**Springboot多module项目工程搭建**

# 说明

版权归属：鑫学猿公司

微信：youhouduanyanfa

Java开发工程师转行辅导体系,保障拿到java工程师offer

图形用户界面, 文本, 应用程序

描述已自动生成

**目标**

1. 学习创建Springboot的多module项目
2. 学习搭建前后端分离的项目工程结构

**内容**

**创建多module项目**

本文通过构建一个包含5个子模块的项目，来演示 **SpringBoot** 在 **Maven** 环境的多模块构建过程。



1. 创建父工程
2. 通过Spring Initializer创建

|  |  |
| --- | --- |
|  |  |

1. 创建好后删除刚创建工程里不需要的文件， 只保留：**.idea** 文件夹 、项目 **pom** 文件、以及一个 **\*.iml** 文件

删除前->删除后

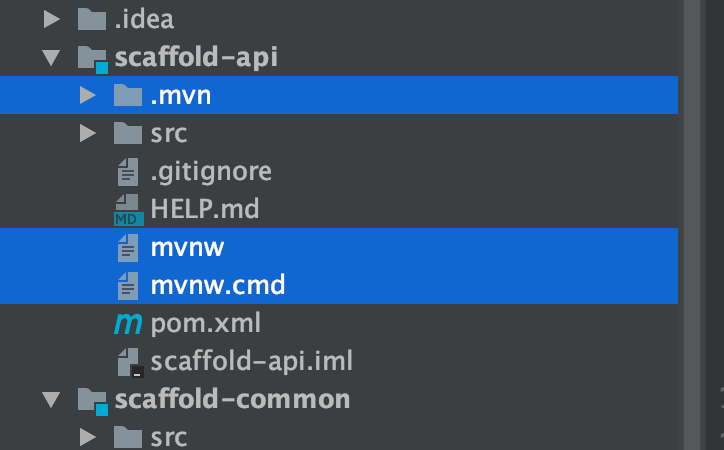
|  |  |
| --- | --- |
|  |  |

1. 创建子模块
2. 右键点击父工程，选择 **New** -> **Module...** 创建子模块。这里依次创建 **scaffold-common**、**scaffold-api、scaffold-dao**、**scaffold-service** 和 **scaffold-web** 共 **5** 个模块

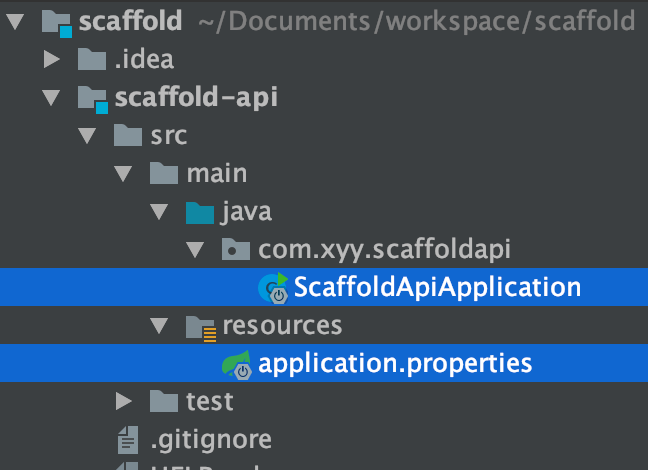
|  |
| --- |
| **注意**：除了 **scaffold-web** 子模块创建时选择添加 **Spring Web** 依赖（当然也可以创建时不添加，等后面再手动编辑 **pom.xml** 文件添加），其他模块暂时不添加依赖。 |

|  |  |
| --- | --- |
|  |  |

1. 将所有子模块的 **mvnw**、**mvnw.cmd** 文件及 **.mvn** 文件夹全部删除



1. 对于 **src** 里的内容，只保留 **scaffold-web** 的启动类和配置文件，其他子模块的的启动类和配置文件都删除



1. 编辑父工程的pom.xml文件
2. 将父工程 **pom.xml** 文件修改成如下内容，里面声明该父工程包含的子模块，同时抽取统一的配置信息和依赖版本控制，这样可以方便子 **pom** 直接引用，简化子 **pom** 的配置

|  |
| --- |
| * 多模块项目中，父模块打包类型必须是 **pom**。 * 因为开发框架是 **spring boot**，父模块默认继承 **spring-boot-starter-parent**，因此可以删除 **spring-boot-starter** 和 **spring-boot-starter-test** 依赖（祖先已经包含了） |

