

README

基于Django和Vue的前后端分离式问卷调查系统

技术选型

1. 设计模式（整体架构模式）： 前后端分离（前端提供页面渲染、后端只提供接口）
2. 前端框架： Vue2.0（使用NPM管理）， NPM基于Node.js V10.15.1
3. 后端框架： Django（基于Python）， Django2.2.12
4. 数据库选型： MySQL发行版->MariaDB
5. 后端架构： MVC（Model-View-Controller）， View视图层改用前后端分离，使用Vue单独编写

前后端分离： 前端只提供界面渲染（用户视图）， 接口实现交由后端

后端提供API接口， 以接口形式封装发送数据， 提供功能实现

前后端分离式架构是一种软件架构模式，用于构建Web应用程序。它的核心思想是将前端（即用户界面）和后端（即数据处理和业务逻辑）分开开发和部署。

在传统的Web应用程序中，前端和后端通常紧密耦合，即前端和后端代码混合在一起。这种耦合使得应用程序的开发和维护变得复杂，难以扩展和重用。而采用前后端分离式架构，前端和后端可以独立开发和演化，彼此之间通过API进行通信。

在前后端分离式架构中，前端通常是一个单页面应用（SPA），由HTML、CSS和JavaScript构建的用户界面。前端通过HTTP请求向后端发送数据，并使用后端提供的API进行数据交互和业务逻辑处理。

后端是一个独立的服务，负责处理数据持久化、业务逻辑和安全性等方面。后端可以使用任何编程语言或框架来实现，常见的选择有Node.js、Java、Python等。后端通过API暴露数据和功能给前端，前端通过API调用实现与后端的通信。

前后端分离式架构的优势包括：

- 提高开发效率：前后端可以并行开发，减少开发时间。
- 提升用户体验：前端可以采用现代化的技术和框架，提供流畅、响应迅速的用户界面。
- 支持多平台：通过提供API，可以轻松支持不同平台（如Web、移动设备等）的客户端。
- 可扩展性：前后端解耦，可以独立扩展前端和后端的功能和性能。

