**接口文档**

一、 概述

本文档针对Unitoto的不同功能，提出了对于后台提供的接口的要求和内容约束，后台前端均需围绕该文档来进行内容的开发，若有变动，将讨论后决定。

二、 共同约定

前端不对后台的内容进行干预，后台只负责处理前端发送的请求，并做出相应的逻辑处理后发回前端；前端只需要根据用户输入将请求发给后台，并根据后台返回的数据进行界面的渲染。

前后端通信的所有数据格式，应严格控制为json或json数组，以免解析出错带来的不必要的麻烦。

三、 验证方式

采用最基本的cookie来进行验证即可，后期会提高安全性

四、 需求

本部分内容由前端定义

1. 主界面

主界面即为动态界面，该界面的内容为所有用户分享的图片或相册动态，需求如下：

a. 传入用户id，返回最新的n张图片的所有信息（json）

b. 传入用户id以及图片id，返回该id之后的15张图片的所有信息（json）

1. 个人界面

个人界面为用户的个人主页，需求如下：

a. 传入用户id，返回用户的基础信息(电话、邮箱、个人描述)

b. 传入用户id，返回用户的所有相册信息（json）

c. 传入用户id，返回用户加入的社区

1. 搜索推荐界面**(后台暂时返回随机数据即可)**

搜索推荐界面主要为搜索和系统推荐，需求如下：

a. 传入用户id，返回机器推荐的图片id

b. 请求，返回每日推荐的图片id

c. 传入用户id以及搜索内容，返回搜索结果的图片信息（json）

1. 社区界面

社区界面，需求如下：

a. 传入用户id，返回推荐的社区信息(包括社区id)

b. 传入用户id和社区id，返回社区内的聊天信息**(暂不需要实现)**

1. 地图界面**(暂不需要实现)**

五、 后台接口

本部分由后台完成，需定义接口地址、请求方式、所需的传入数据、可选的传入数据、数据格式和返回的数据内容及格式

**示例：**

**接口地址：/image**

**接口描述：通过图片id获得具体图片**

**请求方式：GET**

**传入数据及格式：**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| image\_id | string | 是 | 需要取得的图片id |
| user\_id | string | 否 | 鉴定用户权限，可省略 |

**传回的数据及格式：编码的图片流**

**正文如下：**

**-------------------------------------------------------------------**

**图片刷新**

接口地址：/getImageDownflashing.do \*

接口描述：**下拉刷新操作**

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userId | String | 是 | 用户的id |

返回数据：最新的3个项目：[用户，图片，评论]

包括每一个图片（为一个JSONObject）的JSONObject

每一项具体如下

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| userId | String | 用户的id |
| userName | String | 用户的name |
| photoId | String | 图片的Id |
| photoAddress | String | 图片的实际存放地址 |
| photoContext | String | 关于本图片的作者评论（可为null） |
| commentContext | String | 关于本图片的1条评论 |
| photoLikeNum | String | 本图片的点赞数 |
| commentSize | int |  |

接口地址：/getImageUpflashing.do \*

接口描述：**上拉刷新操作**

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userId | String | 是 | 用户的id |
| photoId | String | 否 | 用户显示的最早的图片id |

返回数据：比当前最早显示的图片还要早的3个项目：[用户，图片，评论]

包括每一个图片（为一个JSONObject）的JSONObject

每一项具体如下

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| userId | String | 用户的id |
| userName | String | 用户的name |
| photoId | String | 图片的Id |
| photoAddress | String | 图片的实际存放地址 |
| photoContext | String | 关于本图片的作者评论（可为null） |
| commentContext | String | 关于本图片的1条评论 |
| photoLikeNum | String | 本图片的点赞数 |
| commentSize | int |  |

-------------------------------------------------------------------

接口地址：/uploadSingleImg.do \*

接口描述：**上传单张图片**

请求方式：POST

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| file | file | 是 | 图片文件 |
| photoContext | String | 是 | 上传者的文字评语 |
| userId | String | 是 | 用户的id |
| longitude | String | 否 | 上传图片位置的经度 |
| latitude | String | 否 | 上传图片位置的纬度 |

返回数据

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
|  | JSONObject | 成功时key为“Success”，value为该图片地址，可用于后续请求；失败时key为“Fail”，value为错误信息，包括“type must be one of 'GIF', 'PNG' or 'JPG'”, "type is null!", "file is null" |

接口地址：/uploadPluralImg.do \*

接口描述：**上传相册**

请求方式：POST

返回数据：成功或失败的Json

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| files | file | 是 | 图片文件 |
| photoContext | String | 是 | 上传者的文字评语 |
| userId | String | 是 | 用户的id |
| longitude | String | 否 | 上传图片位置的经度 |
| latitude | String | 否 | 上传图片位置的纬度 |

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
|  | JSONObject | 每一张图片按顺序从1开始编号，每一项的key均为“Success”或者“Fail”再加上图片的编号，譬如“Success1”表示第一张图片上传成功；每一项的value，若上传成功则为该图片的地址，可直接用于后续请求，否则就是错误信息，这些错误信息包括“type must be one of 'GIF', 'PNG' or 'JPG'”, "type is null!", "file is null"，"files is null"，最后一种错误信息只会有一项 |