父工程pom.xml如下：

|  |
| --- |
| XML <?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>   <groupId>com.xyy</groupId>  <artifactId>scaffold</artifactId>  <version>0.0.1-SNAPSHOT</version>  <name>scaffold</name>  <description>Demo project for Spring Boot</description>   <!-- 父模块打包类型必须为pom -->  <packaging>pom</packaging>   <!-- parent指明继承关系，给出被继承的父项目的具体信息-->  <parent>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-parent</artifactId>  <version>2.7.12</version>  <relativePath/> <!-- lookup parent from repository -->  </parent>   <properties>  <java.version>1.8</java.version>  </properties>   <!-- 模块说明：这里声明多个子模块 -->  <modules>  <module>scaffold-web</module>  <module>scaffold-api</module>  <module>scaffold-common</module>  <module>scaffold-dao</module>  <module>scaffold-service</module>  </modules>   <!-- 版本说明：这里统一管理依赖的版本号 -->  <dependencyManagement>  <dependencies>  <!-- 这里可以删除spring-boot-starter 和 spring-boot-starter-test 依赖（祖先已经包含了） -->  <!--<dependency>-->  <!--<groupId>org.springframework.boot</groupId>-->  <!--<artifactId>spring-boot-starter</artifactId>-->  <!--</dependency>-->   <!--<dependency>-->  <!--<groupId>org.springframework.boot</groupId>-->  <!--<artifactId>spring-boot-starter-test</artifactId>-->  <!--<scope>test</scope>-->  <!--</dependency>-->   <dependency>  <groupId>com.xyy</groupId>  <artifactId>scaffold-web</artifactId>  <version>0.0.1-SNAPSHOT</version>  </dependency>  <dependency>  <groupId>com.xyy</groupId>  <artifactId>scaffold-common</artifactId>  <version>0.0.1-SNAPSHOT</version>  </dependency>  <dependency>  <groupId>com.xyy</groupId>  <artifactId>scaffold-dao</artifactId>  <version>0.0.1-SNAPSHOT</version>  </dependency>  <dependency>  <groupId>com.xyy</groupId>  <artifactId>scaffold-service</artifactId>  <version>0.0.1-SNAPSHOT</version>  </dependency>  <dependency>  <groupId>com.xyy</groupId>  <artifactId>scaffold-api</artifactId>  <version>0.0.1-SNAPSHOT</version>  </dependency>  </dependencies>  </dependencyManagement>    <build>  <plugins>  <plugin>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-maven-plugin</artifactId>  </plugin>  </plugins>  </build>  </project> |

1. 编辑子模块pom.xml
2. 子模块 **scaffold-common** 的 **pom.xml** 文件内容如下，其中 **parent** 要使用顶层的父模块，同时由于我们项目用到了 **Lombok** ，所以还添加了 **lombok** 依赖

|  |
| --- |
| 由于子模块的配置信息会继承父模块的，所以子模块原来的 **properties** 可删掉 |

|  |
| --- |
| XML <?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>   <groupId>com.xyy</groupId>  <artifactId>scaffold-common</artifactId>  <version>0.0.1-SNAPSHOT</version>  <name>scaffold-common</name>  <description>Demo project for Spring Boot</description>   <!-- 继承本项目的父工程 -->  <parent>  <groupId>com.xyy</groupId>  <artifactId>scaffold</artifactId>  <version>0.0.1-SNAPSHOT</version>  <relativePath/> <!-- lookup parent from repository -->  </parent>   <properties>  <java.version>1.8</java.version>  </properties>  <dependencies>  <dependency>  <groupId>org.projectlombok</groupId>  <artifactId>lombok</artifactId>  <version>1.18.6</version>  </dependency>  </dependencies> </project> |

