接口文档

接口地址：`http://47.95.214.215/`

Service模块接口说明：

（1） ConcernedService接口

函数名称	函数功能	函数返回类型	函数传递参数
concerned	处理关注请求	ConcernedResponse	ConcernedRequest, Boolean
query	查询好友关系	FriendshipData	FriendshipData
update	更新好友关系	void	FriendshipData
fromJson	从JSON字符串中转换数据	ArrayList<String>	String, Class<ArrayList>
toJson	将数据转换为JSON字符串	String	Object

（2） FanAndConcernedService接口

函数功能	函数返回类型	函数传递参数	函数功能描述
queryFansAndConcernedResponseByUsername	FansAndConcernedResponse	String username	根据用户名查询粉丝和关注关系

（3） PostListService接口

函数名	函数功能	函数返回类型	函数传递参数
getPosts	获取并返回一个乱序的帖子列表	PostListResponse	无
selectAll	获取并返回所有的帖子ID	ArrayList<Integer>	无
subList	返回一个子列表，包含指定数量的帖子ID	ArrayList<Integer>	两个参数，第一个是开始索引，第二个是结束索引
shuffle	将帖子ID列表随机排序	void	ArrayList<Integer>

（4） PostList接口

函数名称	功能	返回类型	参数
add	添加帖子	PostResponse	PostRequest
query	查询帖子是否存在	Object	PostData
savePostImages	保存帖子图片	List<String>	PostRequest
query	查询用户信息	UserInfoData	UserInfoData

(5) RevoewService接口

函数功能	函数返回类型	函数传递参数	函数功能描述
add	ReviewResponse	ReviewRequest	该函数用于添加评论。在添加评论之前，它通过查询检查评论是否存在。如果存在，则通过随机生成新的评论ID来更新评论请求。如果评论不存在，则进行添加操作。
query	null	ReviewData	该函数用于查询评论是否存在。它通过传入一个包含评论ID的ReviewData对象来执行查询。如果评论存在，则返回该评论对象；否则返回null。

(6) SamePostTypeListService接口

函数名	函数功能	函数返回类型	函数传递参数
getPostListByPostType	根据帖子类型获取帖子列表	samePostTypeListResponse	String type

(7) UserDetailsService接口

函数名称	返回类型	参数	函数功能
getUserDetailByUsername	UserDetails	用户名	根据用户名查询用户详细信息
queryByUsername	UserInfoData	用户名	根据用户名查询用户信息
savePostImages	void	图片和用户名	保存帖子图片并返回处理后的图片路径
updateUserDetailsByUsername	void	用户详细请求和用户名	根据用户名更新用户详细信息
update	UserDetailsResponse	用户详细请求	更新用户详细信息并返回响应结果

(8) UserInfoService接口

函数名称	函数功能	返回类型	参数
update	更新用户信息	UserInfoResponse	UserInfoRequest
query	查询用户信息	UserInfoData	UserInfoData
saveAvatar	保存头像	void	String, String

(9) UserService接口

函数名称	返回类型	传递参数	功能描述
add	UserResponse	User user	添加用户，如果用户名已存在则返回错误信息
query	User	User user	查询用户信息，如果用户不存在则返回null

(10) ActionService接口

函数名称	功能	返回类型	参数
like	处理点赞请求，根据请求类型判断是对帖子还是评论进行点赞操作	ActionResponse	ActionRequest
likePost	对帖子进行点赞操作	ActionResponse	ActionRequest
favor	处理收藏请求，根据请求的取消状态判断是收藏还是取消收藏操作	ActionResponse	ActionRequest, Boolean
likePost	增加帖子的点赞数，并更新帖子信息	void	PostData
query	根据给定的帖子ID查询帖子信息	PostData	PostData
update	更新帖子信息	void	PostData

通信接口说明：

注册(一次请求)

添加一个用户

post请求，请求头：/user/add

`post`：

```
{
  "user": {
    "username": "username",
    "password": "password"
  }
}
```

`response`：

```
{
  "success": true//失败时返回false
  "userResponseFailedReason": null//失败时返回具体失败的原因(可删)
  "user": {
    "username": "username",
    "password": "password"
  }
}
```

```
//枚举类型以枚举名字的字符串返回
//如果用户名存在,则userResponseFailedReason字段的值是"USERNAME_ALREADY_EXIST"
public enum UserResponseFailedReason {
    USERNAME_ALREADY_EXIST,
    USERNAME_DOES_NOT_EXIST,
    WRONG_PASSWORD,
}
```