-------------------------------------------------------------------

**添加标签**

接口地址：/addLabel.do \*

接口描述：为上传或者评论的图片添加标签，标签内容自行撰写

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| photoId | String | 是 | 评论或者上传的图片的id |
| labelContext | String | 是 | 加入的标签内容 |

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
|  | boolean | 成功为True，失败为False |

-------------------------------------------------------------------

**评论获取**

接口地址：/getComment.do \*

接口描述：获取一张图片的所有评论

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| photoId | String | 是 | 图片的Id |

返回数据：关于本图片的所有评论

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| commentContext | String | 关于本图片的所有评论 |
| photoContext | String | 关于本图片的作者评论（可为null） |

-------------------------------------------------------------------

**相册获取**

接口地址：/getMultiphoto.do \*

接口描述：获取一个相册的所有图片

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| photoId | String | 是 | 要获取的相册的封面图片，即该相册的第一张图片 |

返回数据：包括每一个图片（为一个JSONObject）的JSONObject

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
|  | JSONObject | 每一项都是一个图片的信息，也就是刷新主页时包括的图片信息：photoId, photoAddress, photoContext, photoComment, commentContext, commentSize photoLikeNum |

-------------------------------------------------------------------

**个人**

接口地址：/register.do \*

请求方式：POST

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | 用户的id |
| userpassword | String | 是 | 用户的密码 |
| username | String | 是 | 用户的昵称 |

接口地址：/login.do \*

请求方式：POST

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | 用户id |
| userpassword | String | 是 | 用户密码 |

接口地址：/updatetUserInfomation.do \*

请求方式：POST

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| username | String | 是 | user的昵称 |
| userphone | String | 是 | user的手机号 |
| useremail | String | 是 | user的email |
| userdescription | String | 是 | user的个人描述 |
| userid | String | 是 | user的id |

接口地址：/updateUserPassword.do \*

请求方式：POST

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | user的id |
| userpassword | String | 是 | user的password |

接口地址：/getUserInformation.do

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | 用户的id |

返回数据：用户的个人信息（昵称userid、手机userphone、邮箱useremail、个人描述userdescription） json

接口地址： /getUserPhotos.do \*

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | 用户的id |

返回数据：用户的图片(图片的id photoid、图片地址 photoadress、图片上传者id photouserid、图片上传者昵称 photousername、图片上传时间photouploadtime、图片标签photolabels{label：labelid，labelcontex}、图片文字描述photocontext、图片评论photocomments{comment：commentid，commentuserid，commentusername，commentcontext，commentuploadtime}、photolikenum） json数组 （注：photolabels和photocomments也为json数组

接口地址：/getUserCommunities.do \*

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | user的id |

返回数据：用户参加的社区id communityid、社区名称 communityname json数组

接口地址：/getFollowings.do \*

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | user的id |

返回数据：用户所关注的人userid、username json数组

接口地址：/getFollowed.do \*

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | user的id |

返回数据：关注当前用户的人userid、username json数组

接口地址：/addFollowings.do \*

请求方式：POST

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | 当前user的id |
| targetid | String | 是 | 需要添加关注的user的id |

接口地址：/addFollowed.do \*

请求方式：POST

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | 当前user的id |
| targetid | String | 是 | 当前用户的粉丝的userid |

接口地址：/deleteFollowings.do \*

请求方式：POST

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | 当前user的id |
| targetid | String | 是 | 需要删除的关注者的userid |

接口地址：/deleteFollowed.do \*

请求方式：POST

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | 当前user的id |
| targetid | String | 是 | 需要删除的粉丝的userid |

-------------------------------------------------------------

**社区**

接口地址：/addCommunity.do \*

请求方式：POST

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| communityid | String | 是 | 社区id |
| userid | String | 是 | 当前user的id |

接口地址：/deleteCommunity.do \*

请求方式：POST

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| communityid | String | 是 | 社区id |
| userid | String | 是 | 当前user的id |

接口地址：/createNewCommunity.do \*

请求方式：POST

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| communityname | String | 是 | 需要创建的社区的名称 |
| userid | String | 是 | 当前user的id |

接口地址：/getRecommendCommunities.do

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| userid | String | 是 | user的id |

返回数据：推荐给用户的社区id communityid、社区名称communityname json数组？不用用户头像？（唐玄X说不需要实现头像）

接口地址：/getCommunityInformation.do \*

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| communityname | String | 是 | 社区名称 |

返回数据：社区的信息（社区id communityid、社区名称communityname、社区创建者 communitycreator{creator： creatorid，creatorname} 、社区成员 communitymembers{user： userid，username} ）json数组 注：communitymembers为json数组

**-------------------------------------------------------------**

**地图**

接口地址：/getPhotoBySite.do（还没想好怎么写）

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |

**phtotoLongitude 经度**

**photoLatitude 纬度**

**-------------------------------------------------------------**

**搜索**

接口地址：/getPhotoByLabel.do

请求方式：GET

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| labelContext | String | 是 | 标签内容 |

六、 前端路由问题

1. @陶文宇 请注意，在向你user路由传入的参数包括userid这个参数，类型为string，代表你需要展示的用户的用户id，在你当前页面并不一定展示是当前用户的主页，也有可能是其他用户的信息，这点需要注意
2. 前端的全局信息利用vuex进行存储，所以大家的全局信息不需要自己存储