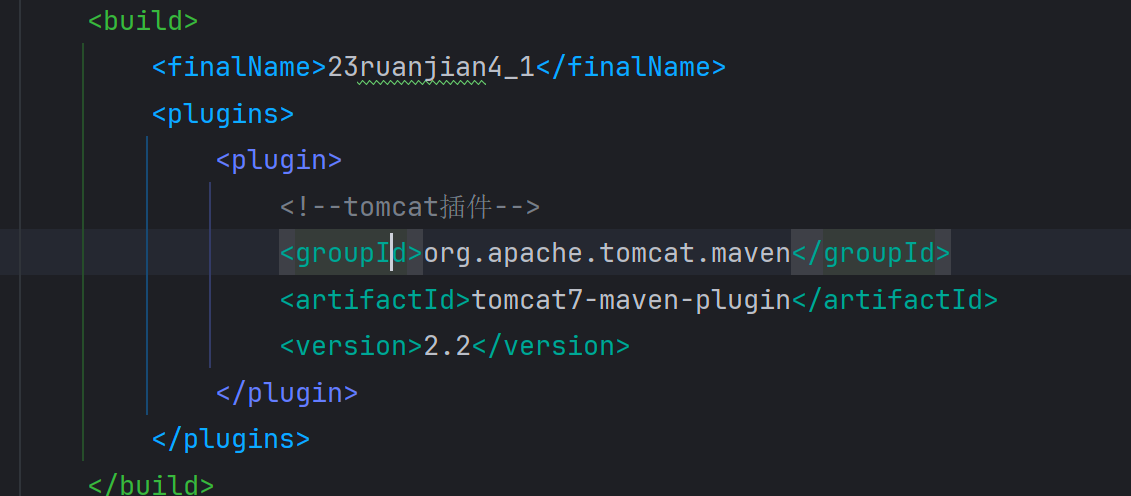
Servlet

1. java提供的开发动态web资源技术
2. 本地部署
3. 骨架：直接在run中设置Eidt Configuriations,选择Tomcat部署
4. 插件（推荐）：在pom中添加
5. 快速入门：创建一个java类，实现Servlet接口，并且实现其中的service方法，设置访问为/demo1。文本

   AI 生成的内容可能不正确。
6. 生命周期
7. 加载和实例化：默认情况下Servlet第一次被访问时，由容器创建Servlet对象，可以通过设置调整创建的时机
8. 初始化：在实例化后，容器会调用Servlet的init（）方法初始化对象，完成一些诸如加载配置文件，创建连接等初始化工作，该方法只能调用一次。
9. 请求处理：每次请求Servlet，Servlet容器都会调用Servlet的service方法请求进行处理
10. 服务终止：当需要释放内存或者容器关闭时，容器会调用destroy（）方法完成资源释放，容器会释放该实例，实例随后会被java的垃圾收集器所回收。
11. 常用方法
12. init：文本

    AI 生成的内容可能不正确。
13. service：文本

    AI 生成的内容可能不正确。
14. destroy：文本

    AI 生成的内容可能不正确。
15. getServletConfig：文本

    AI 生成的内容可能不正确。
16. getServletInfo：文本

    AI 生成的内容可能不正确。
17. Servlet体系结构：

HttpServlet：类继承httpServlet类，重写doGet方法和doPost方法，分别实现对于输入信息和接收信息的处理。

1. Servlet urlPattern配置
2. 一个servlet可以配置多个访问路径
3. urlPattern配置规则：
4. 精确匹配:访问路径和设置路径一致
5. 目录匹配：

捕获一个目录下的所有资源

1. 扩展名匹配：

适合统一处理某一类文件

1. 任意匹配：

适合访问网站后一直要维持的状态或信息

1. xml配置servlet（不推荐，建议使用注解）



1. Reuqest（请求）
2. request继承体系：

ServletRequest：java提供的请求对象的根接口。

HttpServletRequest:java提供的给http协议封装的请求对象接口

RequestFacade：Tomcat定义的实现类

1. Request获取请求数据：

请求行：String getMethod（）：获取请求方式：GET

String getContextPath（）：获取虚拟目录：/Request-demo

StringBuffer getRequestURL（）：获取URL（统一资源定位符）路径

String getRequestURI（）：获取URI（统一资源标识符）

String getQueryString（）：获取请求参数（GET方式）

请求头：String getHeader（String name）：获取请求头的名称，获取值

请求体：

ServletInputStream getInputStream():获取字节输入流

BufferedReader getReader（）：获取字符输入流

1. Request通用方式获取请求参数：

Get方式：String getQueryString（）

POST方式：BufferedReader getReader（）

通用方式获取参数：

文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

GET和POST都可以通过此方式获取请求数据

1. POST和GET出现中文乱码解决方式
2. Post：设置输入流的编码

req.setCharacterEncoding(“UTF-8”)

