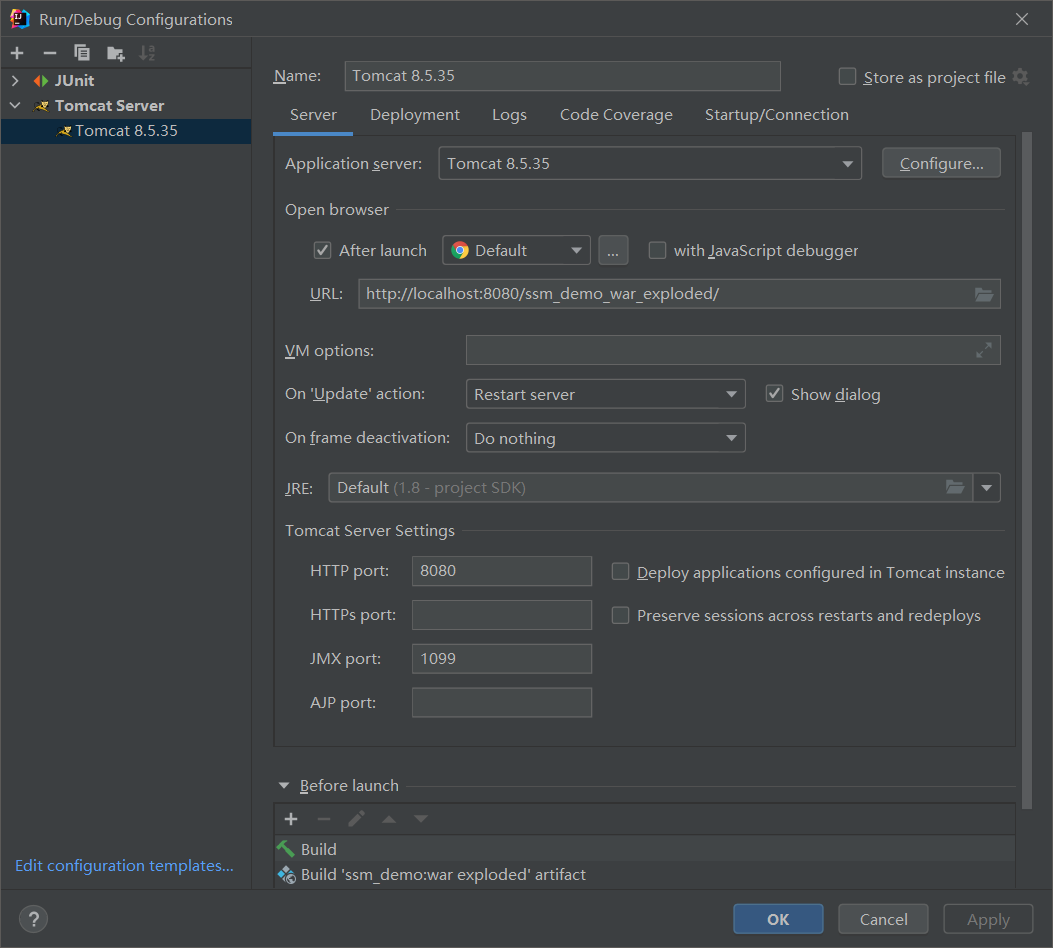
# ssm\_demo示例工程

## 工程概述

ssm\_demo工程是一个纯后端的简单示例工程，适合新手入门学习。SSM（Spring+SpringMVC+MyBatis）。

1. 项目结构

* controller包，代表和前端交互的接口。
* domain包，代表数据库表实体对象，也被称为DO(Data Object)或者entity。
* mapper包，代表实体对象与数据库表的映射，ORM(Object Relational Mapping)，包括增删改查操作。Interface提供给Java代码调用，xml文件将sql语句映射到Java代码接口。
* service包，代表后端业务逻辑，是controller包接口的具体实现类，通常也在service层调用DAO(Data Access Object)层接口，DAO层也就是mapper包。
* resources文件夹，ssm以及项目框架相关的配置文件，也可存放其他资源文件。
* webapp文件夹，web应用前端资源目录。