登录（一次请求）

post请求，请求头：/userInfo/login

`post`:

```
{
  "user": {
    "username": "username",
    "password": "password"
  }
}
```

`response`:

```
{
  "success": true//失败时返回false
  "userResponseFailedReason": null//失败时返回具体失败的原因(可删)
  "userinfo": {
    private String username;
    private String nickname;
    private String personalSign;
    private String avatarUrl;
    private List<String> myPosts;
    private List<String> myCollections;
  }
}
```

```
//枚举类型以枚举名字的字符串返回
//例如如果用户名不存在,则userResponseFailedReason字段的值
是"USERNAME_DOES_NOT_EXIST"
public enum UserResponseFailedReason {
    USER_DOES_NOT_EXIST,
    WRONG_PASSWORD,
}
```

主界面

显示帖子（一次请求）

get请求，请求头：/samePostTypeList/query/{type}

根据分类、页码显示帖子

`response`:

```
{
  "success": true//失败时返回false
  List<Post> postList{
    private String id;
    private String date;
    private String owner;
    private String title;
    private String content;
    private List<String> images;
    private Integer likeNum;
    private List<String> reviews;
    private String PostType;

  }
}
```

帖子总数(无需请求，可以直接在调用显示帖子请求后，直接调用list.length)

帖子详情界面

显示帖子和评论(两次请求)

第一次采用get请求，请求头：post/query/{postId}
根据帖子id，显示帖子详细内容

`response`:

```
{
  private Boolean success;
  private PostResponseFailedReason postResponseFailedReason;
  "Post":
    {
    private String id;
    private String date;
    private String owner;
    private String title;
    private String content;
    private List<String> images;
    private Integer likeNum;
    private List<String> reviews;
    private String PostType;
    }
}
```

第二次采用get请求，显示帖子评论，请求头：/review/query/{reviewId}

`response`:

```
{
 private Boolean success;
 private ReviewResponseFailedReason reviewResponseFailedReason;
  "review":
```

```
    {
     private String id;
    private String targetPost;
    private String date;
    private String username;
    private String content;
    private List<String> images;
    private Integer likeNum;
    }
}
```

点赞和取消点赞帖子（一次请求）

post请求，请求头：/action/like

根据帖子id，点赞帖子

`post`：

```
{
private Boolean isTargetPost;
    private String postId;
    private String reviewId;
    private String username;
}
```

`response`：

```
{
    "success":true
}
```

收藏和取消收藏帖子(一次请求)

请求头：/action/favor（cancelFavor)
根据帖子id，收藏帖子

`post`：

```
{
   private Boolean isTargetPost;
    private String postId;
    private String reviewId;
    private String username;
}
```

`response`：

```
{
    "success":true
}
```

评论帖子（两种格式：一种有图片的评论，一种无图片的评论）（一次请求）

有图片的评论，请求头：/review/add
根据帖子id，评论帖子

`post`:

```
{
    private String id;
    private String targetPost;
    private String username;
    private String content;
    private MultipartFile[] images;
}
```

`response`:

```
{
    private Boolean success;（只需要）
    private ReviewResponseFailedReason reviewResponseFailedReason;
private Review review{
        private String id;
        private String targetPost;
        private String date;
        private String username;
        private String content;
        private List<String> images;
        private Integer likeNum;
    }

}
```

无图片的评论，请求头：/review//addWithoutImage
根据帖子id，评论帖子

`post`:

```
{
    private String id;
    private String targetPost;
    private String username;
    private String content;
    private MultipartFile[] images;
}
```

`response`:

```
{
    private Boolean success;（只需要）
    private ReviewResponseFailedReason reviewResponseFailedReason;
private Review review{
        private String id;
        private String targetPost;
        private String date;
        private String username;
        private String content;
        private List<String> images;
        private Integer likeNum;
    }

}
```

发帖界面

发帖(一次请求)

`post`:

请求头：/post/add

```
{
  private String id; (随便输入一个)
  private String owner;
    private String title;
    private String content;
    private MultipartFile[] images;
    private PostType type;
}
```