1. 子模块**scaffold-dao** 的 **pom.xml** 文件内容如下，同样 **parent** 要使用顶层的父模块，并添加 **scaffold-common** 子模块，以及数据库相关依赖

|  |
| --- |
| XML <?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>   <groupId>com.xyy</groupId>  <artifactId>scaffold-dao</artifactId>  <version>0.0.1-SNAPSHOT</version>  <name>scaffold-dao</name>  <description>Demo project for Spring Boot</description>   <parent>  <groupId>com.xyy</groupId>  <artifactId>scaffold</artifactId>  <version>0.0.1-SNAPSHOT</version>  <relativePath/> <!-- lookup parent from repository -->  </parent>   <properties>  <java.version>1.8</java.version>  </properties>  <dependencies>  <dependency>  <groupId>com.xyy</groupId>  <artifactId>scaffold-common</artifactId>  </dependency>  </dependencies>  </project> |

1. 子模块 **scaffold-service** 的 **pom.xml** 文件内容如下，同样 **parent** 要使用顶层的父模块，并添加 **scaffold-dao** 子模块依赖

|  |
| --- |
| 实际开发中dao模块会引入对数据相关包的依赖，如mysql、Druid连接池、mybatis等 |

|  |
| --- |
| XML <?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>   <groupId>com.xyy</groupId>  <artifactId>scaffold-service</artifactId>  <version>0.0.1-SNAPSHOT</version>  <name>scaffold-service</name>  <description>Demo project for Spring Boot</description>   <parent>  <groupId>com.xyy</groupId>  <artifactId>scaffold</artifactId>  <version>0.0.1-SNAPSHOT</version>  <relativePath/> <!-- lookup parent from repository -->  </parent>   <properties>  <java.version>1.8</java.version>  </properties>  <dependencies>  <dependency>  <groupId>com.xyy</groupId>  <artifactId>scaffold-dao</artifactId>  </dependency>  </dependencies> </project> |

1. 子模块 **scaffold-web** 的 **pom.xml** 文件内容如下，同样 **parent** 要使用顶层的父模块，并添加 **scaffold-service** 子模块依赖

|  |
| --- |
| **注意**： 之前创建这个子模块的时候已经添加了 **spring-boot-starter-web** 依赖，如果没有则手动添加 |

|  |
| --- |
| XML <?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>   <groupId>com.xyy</groupId>  <artifactId>scaffold-web</artifactId>  <version>0.0.1-SNAPSHOT</version>  <name>scaffold-web</name>  <description>Demo project for Spring Boot</description>   <parent>  <groupId>com.xyy</groupId>  <artifactId>scaffold</artifactId>  <version>0.0.1-SNAPSHOT</version>  <relativePath/> <!-- lookup parent from repository -->  </parent>   <properties>  <java.version>1.8</java.version>  </properties>  <dependencies>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>   <dependency>  <groupId>com.xyy</groupId>  <artifactId>scaffold-service</artifactId>  </dependency>  </dependencies>   <build>  <plugins>  <plugin>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-maven-plugin</artifactId>  </plugin>  </plugins>  </build>  </project> |

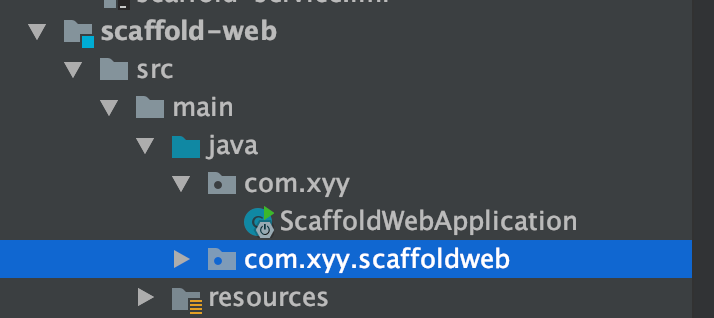
1. 子模块 **scaffold-api** 的 **pom.xml** 文件内容如下，同样 **parent** 要使用顶层的父模块