1. 使用说明
2. 安装MySQL数据库
3. 执行sql命令，CREATE DATABASE ssm\_demo DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci;
4. 执行sql脚本文件，这个文件在resources目录下。Mysql命令：source resources/ssm\_demo.sql，路径需要输入绝对路径（全路径）。
5. 修改jdbc.properties中的数据库连接信息，关键配置是url、username、password。
6. 安装tomcat并通过tomcat启动项目。
7. 通过postman或其他工具测试接口。

## Controller、Service、DAO三层详解

1. Controller层

代码源码文件说明：

1. @Controller是controller层的标识，不写这个注解则不会被spring认为是一个controller，也就是说与前端交互的接口类必须写这个注解。@RestController这个注解代表@Controller+@ResponseBody，使用@RestController时可以省略掉每个方法上的@ResponseBody。
2. @RequestMapping("/user")，是请求路径映射注解，写在类上代表这个类里的所有方法，都需要在/user这个前缀下，写在方法上代表这个方法的url地址，例如@RequestMapping("/add")代表添加user的接口，那么要调用这个接口的url就是/user/add，然后最终url为<http://ip:port/appname/user/add>。

@GetMapping("/user")仅支持get请求

@PostMapping("/user")仅支持post请求

@RequestMapping ("/user")支持get和post请求

1. @Autowired或@Resource代表注入一个bean对象，例如在UserController中注入了被@Service标识的UserService接口。如果没有@Service标识或者@Component标识则会注入bean对象失败。
2. @ResponseBody注解表示这个方法的返回值转为json格式，作为响应体返回给前端。

若写到类名上，代表这个类的所有方法都以json格式返回。

1. @RequestBody注解到方法参数前，代表这个参数以json格式接收，并解析为Java对象。每个方法参数中只能有一个@RequestBody。
2. @RequestParam("flowId")注解到方法参数前，代表以基础类型获取一个参数值，括号中的flowId代表前端传的参数名称。
3. DAO层

实体对象关系映射说明，

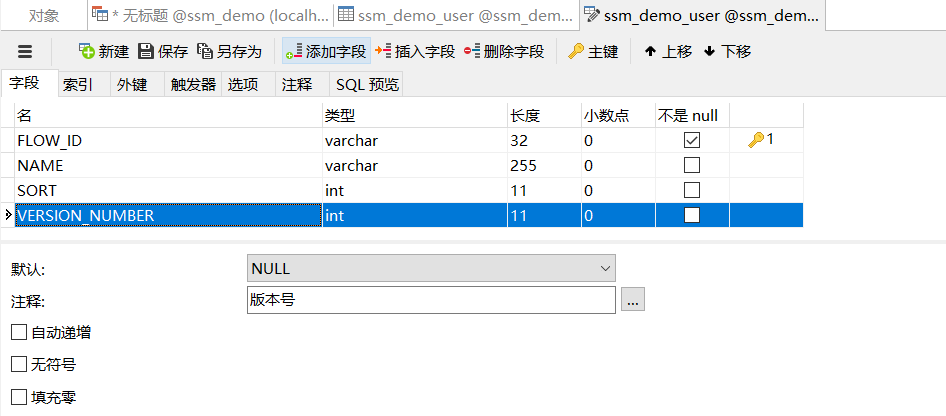
1. User.java就是一个简单的Java类，但他的每一个属性都对应到数据库表里的一个字段，一个类对应到一个表，一个属性对应一个字段，Java类型和数据库字段类型的对应关系到网上可以搜到。
2. UserMapper.java是对数据库表操作的接口声明，他是一个interface（接口），没有接口实现。声明接口的方法名、入参、回参。需要使用@Repository注解来标识接口。
3. UserMapper.xml相当于是UserMapper.java接口的SQL实现，遵守MyBatis的规则，其中的namespace="com.wy.demo.mapper.UserMapper"代表映射的Java类，xml标签<insert id="insert" parameterType="com.wy.demo.domain.User"></>中的id值代表Mapper接口的方法名，parameterType代表这个接口的入参类型，resultType代表这个接口的回参类型。`NAME`是因为NAME是MySQL的关键字，用``符号包起来才可以使用NAME关键字作为查询字段。#{flowId}代表入参中的属性名或实体类中的属性名。
4. Service层