`response`:

```
{
    private Boolean success;
    private PostResponseFailedReason postResponseFailedReason;
    private Post post{
        private String id;
        private String date;
        private String owner;
        private String title;
        private String content;
        private List<String> images;
        private Integer likeNum;
        private List<String> reviews;
        private PostType type;

    }

}
```

enum PostResponseFailedReason {
POST_DOES_NOT_EXIST,
POST_ALREADY_EXISTS,
INVALID_TYPE }

个人主页

获取个人主页信息（分两次请求）

第一次get请求，请求头：/UserDetails/query/{username}

`response`：

```
{
    private String username;
    private String gender;
    private String birthday;
    private String registerTime;
    private String signature;
}
```

第二次请求采用get请求，请求头:/userInfo/query/{username}

`response`：

```
{
    private Boolean success;
    private UserInfoResponseFailedReason userInfoResponseFailedReason;
    private UserInfo userInfo{
        private String username;
        private String nickname;
        private String personalSign;
        private String avatarUrl;
        private List<String> myPosts;
        private List<String> myCollections;
    }
}
```

获取个人帖子（）(首先进行一次请求得到该用户的帖子ID列表，之后循环采用get请求进行post实际内容的查询)

第一次get请求，请求头：/userInfo/query/{username}

`response`：

```
{
    private Boolean success;
    private UserInfoResponseFailedReason userInfoResponseFailedReason;
    private UserInfo userInfo{
        private String username;
        private String nickname;
        private String personalSign;
        private String avatarUrl;
        private List<String> myPosts;
        private List<String> myCollections;
    }
}
```

第一次请求中的myPost中包含所有的postId，循环请求的get请求头定义为：/post/query/{postId}

`response`：

```
{
    private Boolean success;
    private PostResponseFailedReason postResponseFailedReason;
    private Post post{
    private String id;
    private String date;
    private String owner;
    private String title;
    private String content;
    private List<String> images;
    private Integer likeNum;
    private List<String> reviews;
    private PostType type;
}

}
```

获取个人收藏(与获取个人帖子一样)

先采用get请求，请求头：/userInfo/query/{username}

`response`：

```
{
    private Boolean success;
    private UserInfoResponseFailedReason userInfoResponseFailedReason;
private UserInfo userInfo{
    private String username;
    private String nickname;
    private String personalSign;
    private String avatarUrl;
    private List<String> myPosts;
    private List<String> myCollections;

    }
}
```

第一次请求中的myCollections中包含所有的postId，循环请求的get请求头定义为：/post/query/{postId}

`response`:

```
{
    private Boolean success;
    private PostResponseFailedReason postResponseFailedReason;
    private Post post{
    private String id;
    private String date;
    private String owner;
    private String title;
    private String content;
    private List<String> images;
    private Integer likeNum;
    private List<String> reviews;
    private PostType type;
}

}
```

获取个人关注(首先进行请求得到目标用户的关注对象与粉丝的id列表，之后进行循环请求根据id查询fans与关注者的详细信息)

get请求，请求头：FansAndConcerned/Toget/{username}

`response`:

```
{
    private Boolean success;
    private Friendship friendship{
        private String username;
        private List<String> myConcerned;
        private List<String> myFans;
        private Integer numConcerned;
        private Integer numFans;
    }

}
```

在进行第一次的请求之后可以得到myConcerned与myFans列表，列表里面的是fans与关注对象的id信息，（对应的是目标用户的username)之后进行循环查询目标用户的具体信息，查询的请求头为：/userInfo/query/{username}

`response`:

```
{
    private Boolean success;
    private UserInfoResponseFailedReason userInfoResponseFailedReason;
    private UserInfo userInfo{
        private String username;
        private String nickname;
        private String personalSign;
        private String avatarUrl;
        private List<String> myPosts;
        private List<String> myCollections;
}

}
```

获取个人粉丝(请求方式与请求关注列表一样，首先请求id列表，之后用id列表进行查询)

、

关注/取关他人（一次请求）

请求头：/Concerned/concerned(cancelconcerned)

`post`：

```
{
    private String concernedId;
    private String username;
}
```

`response`：

```
{
    "success":true
}
```

修改个人信息(一次请求)

请求头：/UserDetails/update

`post`：

```
{
    private String username;
    private String nickname;
    private MultipartFile[] avatar;
    private String birthday;
    private String gender;
    private String signature;
}
```

`response`:

```
{
    "success":true
}
```