1. Get：通用方式（get/post）：先解码，再编码

new String(username.getBytes(“ISO-8895-1”),”UTF-8”)

1. URL编码实现方式：

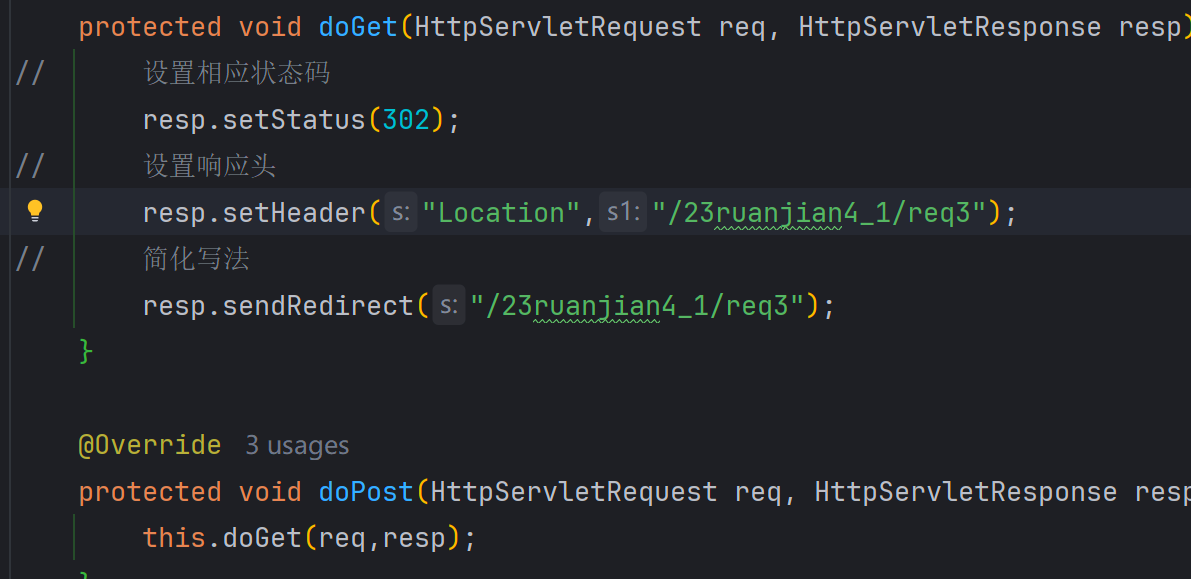
编码：URLEncoder.encode(str,”UTF-8”);

解码：URLDecoder.decode(s,”ISO-8895-1”);

1. Request请求转发
2. 定义：一种在服务器内部的资源跳转的方式
3. 实现：通过req.getRequestDispatcher("访问路径").forward(req,resp);
4. 功能：将当前进程的资源发送到另一个进程中
5. 共享资源：req.setAttribute(key,value);相传输数据以键值对的形式展现
6. 特点：浏览器的地址栏不会发生改变；只能够在服务器内部进行数据的传输；能够实现服务器内部的资源共享
7. Response对象
8. 功能：设置响应数据
9. 响应数据组成：

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

1. 重定向：一种资源跳转的方式。返回响应给客户端，并告诉客户端要去重新访问的位置。
2. 重定向实现：
3. 重定向特点：地址栏发生改变；可以重定向到任意位置的资源（服务器内、外都可以）；两次请求，不能够在多个资源使用Request共享数据
4. 响应字符/响应字节

响应字符：电脑萤幕的截图

AI 生成的内容可能不正确。

响应字节：

文本

AI 生成的内容可能不正确。

内置IO插件：文本

AI 生成的内容可能不正确。