## 一、项目开发介绍:

三维数字铁路网络应用系统，前端使用HTML5， javascript， css和jQuery开发。后端使用python flask框架开发, 开发使用的python版本为3.7.3（3.5+均可）。

## 二、文件结构：

1. xls\_digital\_railway/ 项目根目录：

.git/：git仓库目录。

.gitignore: 内含不纳入git版本控制的文件和文件夹。

app/：项目的源文件。

logs/: 服务器运行日志文件

server.py: web服务器的启动文件。

demon.py: 守护进程脚本，可执行定时检查、查询、终止进程等操作。

config.py: web服务器配置文件, 主要用于设置服务器的SECRET\_KEY, 数据库连接的配置等。

requirements.txt: 项目所需的python依赖包。

tsconfig.json: 编写projects相关的js文件时, 用到了typescript语言, 此文件为vscode自动编译ts文件成js文件的配置文件。

1. xls\_digital\_railway/app/ 项目源文件目录

2.1 main/: 主页相关路由[[1]](#footnote-1)和视图函数[[2]](#footnote-2)。

\_\_init\_\_.py: 将目录变为python包, 便于其他文件使用import导入。

errors.py: 错误页面路由和视图函数。

views.py: 首页、风采展示、视频页面、视频搜索页面等的路由和视图函数。

2.2 auth/: 登录注册相关路由和视图函数。

views.py: 登录、注销、注册、用户信息页面、用户资料修改等功能的路由和视图函数。

2.3 projects/: 项目页面相关路由和视图函数。

views.py: 项目列表页面(/projects), 项目浏览页面(/projects/explore?pid=<id>), 处理地面线、横断面文件上传的路由和视图函数。

sidebar\_menu.py: ITEMS变量储存了项目浏览页面右侧弹出菜单的按钮图片名、id的全部数据。project\_explore函数会根据此变量渲染html页面。

2.4 admin/: 管理员页面相关路由和视图函数。

views.py: 管理员页面渲染、数据修改的路由和视图函数。

2.5 static/: 所有静态文件均储存在此处。更多信息见 3. xls\_digital\_railway/app/static

2.6 templates/: 所有html模板文件均储存在此处。更多信息见4. 。

2.7 \_\_init\_\_.py: 进行服务器初始化工作, 指定了各类资源的根目录， 注册main、auth、admin、projects蓝本， 初始化了服务器变量app和数据库变量db。

2.8 models.py: 数据库表文件。借助flask-sqlalchemy扩展提供的ORM框架定义了roles， users， projects, logs四个数据表，和相关的函数。

1. xls\_digital\_railway/app/static 静态文件目录
   1. css/ 所有css文件。
   2. js/ 所有js文件。
   3. images/ 图片文件。
   4. tools/ jQuery, font-awesome等工具文件。
   5. upload/ 面线、横断面文件上传目录
   6. control\_panel/ 项目浏览页面右侧弹出菜单的按钮图片。
   7. plugins/ 项目浏览页面的其他依赖。
   8. FLYProject/: 指向局域网路径 [\\192.10.15.200\FLYProject](file:///\\192.10.15.200\FLYProject) 的软链接。
   9. video/: 指向局域网路径 [\\192.10.15.100\video](file:///\\192.10.15.100\video) 的软链接。
2. xls\_digital\_railway/app/templates

base.html: 导航栏、登录窗口的模板。

index.html: 首页(/index)

search.html: 视频搜索结果页

content\_bast.html: 风采展示、视频列表的模板

overview.html: 风采展示页

video.html: 视频列表页

auth/account.html: 用户信息页面

auth/register.html: 注册页面

projects/project\_index.html: 项目列表页, url: /projects

projects/project\_explore.html: 项目浏览页, url: /projects/explore?pid=<id>

admin/\*.html: 后台管理页面

error/404.html: 错误页面(404, 500等)

html文件关联的css、js文件在app/static/css与app/static/js目录下。

## 三、服务器部署说明

1. 源代码位置

源代码保存在192.10.12.40计算机, D:/projects/xls\_digital\_railway目录下。

git远程仓库地址:

2. 环境说明

windows平台不适合部署的生产环境的web服务器，由于本项目服务范围较小，直接使用flask自带的开发服务器足以满足使用需求。

部署之前，系统需安装python3.5+环境[[3]](#footnote-3)。以管理员权限启动命令行，切换到项目根目录，执行 pip install –r requirements.txt 一键安装依赖包。或直接输入pip install flask flask-sqlalchemy psycopg2安装。

demon.py脚本可以根据服务器监听的端口号监测后台进程，持续监测服务器运行状态，可以随时终止、重启服务器。

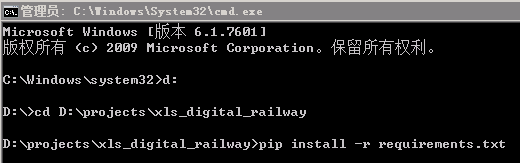


图3.1 安装依赖包

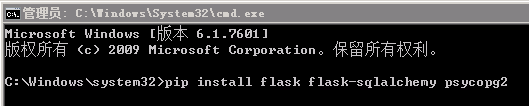


图3.2 安装依赖包(需管理员权限)

3. 服务器启动方法

3.1 **首次启动生产环境(重要)**

开发环境使用的数据库为tsy\_dev，部署生产环境时，应该事先切换数据库至tsy\_digital\_railway，并初始化管理员用户。

打开config.py，将default设置为ProductionConfig。

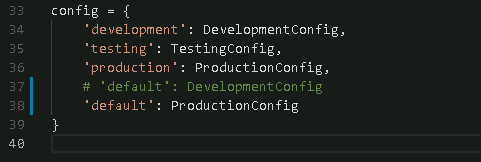


图3.3 更改启动环境

命令行切换到项目根目录，依次输入以下命令，初始化数据库，并加入管理员用户。

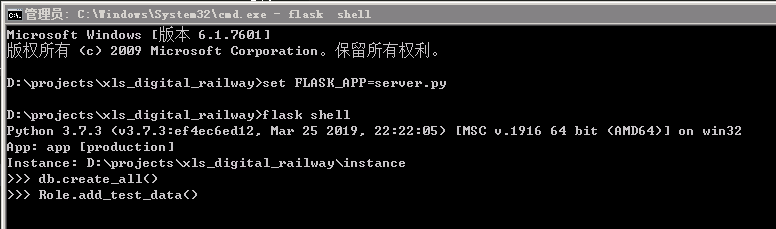


图3.4 初始化数据库和管理员用户

3.2 方式一：直接在前台运行服务器(**不推荐**)

命令行切换到项目根目录，输入命令：python server.py --host 0.0.0.0 –p 8000 启动服务器。可选命令：

--host <IP地址>;

--post <端口> 或 -p <端口>;

-d 1 以debug模式启动。



图3.5 直接启动服务器

3.3方式二：以守护进程方式运行服务器(**推荐**)

