



# 软件工程基础 第四次实验

后端入门基础——Spring Boot



# 本次实验将带大家从 零开始创建一个 SpringBoot后端项目



# 项目创建

# 通过官网生成： Spring Initializer 本地生成： Maven project

所有的依赖包都可以通过复制粘贴方式加入pom.xml，由IDE自动导入



# 项目结构



**SpringBoot项目大致分为四层：**

- controller**
- entity**
- mapper**
- service**

控制层接收前端请求，调用后端服务层，实现前后端的交互。

**@Controller**: 标记为控制类，处理HTTP请求

**@ResponseBody**: 将返回值打包为Json格式作为responsebody返回给前端

(@RestController= @Controller+ @ResponseBody)

**@RequestBody**: 标记参数来源

**@RequestMapping (PostMapping/GetMapping)** : 声明url和请求方式

**实体层装有项目所需的实体类，并与数据库表项相对应**

**@Entity：标记为实体类，对应数据库表**

**@Id：设置属性为主键，建表必不可少**

**@GeneratedValue：设置主键的自动生成方式，通过参数strategy控制**

(实体层的文件路径需要在配置文件中标记)



**数据层相应服务层的调用，利用数据库语句对数据库直接进行增删改查**

**@mapper: 标记为数据类，告诉SpringBoot这是一个数据层接口**

**\*.xml: 数据库操作语句的集合，做数据库操作，其中需要配置数据层类的目录，并且要在配置文件中声明所有xml的路径**

**服务层相应控制层调用，如果需要则调用数据层，本质为后端逻辑。**

**@Service：标记为服务层接口实体**

**需要提供接口类和实体类两类。**

配置文件处在resources文件夹下，有.properties和.yml两种后缀，除格式外无本质区别。

基本配置文件格式举例（.yml）：

```
spring:
  datasource:
    url: jdbc:mysql://127.0.0.1:3306/lab
    username: root
    password:
  jpa:
    hibernate:
      ddl-auto: update
mybatis:
  mapper-locations: classpath:mapper/*.xml
  type-aliases-package: com.example.demo.Entity
server:
  port: 8090
```



**项目启动器位于四层结构同一目录下（[项目名称]Application.java），本质为一个main函数作为启动项，需要使用@SpringBootApplication**



# session处理办法



# 1.HttpSession

使用：

```
HttpServletRequest request = ((ServletRequestAttributes)  
RequestContextHolder.getRequestAttributes()).getRequest();
```

来获取当前HTTP请求

之后使用：

```
HttpSession session = request.getSession();  
session.setAttribute("xxx", xxx);
```

来设置session中的值

(也可以将controller的参数直接写为xxxController(HttpServletRequest request)直接获取当前请求)



