = {RequestMethod.GET})

public ModelAndView index(){

ModelAndView modelAndView = new ModelAndView();

modelAndView.addObject("message", "Hello World!");

modelAndView.setViewName("index"); //指定该action所对应的视图名称，解析视图会在dispatcher-servlet.xml

//文件中指定的视图文件夹

return modelAndView;

}

}

@Controller注解将该class指定为controller,controller上的@RequestMapping注解的value指定该controller所映射的请求，方法上的@RequestMapping将该方法为一个action,value值指定为该action所映射的请求，method