使用demon.py将server.py作为后台进程启动。

方法一：命令行切换到项目根目录, 输入命令：python demon.py 8000 启动守护进程脚本，指定服务器监听8000端口。

方法二：双击demon.py，以默认端口8000启动该脚本。

方法三：将demon.py加入windows计划任务。详见3.5。

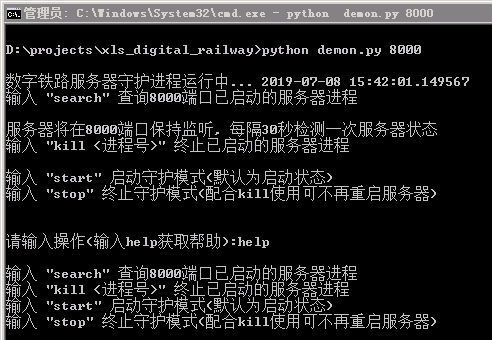


图3.6 启动demon.py守护脚本

3.4 demon.py守护进程脚本操作说明

输入search，根据监听端口查询已经启动的服务器进程号。

输入kill <pid>，强制终止进程号为pid的命令。

输入start，启动守护模式(默认未启动状态)。

输入stop，停止守护模式。需要完全关闭服务器时，可先使用stop命令，再使用kill命令，确保服务不再自动重启。

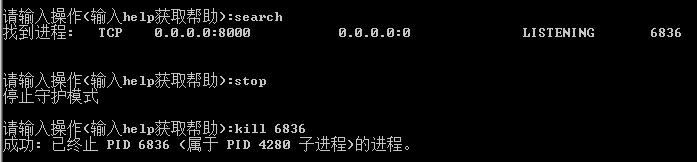


图3.7 搜索、停止服务器进程

3.5 将demon.py加入windows计划任务(**已加入**)

