智能bi开发文档

### 需求分析

1. 智能分析：用户输入目标和原始数据，可以自动生成图表和分析结论
2. 图表的管理：增删改查
3. 图表生成的异步化
4. 对接AI的能力

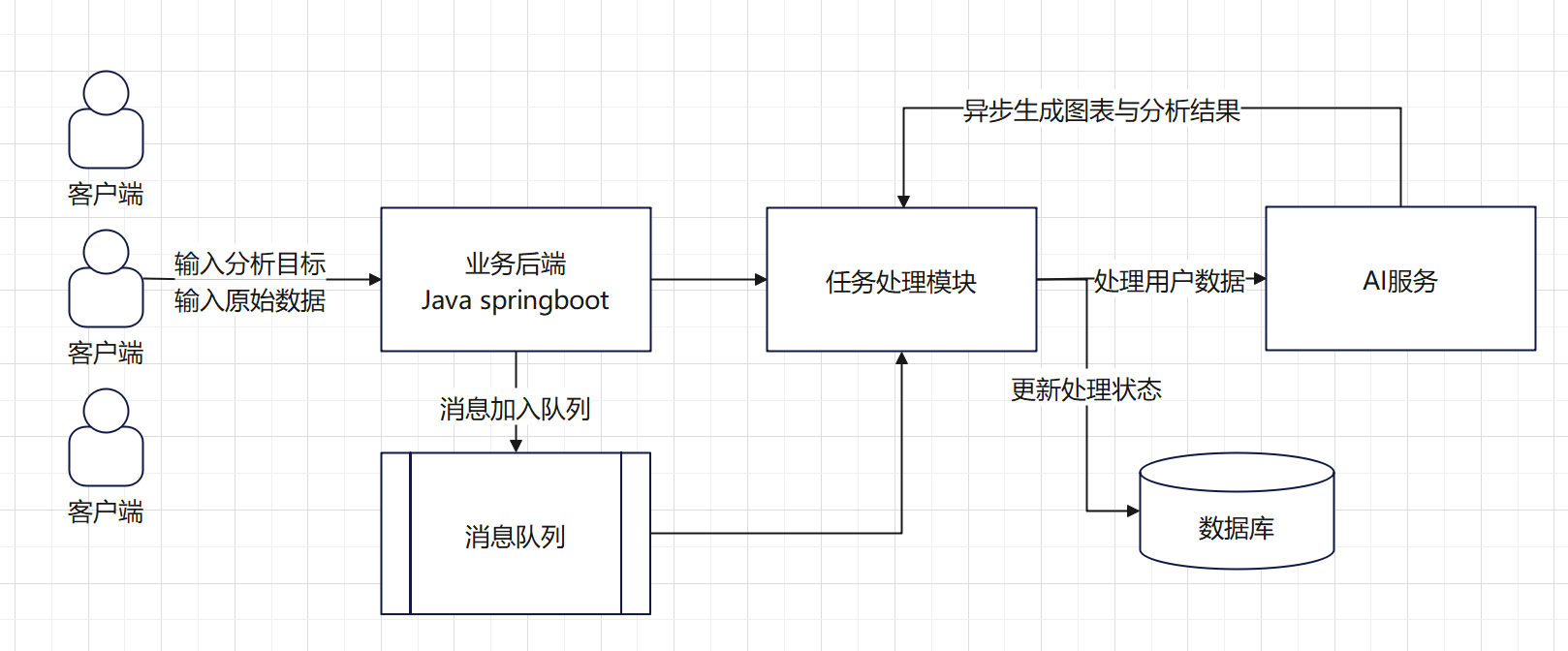
### 架构分析

基础方案：客户输入**分析目标**和**原始数据**，向业务后端发送请求。业务后端利用AI服务处理客户端数据，**保存到数据库**，并生成**图表**和**分析结果**。处理后的数据由业务后端发送给AI服务，AI服务生成结果并返回给后端，最终将结果返回客户端展示。

问题：AI服务（ChatGPT）回复需要时间且分钟内需求个数有限制，用户发送消息速率与AI服务接受速率存在同步场景。

改进方案：**增加消息队列，实现异步化**

客户输入**分析目标**和**原始数据**，向业务后端发送请求。业务后端将请求事件放入消息队列，并为客户端生成排队号，消息队列根据AI服务负载情况，定期检查进度。如AI服务能处理更多请求，就向任务模块发送消息。



### 开发流程

#### 准备工作

**前端开发**

安装nodejs

<https://blog.csdn.net/WHF__/article/details/129362462>

安装webstrom/vscode(我内存容易炸所以选vscode)

