后端:不需要用数据库,所以不用Pydantic校验了,只剩下两点:

1) 在`app/controllers/`下增加模块操作文件，主要是供api中调用

2) 注册路由，`app/api/v1/`下新增api文件夹及对应文件(比如xxx模块,就建立xxx文件夹,里面放xxx.py和\_\_init\_\_.py)，并相应修改`app/api/v1/\_\_init\_\_.py`注册路由，这里使用了fastapi

3) 在`app/core/init\_app.py`中增加初始化操作 (非必要操作,暂时不用)

前端:

1) 在`web/src/api/index.js`中增加api的接口定义

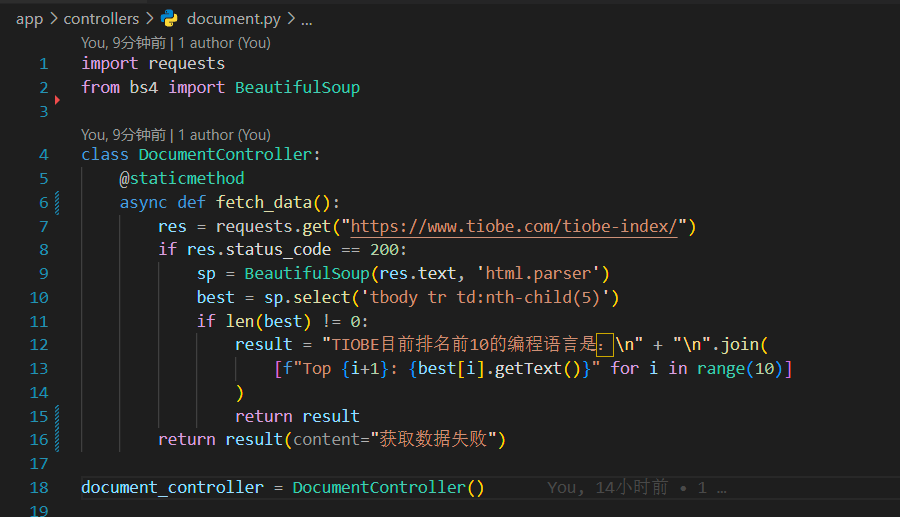
2) 在`web/src/router/routes/index.js`中新增页面路由

3) 在`web/src/views`中合适位置新增页面即可

Demo流程:

后端1) 在`app/controllers/`下增加模块操作文件，主要是供api中调用

示例: app/controllers/document.py 这里主要是爬虫逻辑处理



后端2) 注册路由，`app/api/v1/`下新增api文件夹及对应文件(比如xxx模块,就建立xxx文件夹,里面放xxx.py和\_\_init\_\_.py)，并相应修改`app/api/v1/\_\_init\_\_.py`注册路由

图形用户界面

中度可信度描述已自动生成

内部文件如下:  
文本

描述已自动生成

文本

描述已自动生成  
注意:这里的返回数据要加在return Success中;

前端1) 在`web/src/api/index.js`中增加api的接口定义

图形用户界面, 文本

描述已自动生成

前端2) 在`web/src/router/routes/index.js`中新增页面路由;  
这里我是添加了所有的”数据收集”下的所有子菜单内容;

文本

描述已自动生成

其中component字段为:点击子菜单后显示什么vue界面;

3) 在`web/src/views`中合适位置新增页面即可

即页面的显示内容,这个在对应文件夹下 的/views/data-collection/paper/index.vue'中;

文本

描述已自动生成

Data-collection/document/index.vue界面:点击按钮即可输出结果;  
文本

中度可信度描述已自动生成