第一步：打开任务计划程序，点击创建任务：

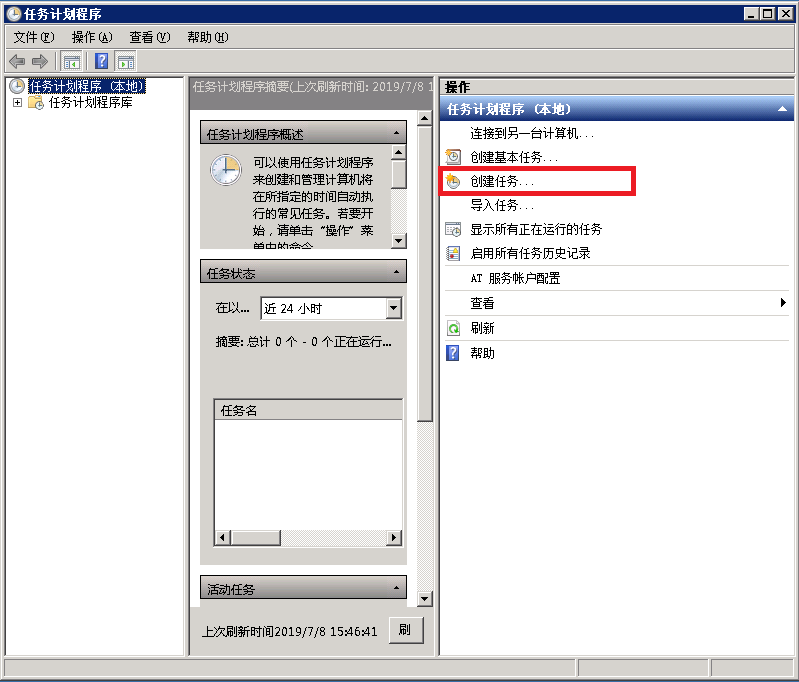


图3.8 创建任务

第二步：添加任务名称、触发器、启动操作和其他设置。

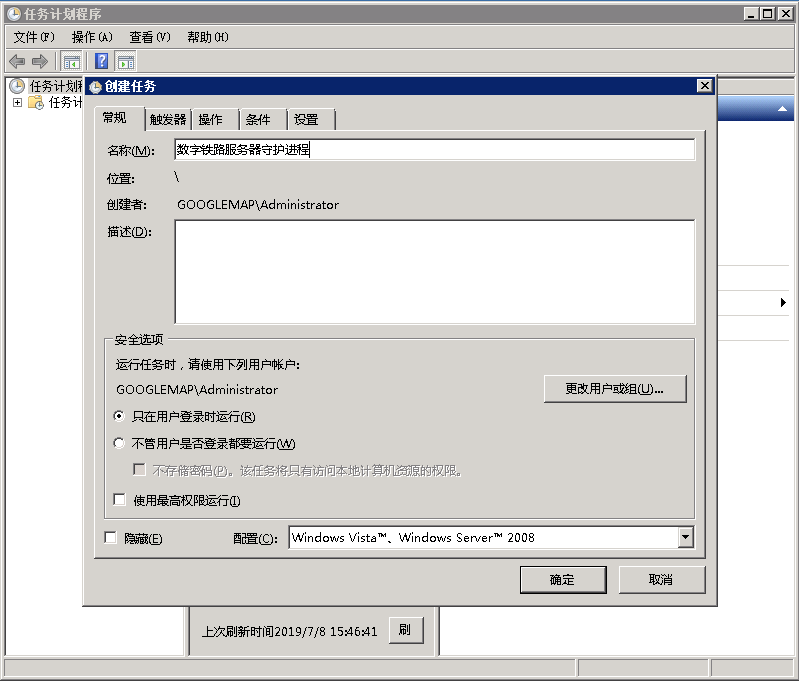


图3.9 添加任务名称

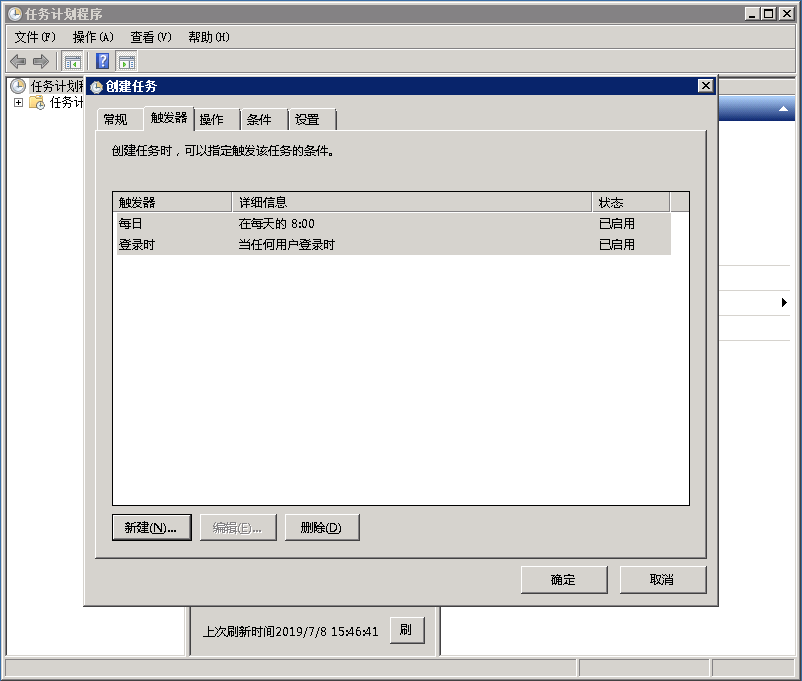


图3.10 添加触发器

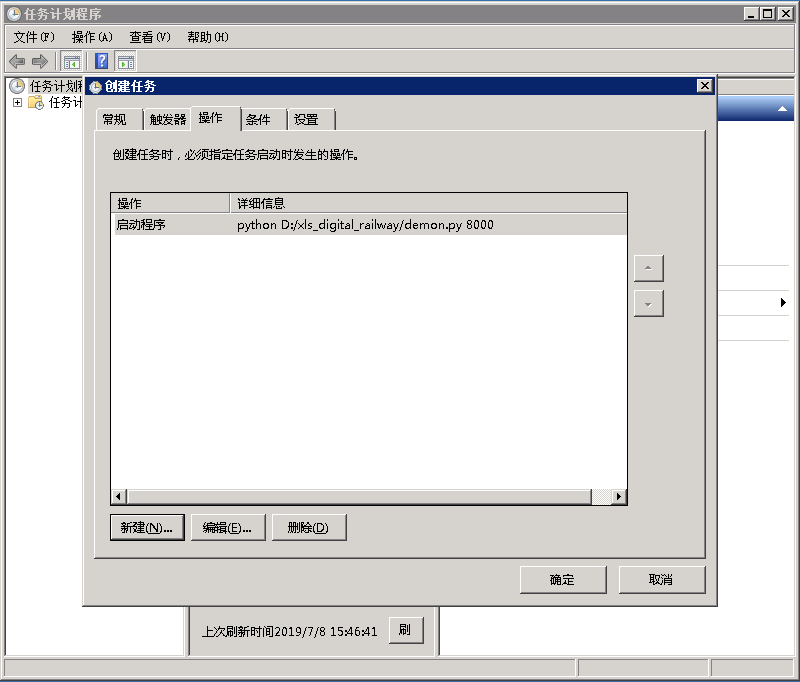


图3.11 添加启动操作