|  |
| --- |
| XML <?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>   <groupId>com.xyy</groupId>  <artifactId>scaffold-api</artifactId>  <version>0.0.1-SNAPSHOT</version>  <name>scaffold-api</name>  <description>Demo project for Spring Boot</description>   <parent>  <groupId>com.xyy</groupId>  <artifactId>scaffold</artifactId>  <version>0.0.1-SNAPSHOT</version>  <relativePath/> <!-- lookup parent from repository -->  </parent>   <properties>  <java.version>1.8</java.version>  </properties>  <dependencies>  <dependency>  <groupId>com.xyy</groupId>  <artifactId>scaffold-common</artifactId>  </dependency>   <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-test</artifactId>  <scope>test</scope>  </dependency>  </dependencies> </project> |

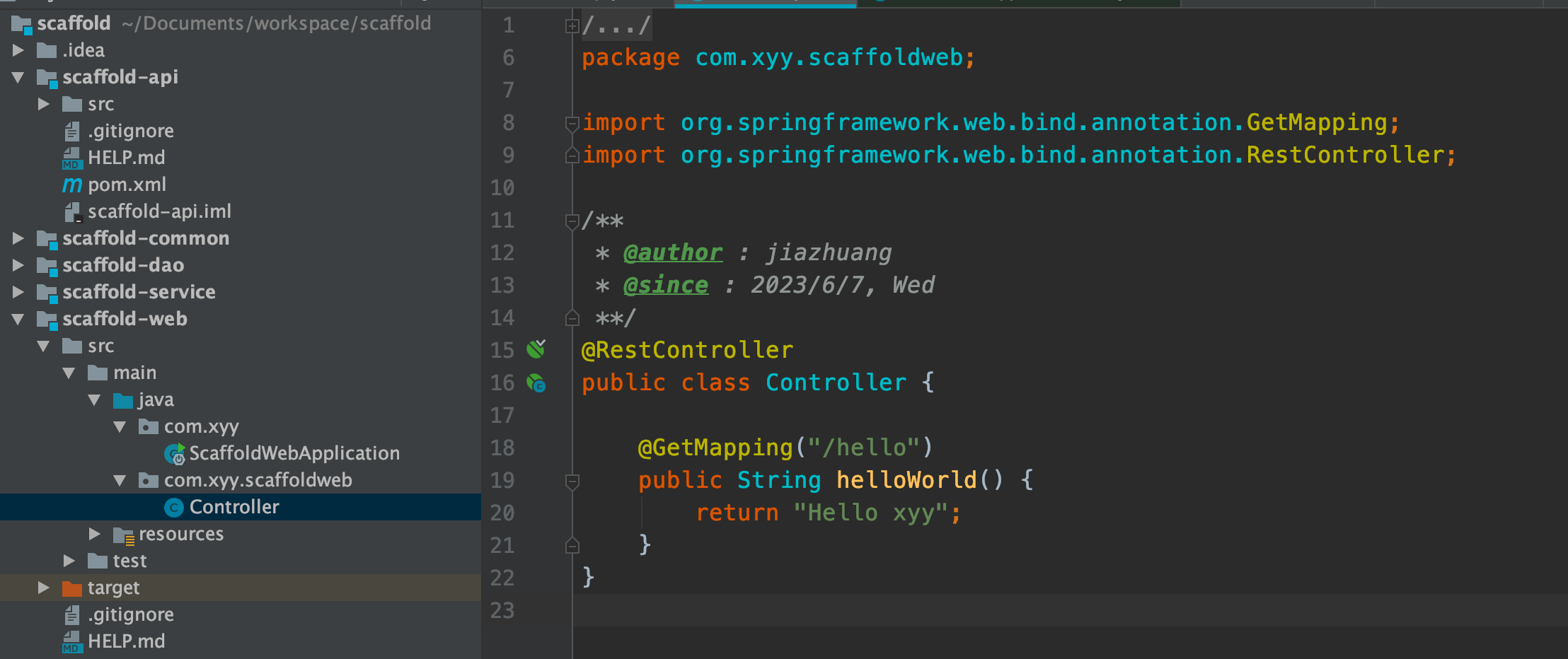
1. 移动项目启动类所在包
2. 目前项目启动类 ScaffoldWebApplication 在 **com.xyy.scaffold** 包下面，我们需要将其移动移动到 **com.xyy** 包下
3. 移动的方式就是右键点击 ScaffoldWebApplication 选择 **Refactor** -> **Move** ，将to package改成com.xyy

