

前后端接口

《软件工程（3）》微信抢票开发实战

目录

1	接口规范	2
1.1	基本类	2
1.1.1	API 类	2
1.1.2	异常类	2
1.2	GET 接口	2
1.3	POST 接口	2
2	手机微信页面	3
2.1	学号绑定	3
接口 1.	/api/u/user/bind	3
2.2	活动详情	3
接口 2.	/api/u/activity/detail	3
2.3	电子票详情	4
接口 3.	/api/u/ticket/detail	4
3	后台管理页面	4
3.1	管理员登录	4
接口 4.	/api/a/login	4
接口 5.	/api/a/logout	4
3.2	活动列表	5
接口 6.	/api/a/activity/list	5
接口 7.	/api/a/activity/delete	5
3.3	创建活动	5
接口 8.	/api/a/activity/create	5
接口 9.	/api/a/image/upload	6
3.4	活动详情	6
接口 10.	/api/a/activity/detail	6
3.5	微信抢票菜单调整	7
接口 11.	/api/a/activity/menu	7
3.6	检票	8
接口 12.	/api/a/activity/checkin	8

1 接口规范

1.1 基本类

1.1.1 API 类

项目框架中封装了 `APIView` 类，此类作为所有前后端接口的基类。以下任何接口的实现，都是继承该类的子类。学号绑定的接口已经通过子类 `userpage.views.UserBind` 部分实现，以供参考。

`APIView` 类提供了 `check_input` 方法，可用于检查输入参数的完整性，建议对每个接口都检查其必须的参数。

任何前后端接口，都进行了返回数据的格式规范化，任何一个前后端接口，返回的数据均为 `application/json` 格式的对象，包含 `code`、`msg`、`data` 三个字段，其中 `code` 字段为错误代码（无错误时为 0），`msg` 为错误信息（无错误时空串），`data` 为真实返回的数据。在 `APIView` 的子类中，具体的 `GET` 或 `POST` 接口只需要返回 `data` 即可，其他包括异常（`Exception`）的捕获、`JSON` 的序列化等，都已经做好了封装。

1.1.2 异常类

在 `codex.baseerror` 中定义了若干常见的错误，它们均为 `Exception` 的子类且带有 `code` 成员，可以在 `APIView` 子类的 `GET` 或 `POST` 接口中遇到处理错误时抛出这些异常，就能向前端返回相应的错误代码和错误信息。

目前已经定义的错误包括 `InputError`（参数错误）、`LogicError`（逻辑错误）、`ValidateError`（校验错误）等，你也可以定义新的异常子类，只需继承 `BaseError` 即可。

1.2 GET 接口

`GET` 接口只能用于获取数据，所有用于修改数据的均应使用 `POST` 接口。

在 `APIView` 的子类中，要支持 `GET` 接口，只需要实现 `get` 方法即可。`GET` 接口的参数通过 `URL` 参数给出，在 `APIView` 的子类对象中可以访问 `self.input` 来获取，是一个 `dict` 对象，例如需要获取 `GET` 的参数 `openid`，访问 `self.input['openid']` 即可。

1.3 POST 接口

`POST` 接口只能用于修改数据，所有用于获取数据的均应使用 `GET` 接口。

在 `APIView` 的子类中，要支持 `POST` 接口，只需要实现 `post` 方法即可。`POST` 接口的参数通过相应请求（`request`）的 `body` 给出，在 `APIView` 的子类对象中可以访问 `self.input` 来获取，是一个 `dict` 对象。例如需要获取 `POST` 的参数 `name`，访问 `self.input['name']` 即可。此外，通过 `self.input` 也可以获取通过 `POST` 提交的文件，例如需要获取 `POST` 中字段 `image` 对应的文件，访问 `self.input['image']` 即可。

2 手机微信页面

2.1 学号绑定

相关页面: /u/bind

URL 参数: openid 为微信用户的 OpenID。

接口 1. </api/u/user/bind>

GET: 获取当前用户的学号绑定状态

请求参数 openid 为微信用户的 OpenID。

返回数据为该用户已绑定的学号（字符串），若未绑定则为空串。

POST: 将当前用户绑定至指定学号

请求参数 openid 为微信用户的 OpenID，student_id 为将绑定的学号，password 为密码。

若绑定成功，则返回 code=0，否则返回 code 非 0。

你只需要实现该接口的 validate_user 方法，该方法中对学号和密码进行验证，若验证失败，在其中抛出 ValidateError 异常即可。

2.2 活动详情

相关页面: /u/activity

URL 参数: id 为活动 ID。

接口 2. </api/u/activity/detail>

GET: 获取活动的详情

请求参数 id 为活动 ID。

只有活动处于已发布状态（status==1）才可返回该活动的详情数据，各字段及其含义如下表所示。其他状态下都应返回 code 非 0。

字段	含义	备注
name	活动名称	
key	活动代称	
description	活动描述	
startTime	活动开始时间	时间戳，精确到秒
endTime	活动结束时间	时间戳，精确到秒
place	活动地点	
bookStart	抢票开始时间	时间戳，精确到秒
bookEnd	抢票结束时间	时间戳，精确到秒
totalTickets	活动预计发放总票数	