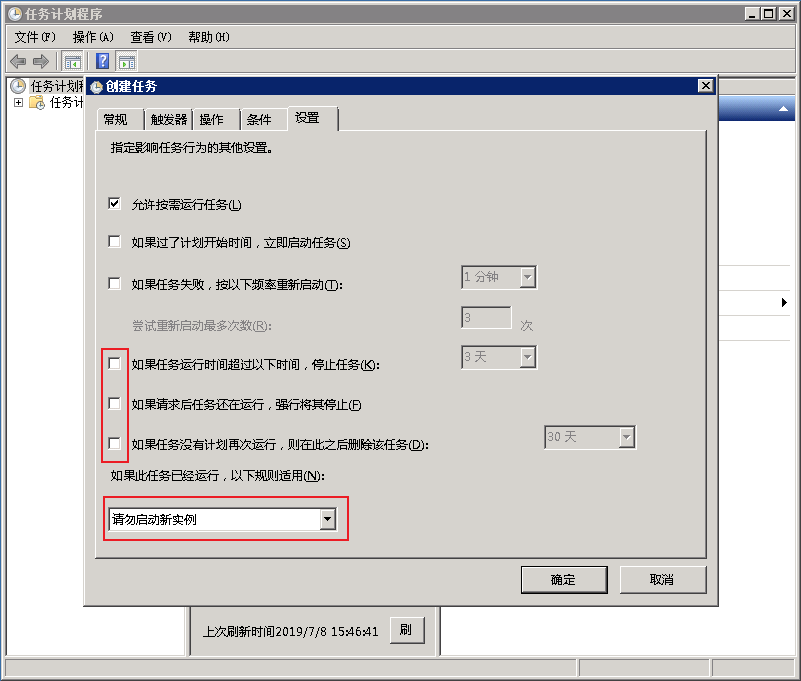


图3.12 其他设置

第三步：点击运行，启动任务

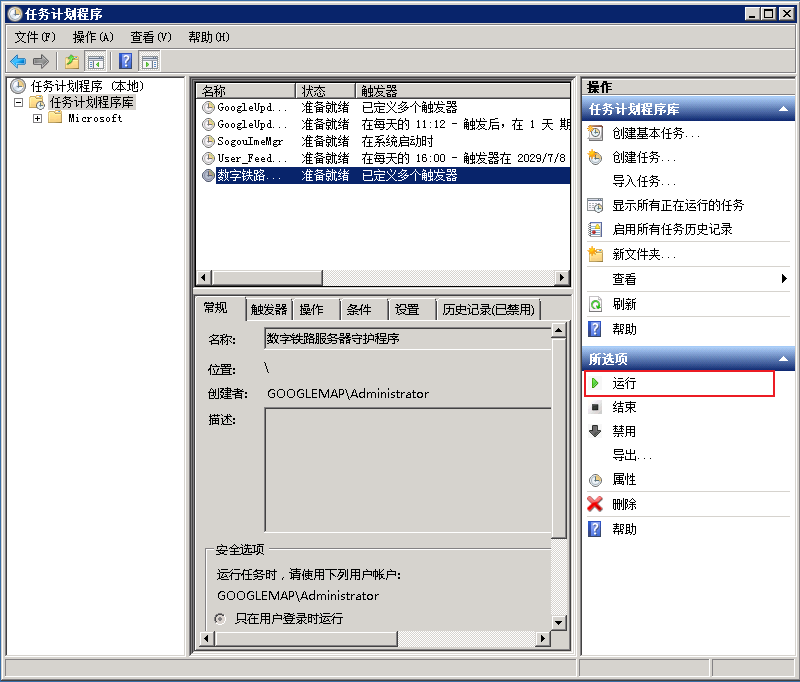


图3.13 点击运行，启动任务

## 四、网站使用方法

1. 网站访问

服务器部署后, 通过内网IP 192.10.12.40:8000即可访问三维数字铁路网络应用系统。

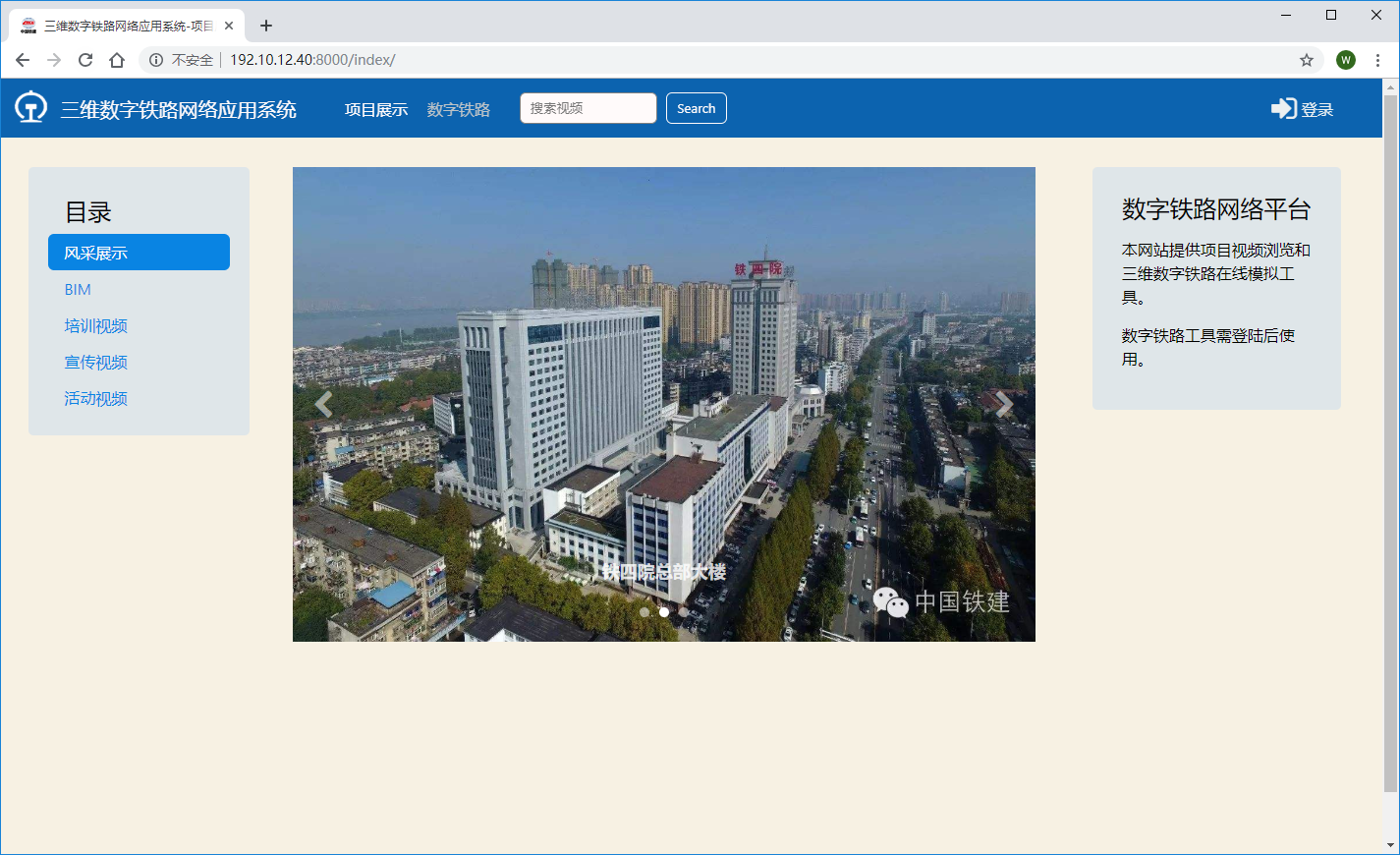


图4.1 首页展示

2．用户信息

用户可以自助注册、登录，登录后修改自己的相关资料，包括电话、邮箱、单位、部门、个人简介等。

3. 视频浏览

无需登录, 就可以观看项目的视频。可以根据关键字搜索视频。

4. 项目浏览

若需要查看数字铁路项目，需要注册登录，等待管理员授予相关项目的浏览权限，才能浏览项目。数字铁路项目依赖IE ActiveX控件，需要通过IE浏览器才能正常访问。

浏览项目时，可实现查看纵断面、查看横断面、提取地面线、加载本地或网络地形文件、线路交通及飞行模拟等功能，此外还提供了TerraExplorer原生的100余种功能，包括测距、地形分析、创建图形、创建特效等。