|  |  |
| --- | --- |
|  |  |

1. 移动后



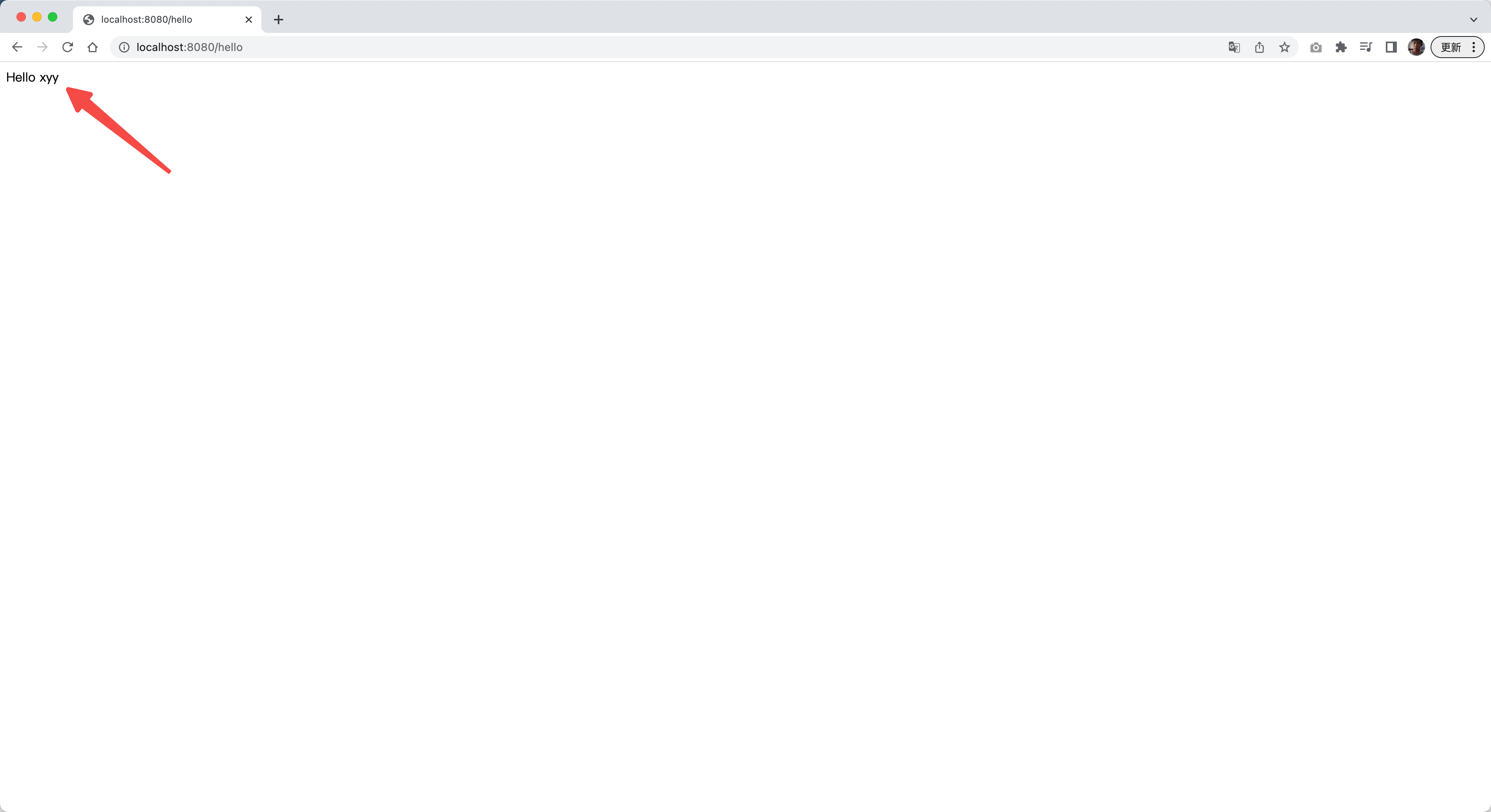
1. 编写controller并启动
2. 在com.xyy.scaffoldweb下写个测试Controller



