

接口文档

所有有可能会用到的实体类型

1. User

表示所有用户的实体

```
ID : long -- 实体唯一标识符
Account : string -- 账号
Password : string -- 密码
```

2. UserInfo

表示所有用户个人信息的实体，与用户是一一对应的关系。

```
ID : long -- 实体唯一标识符
NickName : string -- 昵称
RealName : string -- 真实姓名
University : string -- 学校
School : string -- 学院
Speciality : string -- 专业
SchoolYear : DateTime -- 入学年份
Introduction : string -- 个人简介
HeadImage : string -- 头像uri
```

3. Post

表示所有帖子（包括除了内容外的所有信息）的实体

```
ID : long -- 实体唯一标识符
Title : string -- 标题
Summary : string -- 文章摘要
Focus : string -- 聚焦，关键词
PostDatetime : DateTime -- 发布日期
```

4. PostInfo

表示所有帖子的内容的实体

```
ID : long -- 实体唯一标识符
Content : string -- 内容
```

5. Tag

表示所有标签的实体

```
ID : long -- 实体唯一标识符
_Tag : string -- 标签内容
```

所有接口

*** 这里所有接口提交数据的方法都为表单提交, 所有提交和返回的对象实体均使用JSON格式**

1. User控制器

- /User/GetUser

通过账号请求User实体

方法 : Post

请求参数 :

```
"Account" : string -- 账号
```

返回参数 :

```
User -- 用户实体
```

- /User/GetUserInfo

通过User实体请求UserInfo实体

方法 : Post

请求参数 :

```
"User" : User -- 用户实体
```

返回参数 :

```
UserInfo -- 用户信息实体
```

- /User/SelectTag

通过用户和限定条件选择标签

方法 : Post

请求参数 :

"User" : User -- 用户实体

"Selections" : JSONArray -- 选择用户标签的类型, 例如 [{"Type" : "Interested"}] 表示用户选择标签类型为"感兴趣"的标签

返回参数:

Array<Tag> -- 标签实体数组

- /User/SelectPost

通过用户和限定条件选择帖子

方法: Post

请求参数:

"User" : User -- 用户实体

"Selections" : JSONArray -- 选择用户帖子的类型, 例如 [{"Type" : "Owned"}] 表示用户选择标签类型为"自己拥有"的帖子

返回参数:

Array<Post> -- 帖子实体数组

- /User/SelectUserInitiative

通过本用户和限定条件选择主动去采取行为的用户.例如关注的用户,拉黑的用户

方法: Post

请求参数:

"User" : User -- 用户实体

"Selections" : JSONArray -- 选择用户采取行为的类型, 例如 [{"Type" : "Focus"}] 表示用户选择自己关注的用户

返回参数:

Array<User> -- 帖子实体数组

- /User/SelectUserPassive

通过本用户和限定条件选择对本用户采取行为的用户.例如关注你的用户,拉黑你的用户

方法: Post

请求参数:

"User" : User -- 用户实体

"Selections" : JSONArray -- 选择用户采取行为的类型, 例如 [{"Type" : "Focus"}] 表示用户选择自己关注的用户

返回参数:

Array<User> -- 帖子实体数组

- User/MappingPostsByTag

通过标签请求拥有该标签的帖子

方法: Post

请求参数:

"User" : User -- 用户实体

"Selections" : JSONArray -- 选择帖子拥有标签的类型, 此处类型默认为空, 填 [] 即可.