5. 管理员模块：

管理员可以查看修改所有用户的资料。

管理员可以检查新增项目、删除失效项目。

每个用户都属于一个用户组，管理员可以新建、修改、删除用户的组别。

用户和用户组分别有属于自己的权限，管理员可以修改此权限。

每个项目有一个分组，管理员可以修改此分组。

用户和用户组的权限可以是一个数字，或是小数点分隔的多个数字。其中任何一个数字和项目的分组相等时，该用户即可浏览此项目。例如：用户张三的个人权限为1.5，张三所在的线站处用户组的权限为2，那么张三可以浏览分组为1、2、5的项目。

## 五、网站资源维护

1. 首页风采展示

风采展示图片目录为 [\\192.10.15.200\FLYProject\其他资源\风采展示](file:///\\192.10.15.200\FLYProject\其他资源\风采展示)，请使用.jpg格式的图片。图片名称会作为标题显示在首页幻灯片上。

2. 视频目录

视频目录为 [\\192.10.15.100\video](file:///\\192.10.15.100\video)，请使用.mp4格式的视频。视频缩略图请使用.jpg格式，并与视频同名。视频描述请使用.txt格式，保存为utf-8编码，并与视频同名。

视频文件的改动会自动同步到服务器端，不需要进行额外的操作。

3. 项目目录

项目目录为 [\\192.10.15.200\FLYProject](file:///\\192.10.15.200\FLYProject)。项目主文件命名为main.fly。项目展示图片为logo.jpg。项目介绍文件为note.txt，保存为utf-8编码。

项目资料在第一次被扫描时会全部存入数据库，包括名称、url、描述等。新增项目是根据项目文件夹名称进行识别的，新增或修改项目后需进入项目管理页面进行同步。服务器不会自动从数据库中删除已经不存在的项目，包括修改了名称或删除的项目，需要在项目管理页面手动删除。

## 六、代码介绍

1. 数据库

本项目使用flask-sqlalchemy ORM框架进行数据库操作。数据库连接参数由config.py中的SQLALCHEMY\_DATABASE\_URI常量定义。开发环境数据库名为tsy\_dev，生产环境数据库名为tsy\_digital\_railway，均已在192.10.12.40的postgresql服务端创建完成。数据表结构见models.py文件，有详细注释。