1. 启动项目

|  |  |
| --- | --- |
|  |  |

1. 打开浏览器访问



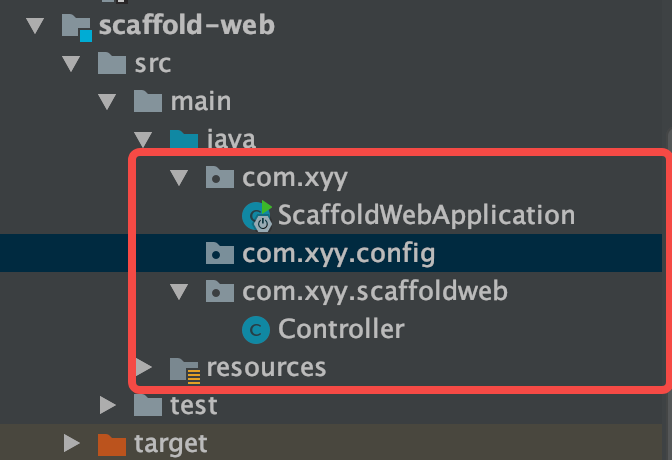
**搭建项目工程结构**

搭建好多module项目并成功启动后，我们再来构建项目工程结构

1. scaffold-web模块

主要包含一个启动类、一个web和一个config包

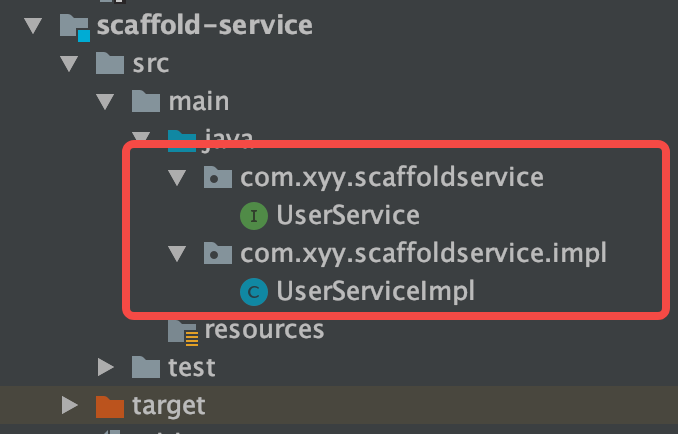
* web包：包含所有controller
* config包：负责web模块后续的统一配置类



