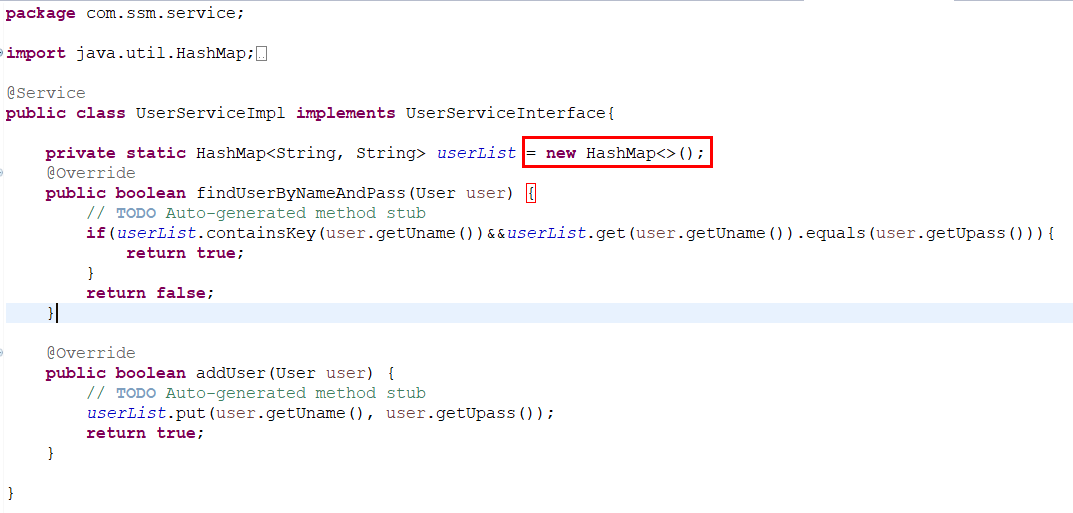
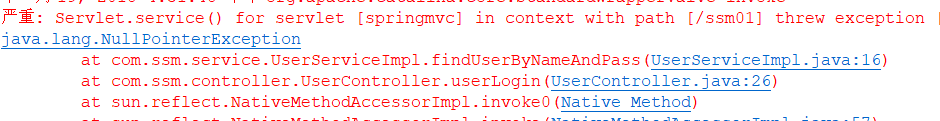
**0、问题**

**问题1：报空指针异常错误**





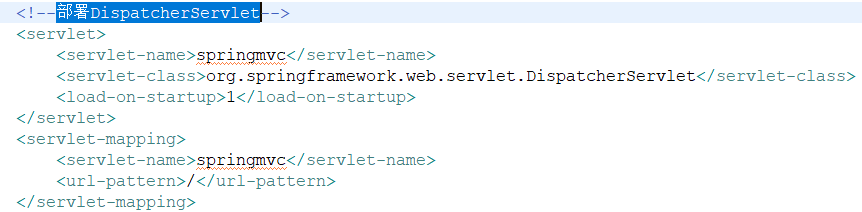
**问题2：路径错误**



basePath ——http://localhost:8080/ssm01/

1、Web.xml文件配置

在web.xml文件中，配置前端控制器DispatcherServlet



**2、SpringMVC.xml文件配置**

在springmvc.xml文件中，配置“处理器映射器、处理器适配器、视图解析器”



**3、Spring MVC工作原理**



总结出Spring MVC的工作流程如下：

1．客户端请求提交到DispatcherServlet；

2．由DispatcherServlet控制器寻找一个或多个HandlerMapping，找到处理请求的Controller；

3．DispatcherServlet将请求提交到Controller；

4．Controller调用业务逻辑处理后，返回ModelAndView；

5．DispatcherServlet寻找一个或多个ViewResoler视图解析器，找到ModelAndView指定的视图；

6．视图负责将结果显示到客户端。

**4、@Controller注解及扫描包**

Controller注解类型，为了让@Controller控制器类被Spring MVC框架扫描到，需要在配置文件中声明spring-context，并使用<context:component-scan/>元素指定控制器类的基本包（请确保所有控制器类都在基本包及其子包下）

<!-- 使用扫描机制，扫描controller包 -->

<context:component-scan base-package=*"com.ssm.controller"*/>

**5、@RequestMapping方法中的参数类型**

@RequestMapping请求处理方法中常出现的参数类型

@RequestMapping("/login")

public String login(**HttpSession session,**

**HttpServletRequest request,**

**HttpServletResponse Response,**

**Model model**

)

{ }

**Controller类中每个请求处理方法可以有多个不同类型的参数，以及一个多种类型的返回结果。**

**6、controller类接受jsp页面参数的方式**

Controller类接收jsp页面传过来的请求参数，共有一下几种方法

2.2.1 通过实体bean接收请求参数

2.2.2 通过处理方法的形参接收请求参数**（必须一一对应）**

2.2.3 通过**HttpServletRequest**接收请求参数

2.2.4 通过**@PathVariable**接收URL中的请求参数

2.2.5 通过**@RequestParam**接收请求参数（**名字一一对应**）

2.2.6 通过**@ModelAttribute**接收请求参数

**7、SpringMVC中的转发和重定向**

**//转发到一个请求方法**（同一个控制器类里，可省略/index/）

return "forward:/index/isLogin";

**//重定向到一个请求方法**

return "redirect:/index/isRegister";

**//转发到一个视图 默认**

return "register";

**8、SpringMVC中的静态资源单独处理**

需要单独处理，即不需要DispatcherServlet去处理的资源

使用：

<mvc:resources location="/html/" mapping="/html/\*\*"></mvc:resources>