

Image Area技术文档

如何运行

1. 需预先安装：
 - Node.js 8.12.0以上的版本
 - MongoDB
2. 解压缩后
 - 在ImageArea/bin文件夹中打开控制台
 - 输入 `node www` 后即可运行
 - 打开浏览器，输入 `127.0.0.1:3000` 进入网站
3. 如果没有安装MongoDB则只能以游客模式浏览
4. 安装MongoDB后可以注册用户并登录

框架

Express.js

视图(📁: ImageArea/views)

1. 模板引擎: Pug

因为本项目后端使用 `express.js` 框架搭建，所以选择Pug作为网页模板。

Pug模板引擎的优势：

- 判断语句 (if-else)
- 循环语句 (each item, i in iterator) —— 图片、标签等的展示大量运用列表。

2. 视图模块概览

- **个人模块**：album.pug（相册），batchupload.pug（批量上传），forgot.pug（忘记密码），image.pug（公开相册图片），info.pug（个人信息），privateimage.pug（私人相册图片），reset.pug（重置密码），upload.pug（上传图片）
- **访问他人模块**：accessalbum.pug（访问他人相册），accessimage.pug（访问他人图片）
- **试验功能模块**：detectionupload.pug（批量识别上传），dusingle.pug（单张图片识别上传），tagcloud.pug（标签云）
- **公共视图模块**：error.pug（错误），home.pug（首页），layout.pug（导航栏布局），register.pug（注册），login.pug（登录），search.pug（搜索）

路由文件(📁: ImageArea/routes)

每个路由文件控制一个模块的所有功能的请求路径，并与前端交互。

1. access.js: `127.0.0.1:3000/access/*`

访问他人相册、图片的路由

2. experiment.js: `127.0.0.1:3000/experiment/*`

试验功能（标签云、内容检索上传）的路由

3. index.js: `127.0.0.1:3000/*`

公共视图模块的路由，以及处理首页图片点赞、评论事件

4. personal.js: `127.0.0.1:3000/personal/*`

个人模块的路由，包含个人信息、个人相册、个人图片的展示与事件处理

5. search.js: `127.0.0.1:3000/search`

搜索功能的路由

6. upload.js: `127.0.0.1:3000/upload/*`

处理上传事件的路由

7. users.js: `127.0.0.1:3000/user`

处理用户注册、登录、忘记密码的路由

样式与前端JS(📁: ImageArea/public)

网页样式CSS文件，前端JS文件

后端(📁: ImageArea/models)

与OSS对象存储的交互中间件，技术聚合文件，以及MongoDB用户信息存储模型

1. OSS对象存储中间件

- imageAreaOSS.js: 管理用户所有图片信息
- profileOSS.js: 管理用户的头像

2. 技术聚合文件

- technologyOSS.js: 技术聚合文件

3. MongoDB用户信息存储模型

- user.js: MongoDB用户数据库管理

内容检索(📁: ImageArea/object_detection)

内容检索使用python的tensorflow包

- 核心文件: `object_detection_image2json.py`，将识别结果输出到json文件，供调用。

路由总览

- "<>"包裹的部分表示可变内容
- 省略根目录 `127.0.0.1:3000`

1. /access/*

- `/access/<username>` 访问他人所有公开相册
- `/access/<username>/public/<albumname>` 访问他人公开相册中的图片

2. `/experiment/*`

- `/experiment/tagcloud` 进入标签云页面
- `/3000/experiment/objectdetection` 批量内容检索上传图片页面
- `/3000/experiment/dusingle` 单张检索上传图片页面
- `POST /experiment/objectdetection` 后台识别, 向前端返回结果
- `POST /experiment/detectionrelatively` 后台单张识别, 向前端返回结果

3. `/*`

- `/` 进入首页
- `/images/` 请求首页图片, 返回五张图片
- `/messages` 处理图片评论上传
- `/like` 处理图片点赞事件

4. `/personal/*`

- `/personal/album/<policy>` 访问个人所有公开(public)/私密(private)相册
- `/personal/album/<policy>/<albumname>` 访问个人相册中的所有图片
- `/personal/info` 访问个人信息
- `POST /personal/info/profile` 更新头像
- `POST /personal/introduction` 更新个人说明
- `POST /personal/info/pc` 更新私密量 (会员)
- `POST /personal/album/new` 新建相册
- `POST /personal/vip/open` 开通会员
- `POST /personal/vip/close` 注销会员
- `POST /personal/delete/image` 删除单张图片
- `POST /personal/delete/batchimage` 删除多张图片
- `POST /personal/info/tag` 添加标签 (会员)

