Vue3没啥好说的,感觉就是注意下比vue2多的语法糖,然后组件间通信即可。

归功于那个教程讲的非常好：[开始使用 Vue - 学习 Web 开发 | MDN](https://developer.mozilla.org/zh-CN/docs/Learn_web_development/Core/Frameworks_libraries/Vue_getting_started)

首先navicat可以用datagrip平替。

接着讲下这个mybatis框架的毒点以及开发的坑点。[Mybatis 框架快速入门（超详细）\_mybatis csdn-CSDN博客](https://blog.csdn.net/weixin_43883917/article/details/113731380" \l "MyBatis__1) 这个比老师推荐的好多了。

1.自己添加依赖项：mysql-connector的驱动可能会出现无法定位,那就自己添加jar包。形如：

C:\Users\周锦耀\.m2\repository\com\mysql\mysql-connector-j\8.3.0\mysql-connector-j-8.3.0.jar。

在 文件>项目结构>模块>添加 即可。

1. 关键代码：

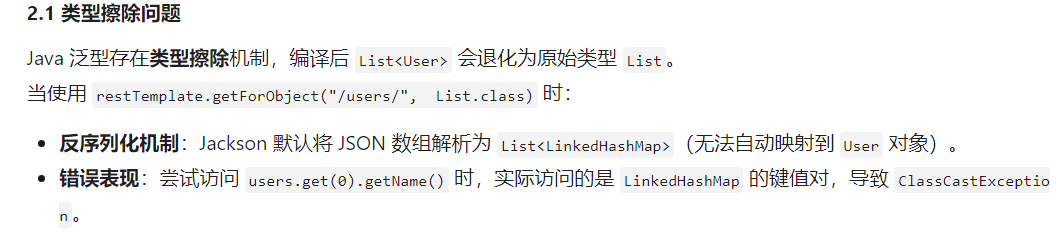
//1.读取配置文件  
InputStream in = Resources.*getResourceAsStream*("SqlMapConfig.xml");  
//2.创建SqlSessionFactory工厂  
SqlSessionFactoryBuilder builder = new SqlSessionFactoryBuilder();  
SqlSessionFactory factory = builder.build(in);  
//3.使用工厂生产SqlSession对象  
SqlSession session = factory.openSession();  
//4.使用SqlSession创建Dao接口的代理对象  
UserMapper userMP = session.getMapper(UserMapper.class);

动态代理机制：自动为接口生成类，当调用 getMapper() 方法时，MyBatis 会根据 UserMapper 接口的方法签名和映射文件（XML 或注解），动态创建一个代理对象。这个代理对象负责将接口方法调用转化为具体的 SQL 操作（如执行 select、insert 等）。

1. 这个测试类很有意思：

void testGetUserListViaHttp() {  
 // 发送 GET 请求并验证响应  
 ResponseEntity<List<User>> response = restTemplate.exchange(  
 "/users/", // URL  
 HttpMethod.*GET*, // HTTP方法  
 null, // 请求体（GET请求可传null）  
 new ParameterizedTypeReference<List<User>>() {} // 泛型类型包装  
 );  
 List<User> users = response.getBody();  
 *assertFalse*(users.isEmpty(), "返回的用户列表不应为空");  
 *assertTrue*(users.get(0).getName().equals("王明"), "应包含预设用户王明");  
}

之前的错误部分是直接用List<User> users = restTemplate.getForObject("/users/", List.class)；得到的是直接将一大团反序列化为一个哈希值，所以有问题，必须先得到response，再从里边得到主体内容。



还有一点就是用post提交的时候，不是传json文件！！就是像平常一样传类即可。见代码

public String postUser(@RequestBody User user) throws IOException

1. 配置方面：配置SqlMapConfig.xml(启动数据库用),xxx.xml：为接口提供动态代理机制，简化代码，是mybatis框架的很有意思的一个功能。
2. 第二次作业位于demo文件夹下