黑利博瑞house web 开发计划 v1.2

开发目标：1，安全稳定保存house相关的文件a

2，互联网中展示house的相关图文信息

3，简易化操作流程更加方便的使用

项目位置：

提交者：peter.duan & Jason

计划创建时间：2019/8/26

计划更新时间：2019/9/05

目录：

[1，程序大体思路与宏观模块定义](#程序大体思路与宏观模块定义)

[2，项目文件夹结构定义](#项目文件夹结构定义)

[3，项目资源定义](#项目资源定义)

[4，公开发布模块详细结构设计](#公开发布模块详细结构设计)

[5，数据库结构设计](#数据库结构设计)

[6，程序URL路由 与 API定义](#程序URL路由与API定义)

7，核心控制模块详细结构设计

8，

程序大体思路与宏观模块定义：

为了实现程序开发目标，我们将整个项目分为四个模块

分别有：存储模块，公开发布模块，用户控制模块，管理控制模块，保护模块

存储模块定义：

存储模块用于长期 安全 稳定地保存文档等所有类型文件

自动化归档公开模块的图文备份

实现基于版本管理的自动化全内容备份（发送至保护模块）

公开发布模块定义：

公开发布模块用于发布用于展示的图文信息

此模块只接受管理控制模块的调用

核心控制模块定义：

由开发者直接操作的具有最高权限的模块

用于控制各个模块的功能，用于上传图文消息至公开发布模块

保护模块定义：

用于备份整个服务器的安全服务器，保护模块相对独立。

文件目录结构定义：（项目并不完整）

Hlby\_house\_web #项目文件夹

|— app #程序主体文件夹

|— main.py #主程序

|— tool.py #环境设置工具

|— static #静态模版资源

|— img #图片资源文件夹

|— year\_min #年度缩略图存储文件夹

|— act\_min #活动缩略图存储文件夹

|— det #图文图片存储位置

|— templates #动态模版资源

|— index.html #html主页

|— test #测试代码文件夹

|— control #控制端文件夹

|— core\_control.py #最高级控制程序

|— user.py #用户控制程序

|— ssl #ssl 证书保存文件夹

|— requirements.txt #python 开发模块记录

|— architecture #程序开发结构图文件夹

项目资源定义：

此程序考虑安全因素，主要设计为在内网运行

安全优化可以延后考虑

开发语言：python2X 并使用 flask web框架 构建

程序主服务器IP：

程序端口：443

程序使用https协议

开发环境 Google cloud 服务器

台湾 任意地区 Linux Debian 9 系统

程序使用路径使用绝对路径，项目文件夹放置在Linux主机的绝对位置上

https证书由“let`s encrypt”机构免费提供

公开发布模块详细结构设计：

公开发布模块 信息存储基本单位为 一条图文消息

模块中含有三个层级，分别是 年度层 活动层 以及具体 图文层级

年度层为整个年度学期的分化，类似于 2019上半学期 2019下半学期

活动层位于年度层之下，是具体活动的分化，例：开学军训

图文层级位于活动层之下，是具体图文的分化，例：男生军训

访问时顺序为 年度层级 -> 活动层级 -> 具体图文消息

年度层 以及 活动层 和 具体图文消息 全部需要独立添加

添加顺序同访问顺序，不可出现重复问题，也不可出现顺序错误！

数据库结构设计： （图纸参见项目数据库图纸）

数据库分为三张数据表，学期表，活动表，图文表

第一张表为学期表，是所有数据表中范围最大的表 表名称：years

目的是记录所有学期的信息概况

|  |  |  |  |
| --- | --- | --- | --- |
| id | year | describe | photo |
| 1 | 2019年上学期 | 快乐的一年 | static/a1/a.png |

id为主键

year为记录年度标题名称的列 类型为字符串

describe为记录年度标题的简略描述 类型为字符串

photo为记录年度标题中缩略图的位置 类型为字符串

第二张表为活动表，是用于记录所有活动的表 表名称：act

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | activity | describe | photo | years\_id |
| 1 | 新学期军训 | 同学很认真 | static/a1/b.png | 1 |

id为主键

activity为记录活动名称（标题）的列 类型为字符串

describe为记录活动的简略描述 类型为字符串

photo为记录活动标题中缩略图的位置 类型为字符串

years\_id 关联years表中的主键 形成关系表

第三张表为图文表 用于具体记录单个图文信息 表名称：det

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | name | describe | photo | body | act\_id |
| 1 | green1 | 同学开心 | b.png | 正文 | 1 |

id为主键

name 为记录单个图文信息名称的列 类型字符串

describe 用于记录图文信息名称下方的描述 类型字符串

photo 用于记录图文信息的缩略图位置 类型字符串

body 用于记录整个图文信息正文内容 类型字符串

act\_id 关联act表中的主键 形成关系表

body 列的信息结构是一个 二维列表 由以下部分组成

[ [ 1 , 大家好 ] , [ 2 ，abc/he/a.png ] ]