## 2.加密后放入body中前后 端传递



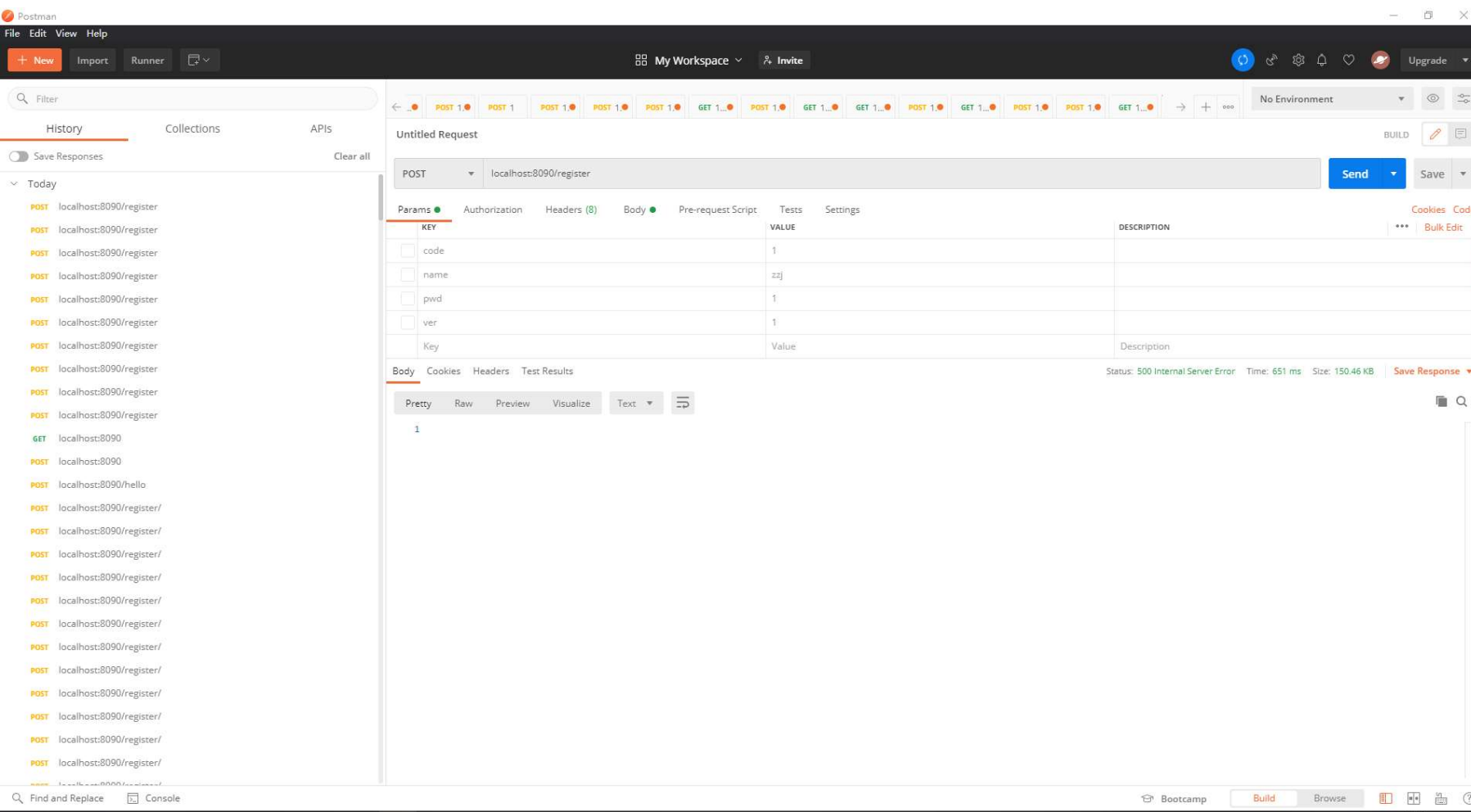


# 相关应用推荐



# 1. 后端测试工具

# Postman

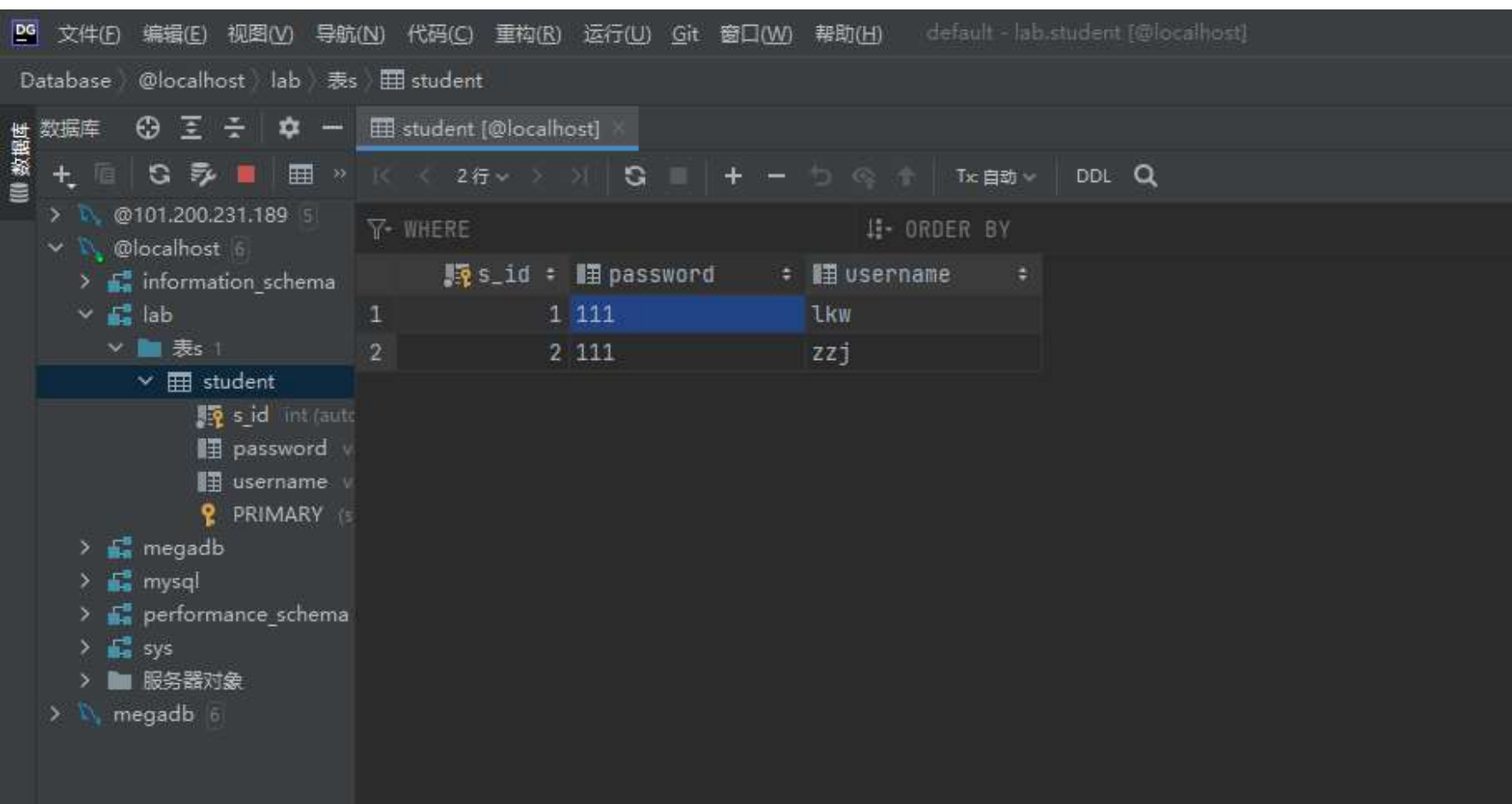


可在没有前端的情况下测试后端API，模拟HTML请求，前后端分离开发的必备利器！



## 2.数据库可视化工具

# Datagrip



处理数据库相当方便，不用牢记所有数据库指令，界面美观（个人推荐）



谢谢