<https://www.jb51.net/softs/598714.html>

<https://blog.csdn.net/qq_44757223/article/details/122676919>

**后端开发**

安装idea

https://www.exception.site/essay/how-to-free-use-idea-202021-by-resigter-code

安装java

<https://www.exception.site/essay/how-to-free-use-idea-202021-by-resigter-code>

安装maven

<https://www.cnblogs.com/jywang/p/15828707.html>

**其他工具**

Mysql

<https://www.cnblogs.com/itcui/p/15511683.html>

git

<https://blog.csdn.net/weixin_47638941/article/details/120632890>

yarn

cmd安装yarn

npm install -g yarn

#### 前端项目初始化

使用Ant design pro搭建前端框架

1、按照官方文档初始化

参考官方文档：<https://pro.ant.design/zh-CN/docs/getting-started/>

初始化脚手架

# 使用 npm

npm i @ant-design/pro-cli -g

pro create bi-frontend

选择umi4

2、项目试运行（npm run dev/start）

试运行package.json

"dev": "npm run start:dev",

yarn run dev

报错原因，未安装依赖

yarn init

试运行成功，按默认账号登录

admin

ant.design

3、代码托管

Git init

Git add .

Git commit -m ‘first commit’

4、移除不必要能力（比如国际化）

运行package.json

"i18n-remove": "pro i18n-remove --locale=zh-CN --write",

报错解决：

找到官方项目地址

<https://github.com/ant-design/ant-design-pro/issues/10452>

执行 yarn add eslint-config-prettier --dev yarn add eslint-plugin-unicorn --dev

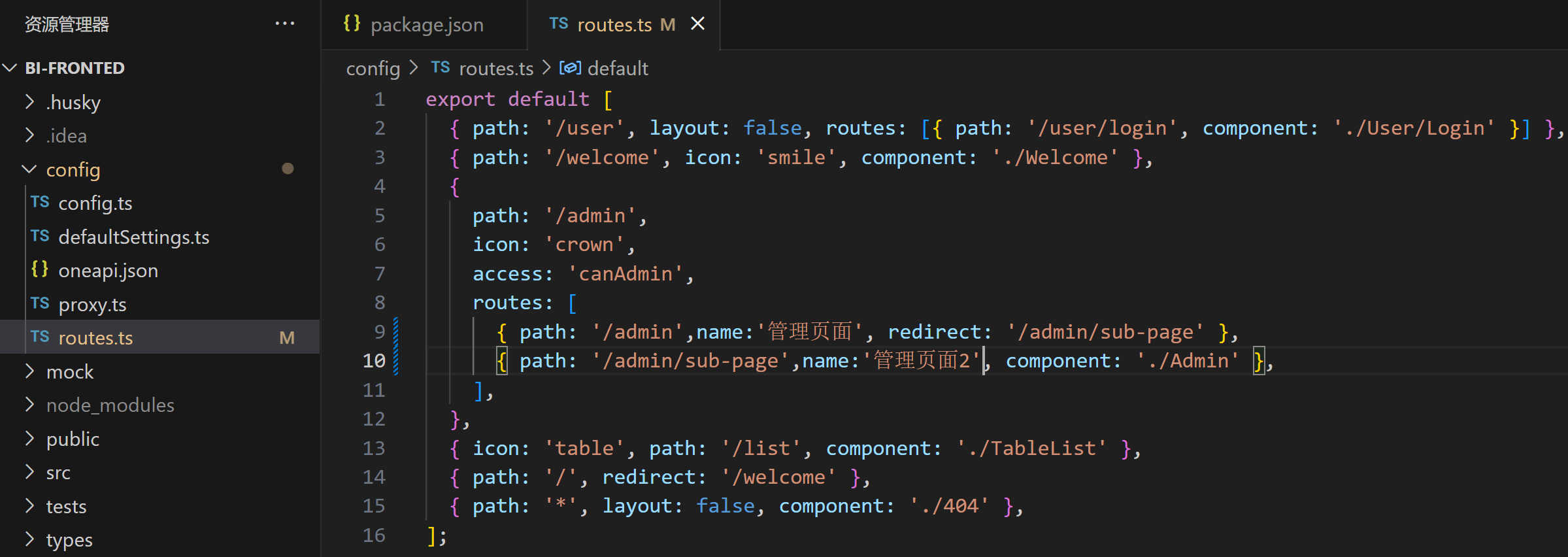
然后修改node\_modules/@umijs/lint/dist/config/eslint/index.js文件注释