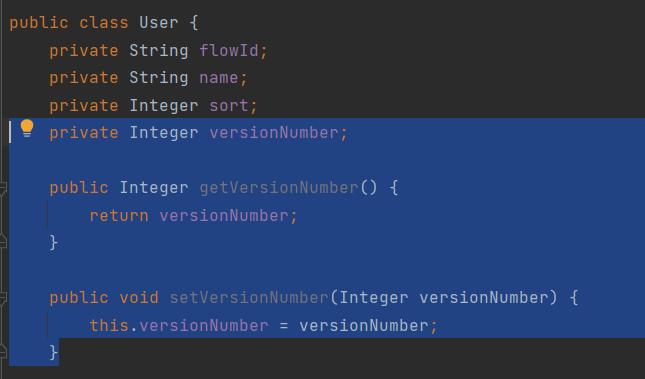
1. UserSevrvice是Java接口类，声明面向Controller层或者其他Service提供的接口。
2. UserServiceImpl是UserService接口的实现，implements UserService。必须实现UserService的所有接口，且通过@Service标识这个类。
3. @Service标识，标识这个类为Service层的实现类，当UserService被作为bean注入时，会自动找到他的实现类进行注入。
4. @Autowired注入一个bean对象，在Service层注入DAO层对象，Controller调用Service，Service调用DAO，业务逻辑就写在Service中。
5. @Override代表这个方法是实现UserSevrvice接口的方法，可以不写，但是写了就一定是UserSevrvice类的方法实现。而且UserSevrvice的所有方法都必须有实现。

## 数据库表修改步骤

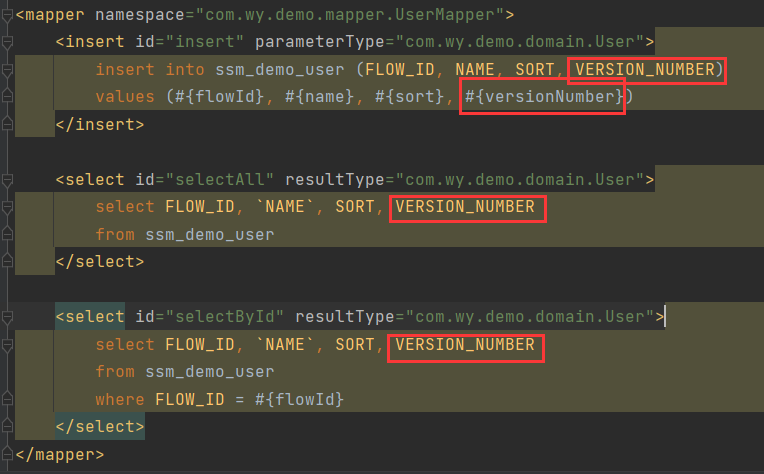
1. ssm\_demo\_user数据表添加VERSION\_NUMBER版本号字段，执行SQL命令ALTER TABLE `ssm\_demo\_user` ADD COLUMN `VERSION\_NUMBER` INT(11) DEFAULT NULL COMMENT '版本号';