2. 后端运行模式

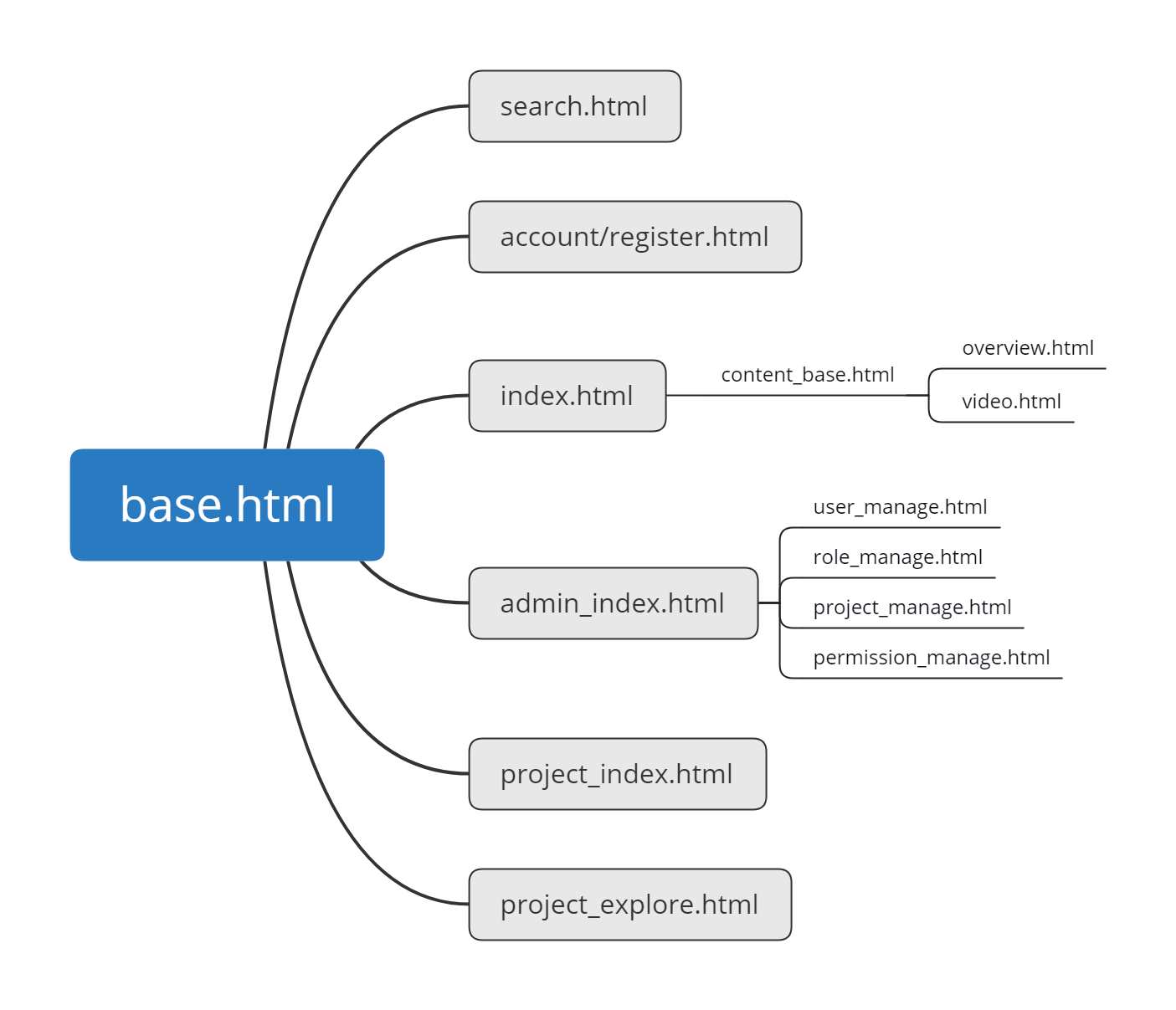
flask框架使用的是MTV开发模式(Model-Template-View)。用户发起http请求，视图函数接受请求，调用数据库模型(如果需要)，将返回的数据渲染进模板，返回给用户。

main、auth、admin、projects为四个视图蓝本，所有的视图函数均定义在内。html模板文件均在templates文件夹中。数据库模型储存在models.py中。

3. 前端代码

前端代码分为html、javascript、css三部分。html作为模板储存在templates文件夹中。绝大部分的css和js文件均独立于html文件。css文件、js文件分别储存在app/static/css和app/static/js中。jQuery源代码和font-awesome图标库储存在app/static/tools中。

html视图通过flask自带的jinja2模板语言进行渲染，{{ expression }}中可书写python表达式，{% sentence %} 中可书写循环或分支语句，通过{% block name %}可进行模板继承。详细语法可参见官方文档[[4]](#footnote-4)。模板继承关系图如下。



­图6.1 模板继承关系图

1. 路由：指相对于http域名的相对路径， 如/index, /admin等，在@route(‘/index’)中指定。 [↑](#footnote-ref-1)
2. 视图函数：处理客户端对应路由请求的python函数。由@route(...)装饰器修饰的函数。 [↑](#footnote-ref-2)
3. 可参考<https://www.liaoxuefeng.com/wiki/1016959663602400/1016959856222624> [↑](#footnote-ref-3)
4. jinja2中文文档 <http://docs.jinkan.org/docs/jinja2/> [↑](#footnote-ref-4)