其中 数字“1”和数字“2”为类型头

后面的字符串为正文内容

第一层列表为一个大型载体，目的是为了更好的循环读取

第二层列表中分为两个部分，类型头和正文

类型头存储 字符串类型的十进制数字 用于标志不同的文本类型

正文存储与类型头相对的数据，如文本，照片静态位置等

body 列存储时使用字符串形式存储，所以定义了一套列表分化系统

[列表分化系统](#列表分化系统) 在这里并不详细介绍

程序URL路由 与 API定义：

主程序公开发布模块URL

“/”#公开模块主要路由，此页面为house宣传主页

“/years/<v>/”#年度学期分类URL 变量v 为年度名称，此URL访问来源于

核心主页引导直接转跳

“/act/<v>/”#活动分类传参URL 变量v 为活动名称，此URL访问来源于

years主页引导直接转跳

“/det/<v>”#详细图文传参URL变量v 为图文名称，次URL访问来源于act/

下详细活动引导转跳

API目前共有三个模块

上传模块，删除模块，查询模块

“/api/?config=”

API中参数config的值为一个包含多个信息的字符串

字符串中信息分割使用字符串“å”分割

此字符串参数根据不同API有不同长度输入具体定义如下：

“head å cho å body å body”

此字符串中由一个head头部信息和cho层级选择以及其他数个body正文组成

head头部信息有三个内容 分别对应着三个API模块

有“update”和“delete”以及“search”

Cho层级选择也有着三个内容分别是 year ，activity ，det

分别对应着 年度学期，活动，具体图文 三个层级（层级内容详见公开模块）

API三个功能模块都可以分别调用三个cho层级选择器

Head与Cho还有body正文内容根据功能不同有着不同的定义

使用Search功能以及任何一个cho选择器时 不需要加入任何body正文

格式为：“search å years”

使用Delete 功能以及任何一个cho选择器时 基本参数全部相同

格式为：“Delete å years å name”

其中，在body位置的参数为name，是与前cho选择器相关联的名称

update 上传功能最为复杂，三个cho选择器分别有三种不同的提交格式

update上传年度 格式以及信息如下

“update å years å year\_name å describe å photo\_path”

格式中body位置分别是 年度名称，详细描述 和 缩略图路径 三个参数

update上传活动 格式以及信息如下

“update å activity å activity\_name å describe å photo\_path å year\_id”

格式中body位置分别是 活动名称，详细描述，缩略图路径 以及 年度外键

年度外键就是 年度的名称

update上传图文 格式以及信息如下

“update å det å det\_name å describe å photo\_path å body å act\_id”

Update上传图文参数 除了头信息和cho选择器

分别是 图文名称，图文描述，缩略图路径，图文正文 以及 活动外键

其中活动外键就是 此图文隶属于的那个活动的名称

参数图文正文 是整篇图文的正文内容 为了实现多样化的图文效果

我们定义了 图文描述规则 详细参见 [图文描述规则](#图文描述规则)

API 图片上传函数路由定义为：

"/api/upload/"

此为上传图片的核心API 图片上传时 需使用特殊的命名法则

服务器位置中放置图片一共四个分类

图片名称一共有两个部分 第一个为类型头 第二个为uuid

例子“act\_min!uuid”

类型头与此文件所属的文件夹名称相同

附加资源：

字符串与列表转换 列表分化系统

因需要将二维列表转换为一个完整的字符串进行保存

所以整个字符串由“∂”和“ƒ”进行分隔

例1： ∂ 1 ƒ大家好 ∂ 2 ƒ我是一个人类 ∂

例2：[ [ 1 , 大家好 ] , [ 2 ，我是一个人类 ] ]

例1与例2 可以相互互译 使用python分割函数进行处理

“∂”为第一层列表 “ƒ”为第二层列表

图文描述规则

图文描述规则是一套类似于html标记语言的规则

此规则在 存储时使用一套 二维列表实现，类型为字符串

字符串与列表转换 使用上方的 [列表分化系统](#列表分化系统) 实现（在数据库存储时使用）

结构为：

[ [ h1 , 大家好 ] , [ pho , /hello/a.png ] ]

其中 第二层列表中 分为两个部分 （只具有两个部分）

第一个部分“h1”为标记头信息 第二个部分“大家好”为值

“h1”和“pho”都为 标记头信息 用与记录第二部分值的类型

标记头信息对照表：(表中信息大小写不敏感)

|  |  |  |
| --- | --- | --- |
| 标记头部信息 | html对应标签功能 | 值的格式与内容 |
| h1 – h6 | 六个大小不同的标题 | 字符串 |
| p | 段落 | 字符串 |
| br | 换行 | 值任意赋值没有意义 |
| img | 插入图片 | 此标签为图片位置 |
| anthor | 作者 | 字符串 |
| time | 时间 | 字符串 |
| lead | 浅色字体 | 字符串 |
| cite | 引号引用 | 字符串 |
| title | 大标题 | 字符串 |