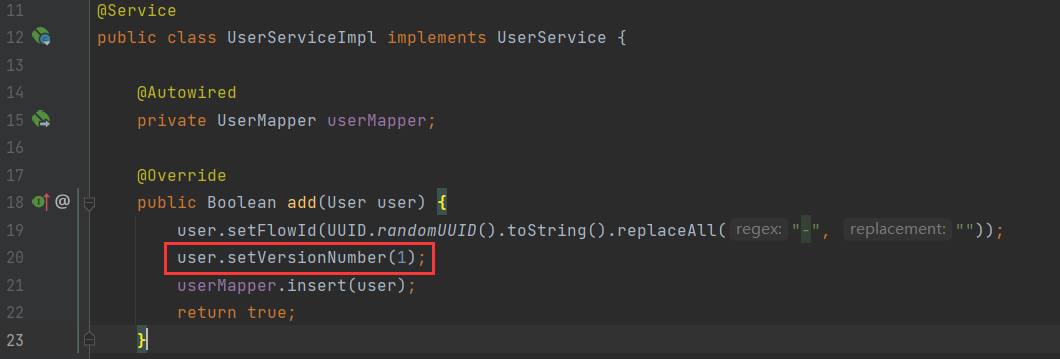
1. User实体类添加版本号属性，



1. UserMapper.xml修改SQL语句

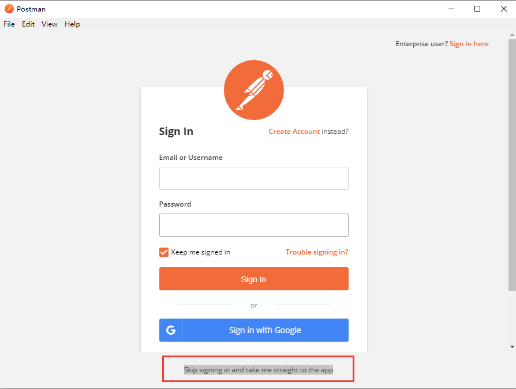


1. 修改业务代码UserServiceImpl，实现版本号字段的输入。至此新增版本号字段的增删改查已经修改完成，

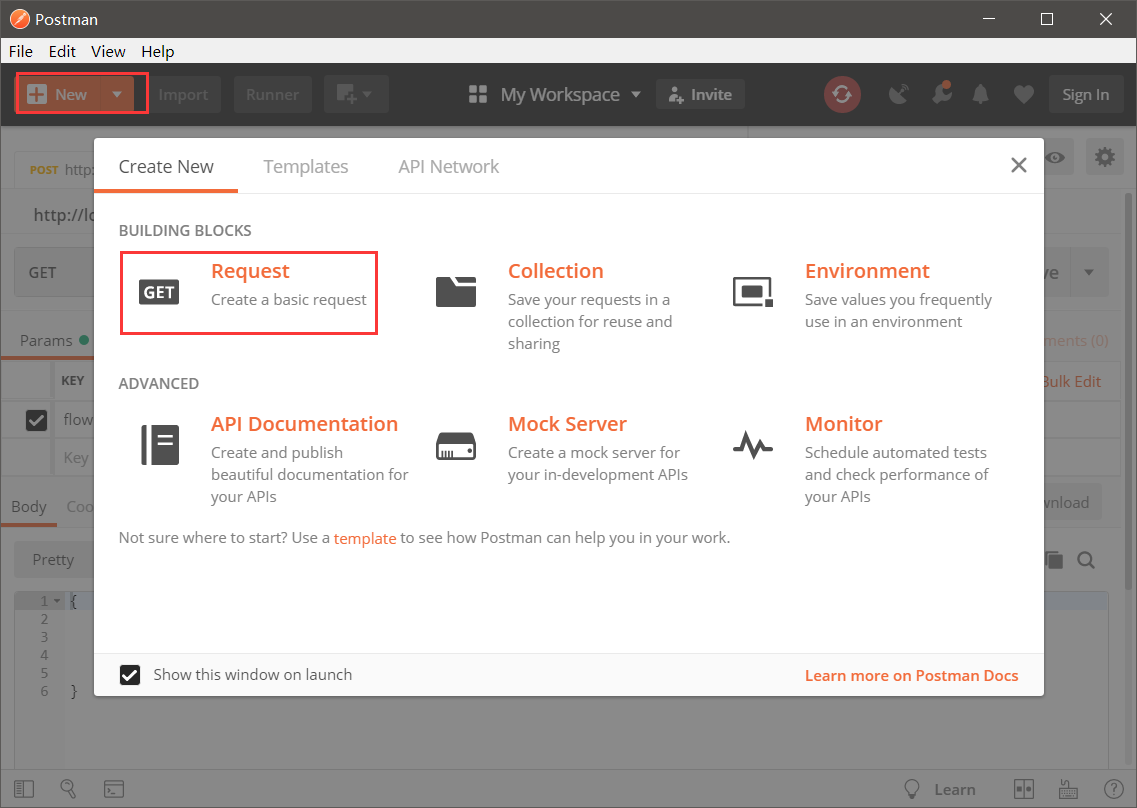