// es2022: true可以解决问题

1. 试运行

登录后不显示菜单

Config/routes.ts routes下添加目录name:’管理页面’



#### 后端项目初始化

Idea基于后端模板修改

1、初始化数据库

Database配置mysql

并修改配置

src/main/resources/application.yml

运行建表

sql/create\_table.sql

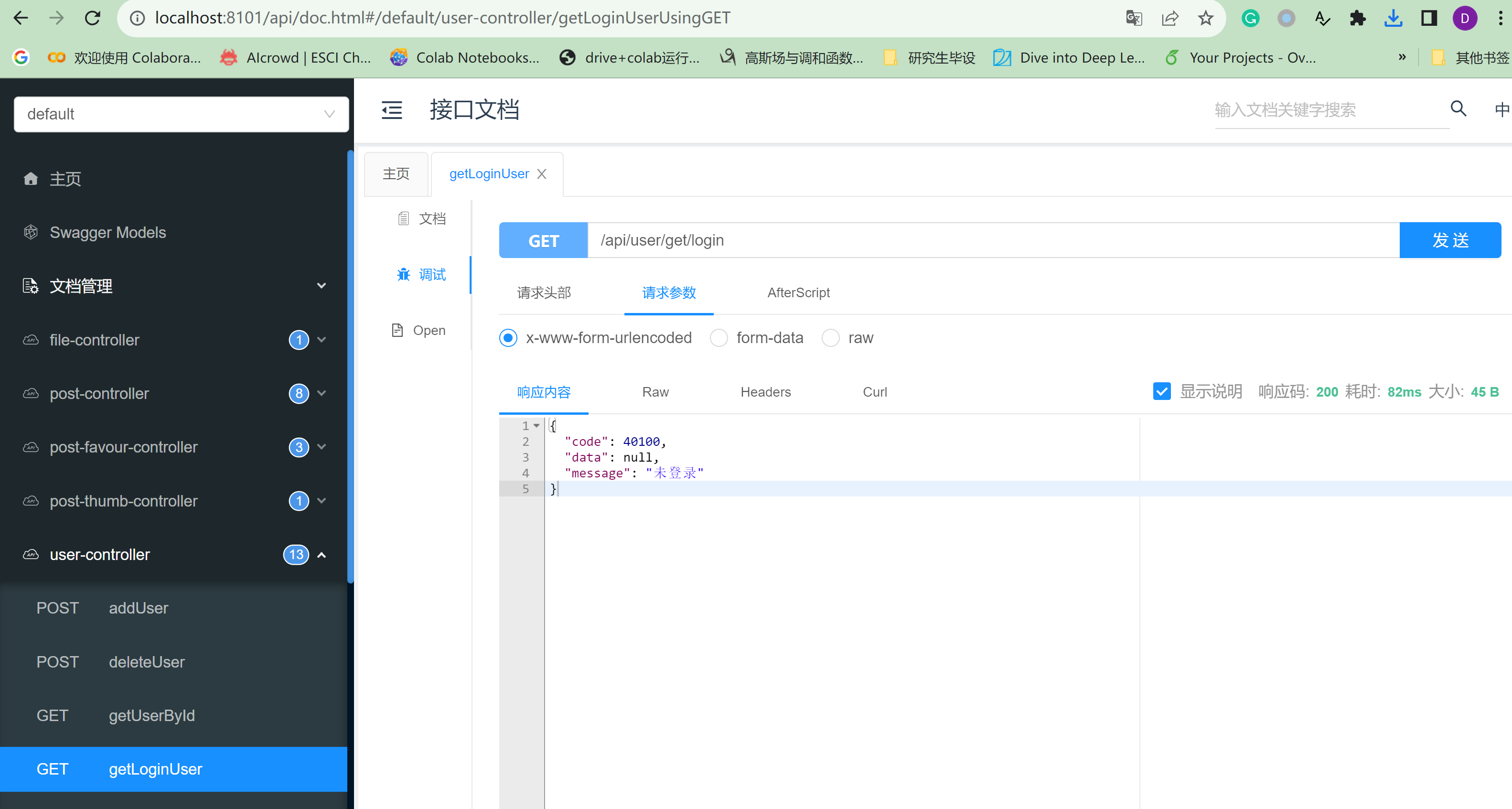
2、试运行

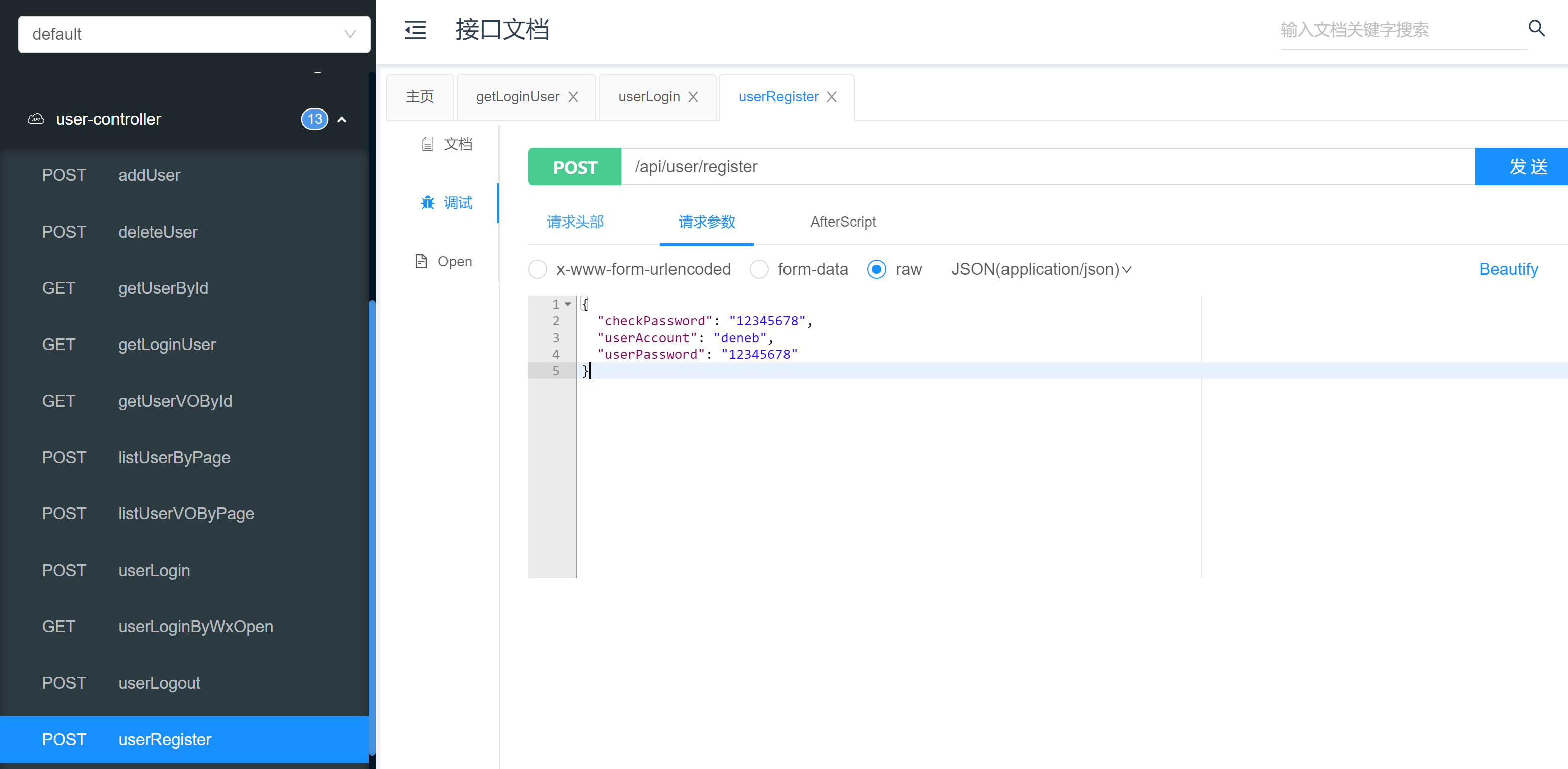
src/main/java/com/yupi/springbootinit/MainApplication.java

浏览器测试，端口号为src/main/resources/application.yml server[‘port’]

<http://localhost:8101/api/doc.html>

注册用户与登录





#### 库表设计

--保存用户信息和图表信息

-- 创建库

create database if not exists bidb;

-- 切换库

use bidb;

-- 用户表

create table if not exists user

(

id bigint auto\_increment comment 'id' primary key,

userAccount varchar(256) not null comment '账号',

userPassword varchar(512) not null comment '密码',

userName varchar(256) null comment '用户昵称',

userAvatar varchar(1024) null comment '用户头像',

userRole varchar(256) default 'user' not null comment '用户角色：user/admin',

createTime datetime default CURRENT\_TIMESTAMP not null comment '创建时间',

updateTime datetime default CURRENT\_TIMESTAMP not null on update CURRENT\_TIMESTAMP comment '更新时间',

isDelete tinyint default 0 not null comment '是否删除',

index idx\_userAccount (userAccount)