返回参数:

Array<Post> -- 帖子实体数组

- User/InsertUser

新增一位用户

方法: Post

请求参数:

"User" : User -- 用户实体, 不填 User.ID, ID由数据库自动生成

返回参数:

long -- 数据库生成的 User.ID

- /User/IsUserRelationExist

判断两个用户是否存在某个特定关系, 如关注或被关注 {"Type" : "Focus"}

方法: Post

请求参数:

"KeyUser" : User -- 用户实体1

"ValueUser" : User -- 用户实体2

"RelationName" : string -- 关系的键 "Type"

"RelationValue" : string -- 关系的值 "Focus"

返回参数：

bool -- 是否存在

- /User/AppendRelation

新增本用户与其他任意实体之间的一个关系, 如用户添加自己与一个标签的"感兴趣"关系

方法：Post

请求参数：

"User" : User -- 用户实体

"EntityType" : string -- 任一实体类型的名称 如 "Tag"

"ID" : string -- 上述实体的ID

"RelationName" : string -- 关系的键 "Type"

"RelationValue" : string -- 关系的值 "Focus"

返回参数：

bool -- 是否新增成功

- /User/AppendRelation

移除本用户与其他任意实体之间的一个关系, 如用户添加自己与一个标签的"感兴趣"关系

方法：Post

请求参数：

"User" : User -- 用户实体

"EntityType" : string -- 任一实体类型的名称 如 "Tag"

"ID" : string -- 上述实体的ID

"RelationName" : string -- 关系的键 "Type"

"RelationValue" : string -- 关系的值 "Focus"

返回参数：

bool -- 是否移除成功

- /User/CarculateSimilarity

计算本用户的与标签某个关系(个人标签或兴趣标签)与另外一个用户的个人标签的相似度 (未经过修正)

方法：Post

请求参数：

"User_1" : User -- 本用户

"User_2" : User -- 另外一个用户

返回参数：

double -- 相似度 (未经过修正)

- /User/CarculateSimilarityFix

计算本用户的与标签某个关系(个人标签或兴趣标签)与另外一个用户的个人标签的相似度 (经过修正)

方法：Post

请求参数：

"User_1" : User -- 本用户

"User_2" : User -- 另外一个用户

返回参数：

double -- 相似度 (经过修正)

2. Tag控制器

- /Tag/TagIndex

根据输入的关键字请求标签

方法：Get

请求参数：

"indexString" : string -- 关键字

返回参数：

Array<Tag> -- 返回所有匹配的标签

- /Tag/FindChildTag

根据父标签请求子标签

方法：Post

请求参数：

"ParentTag" : Tag -- 父标签

返回参数：

Array<Tag> -- 所有子标签

- /Tag/FindParentTag

根据子标签选择父标签

方法 : Post

请求参数 :

"ChildTag" : Tag -- 子标签

返回参数 :

Tag -- 父标签

- /Tag/CalculateSimilarity

计算两个标签的相似度

方法 : Post

请求参数 :

"tag_1" : Tag -- 标签1

"tag_2" : Tag -- 标签2

返回参数 :

double -- 相似度

3. Post控制器

- /Post/GetAllPost

获取所有帖子

方法 : Get

请求参数 : 无

返回参数 :

Array<Post> -- 所有帖子

- Post/GetPostInfo

根据Post请求PostInfo

方法 : Post

请求参数 :

"Post" : Post

返回参数：

PostInfo : 该Post的PostInfo(内容)

- Post/SelectAuthorInfo

根据帖子选择作者信息

方法 : Post

请求参数：

"Post" : Post

返回参数：

UserInfo : 该Post的作者信息

- Post/SelectTags

根据帖子选择该帖子的标签

方法 : Post

请求参数：

"Post" : Post

返回参数：

Array<Tag> : 该Post的所有帖子

- Post/InsertPost

插入一条新的帖子

方法 : Post

请求参数：

"Author" : string

"Title" : string

"Focus" : string

"Summary" : string

"Content" : string

"TagIDs" : Array<long> 拥有的标签ID

返回参数：

bool ：是否成功

4. Search控制器

- Search/SearchUserInfoAndTags

根据关键字搜索用户信息(包括用户拥有的标签)

方法：Get

请求参数：

"searchString" ：搜索字符串

返回参数：

Dictionary<UserInfo, Array<Tag>> -- Key用户信息, Value用户拥有的标签

- Search/SearchPosts

根据关键字搜索帖子信息

方法：Get

请求参数：

"searchString" ：搜索字符串

返回参数：

Array<Post> -- 所有帖子