## Postman接口测试

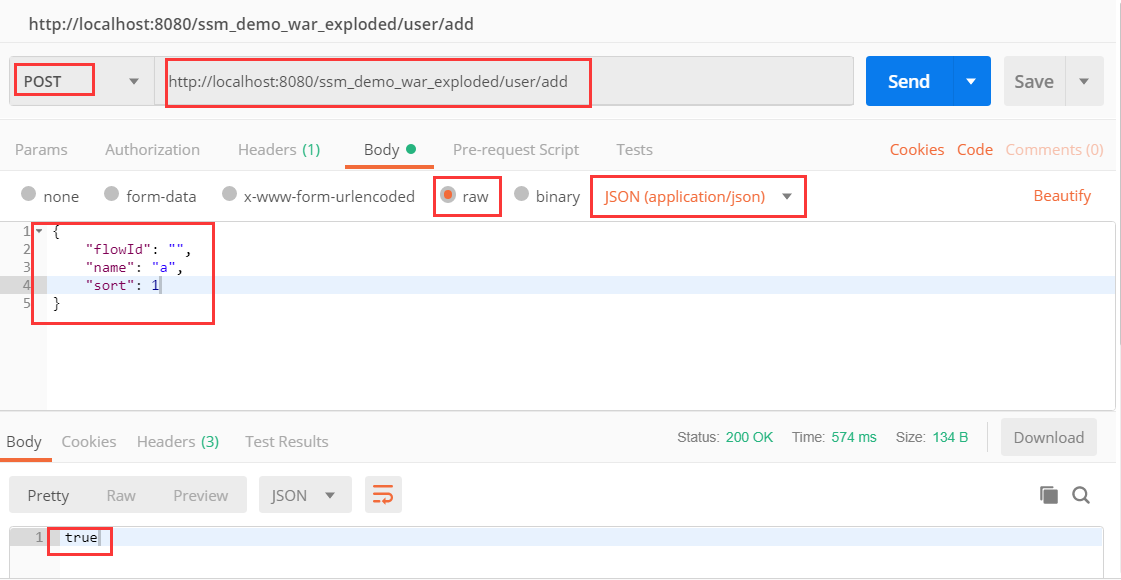
1. 安装postman，打开后选择中间最底部的”skip signing in and take me straight to the app”



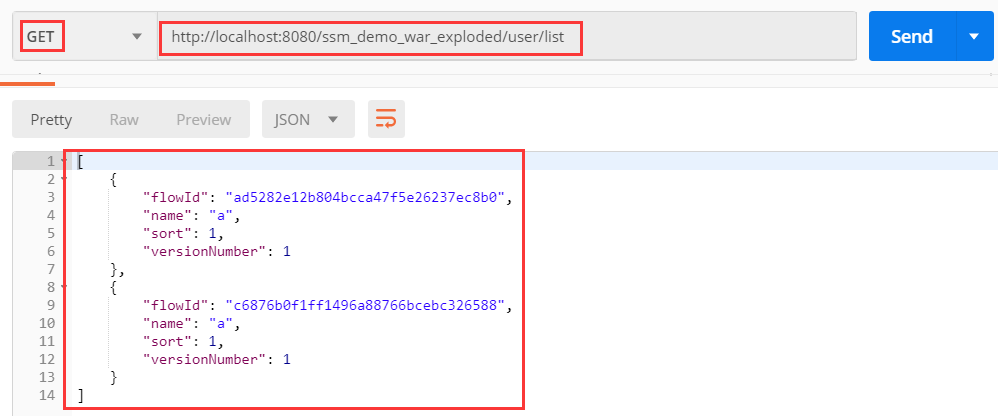
1. 点击postman界面上的new按钮，选择Request。



1. user/add接口



1. user/list接口



1. user/get