) comment '用户' collate = utf8mb4\_unicode\_ci;

-- 图表信息表

create table if not exists chart

(

id bigint auto\_increment comment 'id' primary key,

goal text null comment '分析目标',

`name` varchar(128) null comment '图表名称',

chartData text null comment '图表数据',

chartType varchar(128) null comment '图表类型',

genChart text null comment '生成的图表数据',

genResult text null comment '生成的分析结论',

status varchar(128) not null default 'wait' comment 'wait,running,succeed,failed',

execMessage text null comment '执行信息',

userId bigint null comment '创建用户 id',

createTime datetime default CURRENT\_TIMESTAMP not null comment '创建时间',

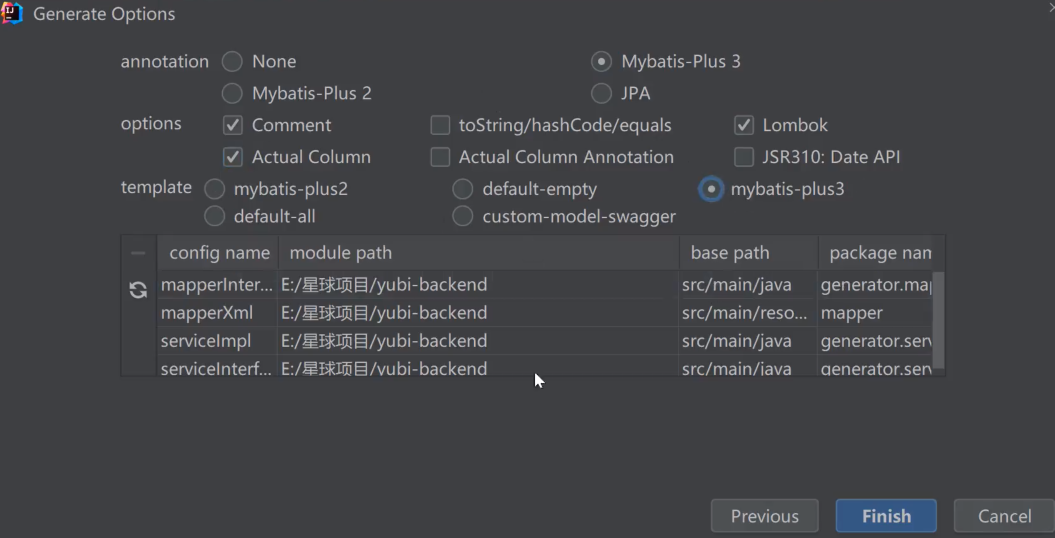
updateTime datetime default CURRENT\_TIMESTAMP not null on update CURRENT\_TIMESTAMP comment '更新时间',

isDelete tinyint default 0 not null comment '是否删除'

) comment '图表信息表' collate = utf8mb4\_unicode\_ci;

#### 数据层实现MyBatis

MyBatisX插件



ID类型为ASSIGN\_ID

（雪花算法生成唯一id，长整型）

@TableLogic

标识逻辑删除字段

删除无用功能

后端测试

<http://localhost:8101/api/doc.html>

#### 控制层模型层适配

1、从controller看缺chart信息

2、model.dto补充对应表的文件

AddRequest增加图表

EditRequest编辑已经生成的图表

QueryRequest查询已有图表（通过id，目标，表类型，用户id查）

UpdateRequest

3、从entity中copy表字段到model.dto，删除功能不需要的字段

4、删除控制器中不需要的功能（校验、转译、填充默认值）

/add

List<String> tags = chartAddRequest.getTags();

if (tags != null) {

chart.setTags(GSON.toJson(tags));

}

chartService.validChart(chart, true);

填充空值

chart.setUserId(loginUser.getId());

chart.setFavourNum(0);

chart.setThumbNum(0);

/delete

参数校验，判断是否存在，软删除

/update

/get查

分页获取列表（封装类）

分页获取当前用户创建的资源列表

修改图表（仅当前用户和管理员）

补充功能：根据前端生成支持mabaties查询条件

chartService.getQueryWrapper

参考service.impl postServiceImpl get QueryWrapper

将帖子用的查询参数换成图表用的

#### 前端生成增删改查代码

Config.ts 中openAPI

修改后端地址为

schemaPath:”<http://localhost:8101/api/v2/api-docs>;”

projectName:

package.json 运行openapi，根据controller生成swagger文件（前端调用后端代码的文件）

src/service/deneb/controller.ts

app.tsx修改request.baseURL