5. `/search`

- `/search` 进入搜索页面
- `POST /search` 后端处理搜索, 向前端返回结果

6. `/upload/*`

- `/upload/images` 进入上传图片页面
- `POST /upload/images` 向云端上传图片
- `/upload/batchimages` 进入批量上传图片页面
- `POST /upload/batchimages` 向云端上传批量处理的图片

7. `/users/*`

- `/users/register` 进入用户注册页面
- `/users/login` 进入用户登录页面
- `/users/logout` 用户登出
- `/users/forgot` 进入忘记密码页面
- `/users/reset/<token>` 进入重置密码页面
- `POST /users/register` 处理用户注册事件
- `POST /users/login` 处理用户登录事件
- `POST /users/forgot` 处理忘记密码事件

- **POST** /users/reset/<token> 处理重置密码事件

后端函数接口总览

- 接口函数全部使用异步回调（callback参数）
- 回调函数第一个参数是错误信息（err）

1. imageAreaOSS.js

函数名	参数	说明
createUserArea	(newUser, callback)	在OSS创建用户信息与空间
uploadImages	(username, images, callback)	上传图片
getImages	(username, albumInfo, callback)	获取个人相册中的所有图片
getAccessImages	(username, albumInfo, callback)	获取他人相册中的所有图片
addAlbum	(username, albumInfo, callback)	添加相册
getTagsAndAlbums	(username, callback)	获取用户的标签和相册名
getPrivateAlbums	(username, callback)	获取所有私密相册信息
getPublicAlbums	(username, callback)	获取所有公开相册信息
uploadMessage	(username, message, callback)	提交评论，成功后返回最新的三条评论用于刷新
getMessagesOfImage	(username, imgId, callback)	
addLikeToImage	(username, imgInfo, callback)	用户点赞一张图片
removeLikeOfImage	(username, imgInfo, callback)	取消点赞
ifUserLikeTheImage	(username, imgId)	判断用户是否喜爱该图片
getUserInformation	(username, callback)	获取用户信息
uploadIntroduction	(username, intro, callback)	上传用户个人说明
openVip	(username, callback)	开通会员
closeVip	(username, callback)	注销会员
changePrivateCapacity	(username, pc, callback)	更改会员用户私密量

addTag	(username, tag, callback)	会员用户添加新的标签
removeTag	(username, tag, callback)	会员用户删除标签
deleteImage	(username, albumname, uuid, callback)	删除云端的图片
deleteBatchImage	(username, albumname, uuids, callback)	批量删除云端的图片

2. profileOSS.js

函数名	参数	说明
uploadProfile	(userId, profile, callback)	上传用户头像
getProfileImage	(userId, callback)	获取用户头像

3. technologyOSS.js

函数名	参数	说明
getRandomImages	(callback)	获取五张首页图片
search	(query, callback)	搜索用户名和标签名
searchUsers	(query, callback)	搜索用户名
searchImages	(query, callback)	搜索所有带该标签的图片
addImageToTags	(imagesInfo, callback)	将图片归类至各个标签
removeImageOfTags	(imagesInfo, callback)	移除标签中的图片信息
getAllTags	(callback)	获取所有标签
detectTagsOfImage	(files, callback)	检索多张图片中的内容
detectTagsOfImageRelatively	(files, callback)	检索单张图片中的内容
getAllTagsWithFreq	(callback)	获取所有标签和它的图片数

4. user.js

函数名	参数	说明
getUserById	(id, callback)	根据ID查找用户
getUserByUsername	(username, callback)	根据用户名查找用户
comparePassword	(candidatePassword, hash, callback)	校对密码
getUserByEmail	(email, callback)	根据邮箱查找用户
createUser	(newUser, callback)	创建用户
ifUserNameExists	(username, callback)	判断用户名是否存在
ifEmailUsed	(email, callback)	判断邮箱是否已注册