详细设计

前端设计

Vue2（集成Webpack、EChart等）

前端：HTML+CSS+JavaScript

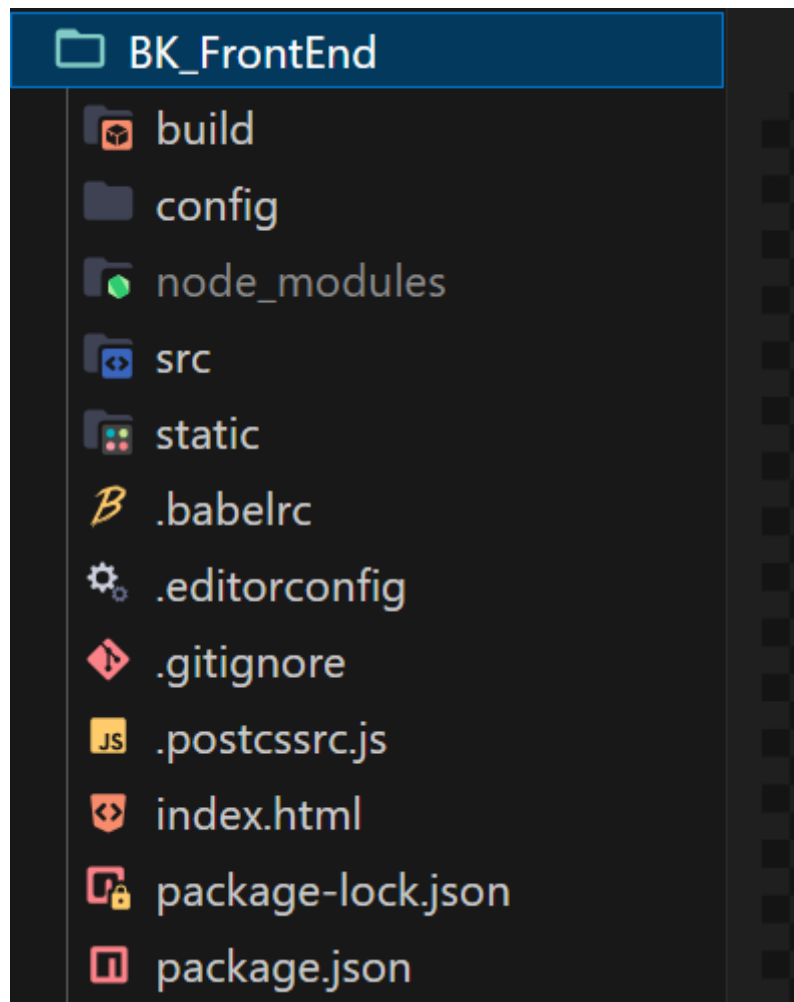
基本三件套->前端框架（BootStrap、Uniapp、Vue）

Vue->组件式、多渲染的前端框架，基于Webpack（迅速的将前端环境打包部署），每一个网站都是一个.vue的组件

xxx.html 一个网页

在Vue中就是一个xxx.vue

前端代码结构如下：



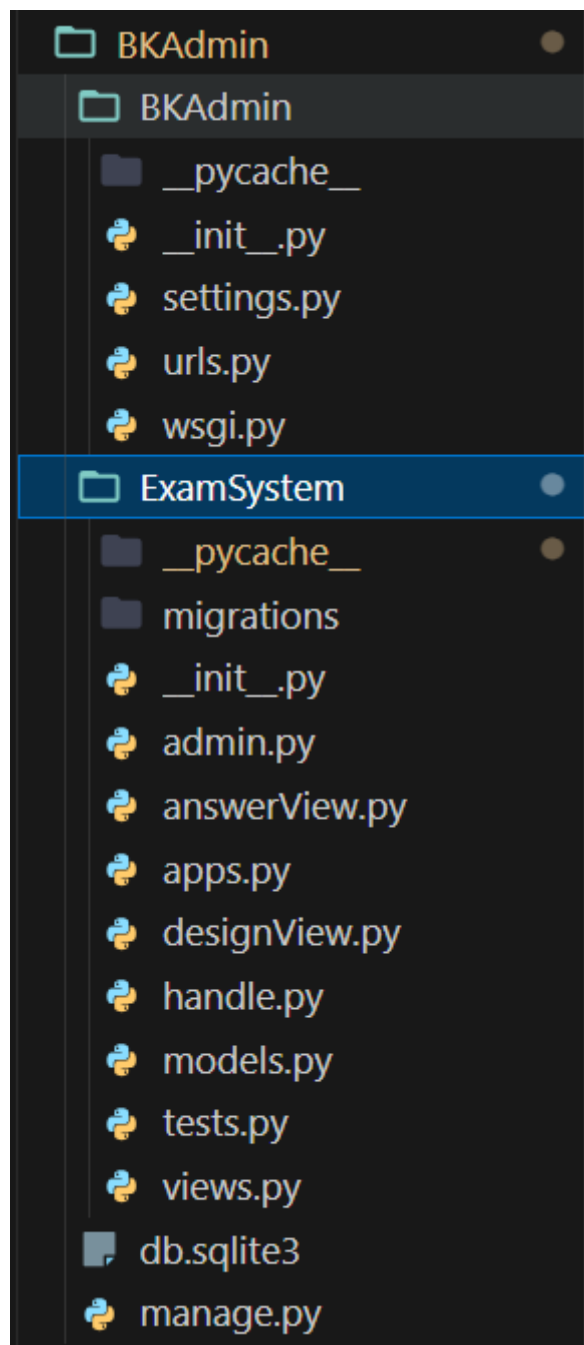
- BK_FrontEnd 前端项目总目录
 - build文件夹：存储编译后的二进制文件
 - config文件夹：项目的配置信息
 - node_modules文件夹：node包，项目所涉及的工具组件，如图表、Webpack（用于容器化集成）
 - src文件夹：源文件夹，用于存储源文件
 - assets：资产目录，存储部分图片，这里存储了logo图标
 - components：组件目录 存储核心网页文件
以index.vue为例->网站首页的代码
api.js 核心！ 连接后端提供的接口，打通前后端
 - router：路由配置 用于配置路由URL，即网页地址
 - App.vue 全局配置文件 此处没有什么大用
 - md5.js 加密算法 密码pwd加密用