1. scaffold-service模块

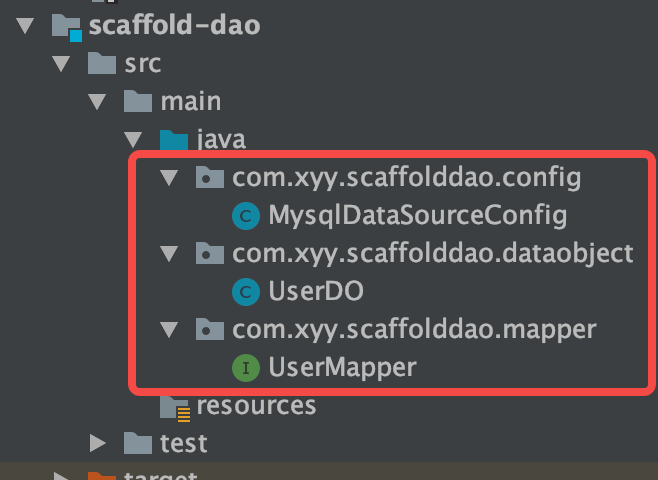
主要包含业务逻辑代码，包含业务接口及其实现类

* impl：包含所有业务接口的实现



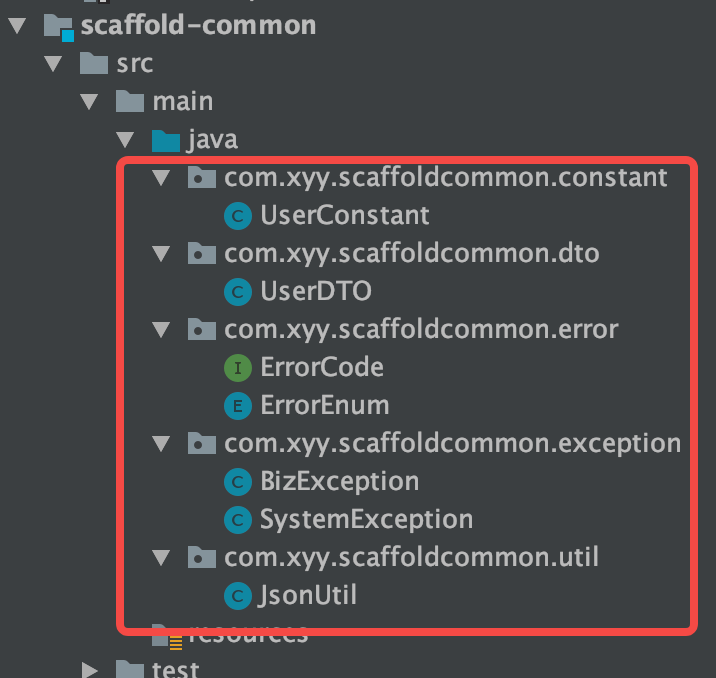
1. scaffold-dao模块

主要包含数据访问层内容，DO定义、DB访问层以及数据库相关配置类



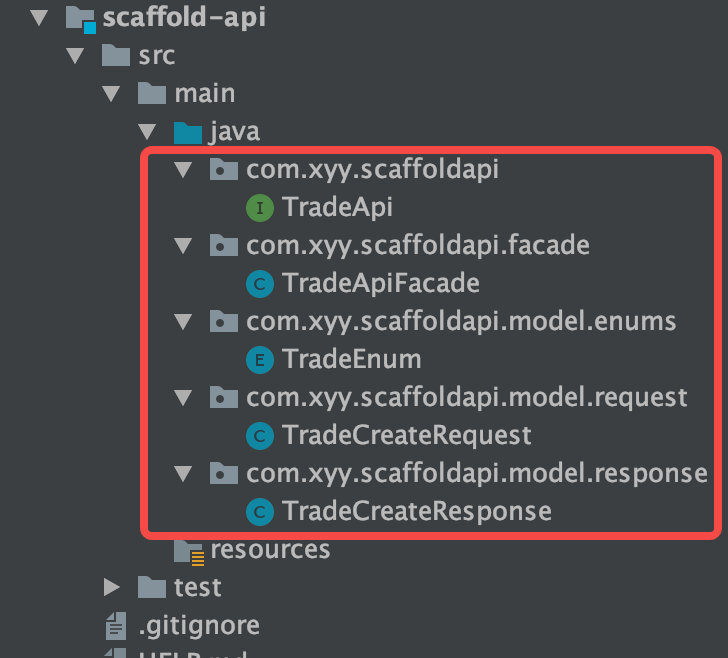
1. scaffold-common模块

包含每个模块可能都会用的一些基础类，如：dto、错误码、util工具类以及全局异常类和常量等



1. scaffold-api模块

主要负责外部依赖服务的管理，包含外部依赖服务的定义以及访问部分，以及通过facade模式做的封装类，如：request、response定义；外部服务api接口和外部服务通用枚举等



1. 整体服务的调用过程如下

**[该类型的内容暂不支持下载]**

**课件代码**

**[scaffold.zip]**

**详见附件**