- static文件夹：用于存储静态文件，比如静态CSS、HTML，以及部分图片等
- package.json 包json文件 和node_modules呼应

后端设计

Django2.2.12+Celery（事务异步处理-分布式技术）+Simple-ui2022+MySQL发行版（Maria DB）

Django是一个流行的Python-Web框架，提供了许多的中间件和便携方法，Django以app划分各个子应用，且自带一套后台管理插件Admin，Django默认的数据库是SQLite，此次使用的数据库是MySQL发行版-MariaDB，项目托管在Github上。



- BKAdmin-项目名称
 - BKAdmin 和项目名称同名的APP 用于配置项目信息
 - `__init__.py`: 用于初始化项目
 - `settings.py`: 项目全局配置文件 主要配置数据库连接、中间件、注册APP等信息
 - `urls.py`: 用于配置项目路由, 即MVC中的C (Controller), 此处由于架构为前后端分离 项目路由 由前端项目中的Vue中的router.js控制
 - ExamSystem 问卷系统 核心子应用
 - `migrations` 文件夹-数据库迁移文件
 - `__init__.py`: 做应用初始化

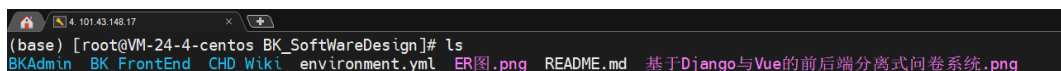
- `admin.py`：配置后台管理界面相关显示信息
- `answerView.py`：回答问卷请求接口，回答问卷后台处理函数API
- `apps.py`：配置该子APP信息，即ExamSystem的信息
- `designView.py`：设计问卷请求接口，主要实现设计问卷相关的API
- `handle.py`：实现数据分析和数据处理、生成图表等功能API
- `models.py`：数据模型层，核心数据库模型实现
- `tests.py`：系统测试文件，该处未涉及
- `views.py`：Django自带的视图模型层文件，由于本项目是前后端分离架构，所以无需编写Views.py文件

项目部署方法

1. 克隆项目到本地，项目开发环境为Centos7 Linux发行版

```
git clone
```

```
https://github.com/EASYGOING45/Tenet_SoftwareDesign.git
```



2. 运行前端项目

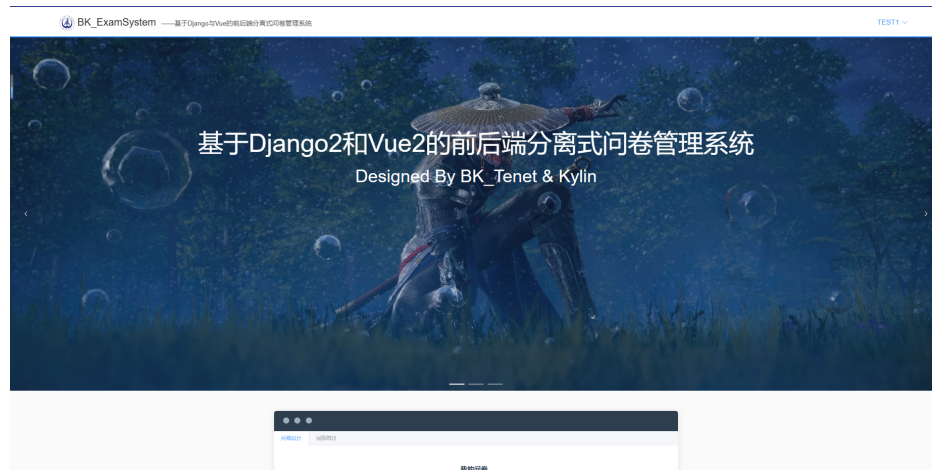
1. 切换路径至前端项目路径（以开发环境为例） `cd`

```
BK_FrontEnd
```

2. 安装前端依赖 `npm install`

3. 运行前端开发环境 `npm run dev`

4. 访问本机8080端口查看是否成功



3. 运行后端项目

1. 切换路径至后端项目路径（以开发环境为例）
2. 切换Python解释器至开发环境 使用conda环境
3. 迁移数据库

```
python manage.py makemigrations
```

```
python manage.py migrate
```

4. 启动后端服务器

```
python manage.py runserver
```