picUrl	活动配图 URL	
remainTickets	当前剩余票数	
currentTime	服务器当前时间	时间戳，精确到秒

2.3 电子票详情

相关页面: /u/ticket

URL 参数: openid 为微信用户的 OpenID, ticket 为代表电子票的 unique_id。

接口 3. </api/u/ticket/detail>

GET: 获取电子票的详情

请求参数 openid 为微信用户的 OpenID, ticket 为代表电子票的 unique_id。

返回数据为电子票的详情数据, 各字段及其含义如下表所示。

字段	含义	备注
activityName	活动名称	
place	活动地点	
activityKey	活动代称	
uniqueId	电子票的唯一 ID	
startTime	活动开始时间	时间戳, 精确到秒
endTime	活动结束时间	时间戳, 精确到秒
currentTime	服务器当前时间	时间戳, 精确到秒
status	电子票状态	0 为无效票 (例如已退票), 1 为未使用, 2 为已使用

3 后台管理页面

3.1 管理员登录

相关页面: /a/login

接口 4. </api/a/login>

GET: 获取登录状态

请求参数为空即可。若已登录, 则返回 code=0, 否则返回 code 非 0。

POST: 通过用户名和密码登录

请求参数为 username 和 password 分别为用户名和密码。

若登录成功, 返回 code=0, 否则返回 code 非 0。

接口 5. </api/a/logout>

POST: 登出

请求参数为空即可。若登出成功, 返回 code=0, 否则返回 code 非 0。

3.2 活动列表

相关页面: </a/activity/list>

接口 6. </api/a/activity/list>

GET: 获取活动列表

该接口需要已登录才可使用。

请求参数为空即可。

返回数据为活动对象的数组 ($status \geq 0$)，每个活动的字段及其含义如下表所示。

字段	含义	备注
id	活动 ID	数据库 ID
name	活动名称	
description	活动描述	
startTime	活动开始时间	时间戳，精确到秒
endTime	活动结束时间	时间戳，精确到秒
place	活动地点	
bookStart	抢票开始时间	时间戳，精确到秒
bookEnd	抢票结束时间	时间戳，精确到秒
currentTime	服务器当前时间	时间戳，精确到秒 简单起见，每个活动都要提供该字段
status	活动状态	0 代表未发布，1 代表已发布

接口 7. </api/a/activity/delete>

POST: 删除活动

请求参数 id 为活动 ID。若删除成功，返回 code=0，否则返回 code 非 0。

3.3 创建活动

相关页面: </a/activity/detail?create=1>

接口 8. </api/a/activity/create>

POST: 提交活动数据以创建活动

该接口需要已登录才可使用。

请求参数如下表所示。

字段	含义	备注
name	活动名称	
key	活动代称	
place	活动地点	
description	活动描述	

picUrl	活动配图 URL	
startTime	活动开始时间	时间戳，精确到秒
endTime	活动结束时间	时间戳，精确到秒
bookStart	抢票开始时间	时间戳，精确到秒
bookEnd	抢票结束时间	时间戳，精确到秒
totalTickets	总票数	
status	暂存或发布	0 代表暂存，1 代表发布

若创建成功，返回 code=0 且 data 为成功创建的活动 ID，否则返回 code 非 0。

接口 9. </api/a/image/upload>

POST: 上传图像并保存至服务器

该接口需要已登录才可使用。

请求参数只有一个 image 字段，为上传的图片文件。

若上传并保存成功，返回 code=0 且 data 为该图片的 URL（含域名），否则返回 code 非 0。

3.4 活动详情

相关页面：/a/activity/detail

URL 参数：id 为活动的 ID。

接口 10. </api/a/activity/detail>

GET: 获取活动详情

该接口需要已登录才可使用。

请求参数为 id 为活动的 ID。

返回数据为活动的详情数据，各字段及其含义如下表所示。

字段	含义	备注
name	活动名称	
key	活动代称	
description	活动描述	
startTime	活动开始时间	时间戳，精确到秒
endTime	活动结束时间	时间戳，精确到秒
place	活动地点	
bookStart	抢票开始时间	时间戳，精确到秒
bookEnd	抢票结束时间	时间戳，精确到秒
totalTickets	活动预计发放总票数	
picUrl	活动配图 URL	
bookedTickets	当前已被抢的票数	
usedTickets	当前已使用的票数	
currentTime	服务器当前时间	时间戳，精确到秒

status	活动状态	0 代表未发布，1 代表已发布
--------	------	-----------------

POST: 修改活动详情

该接口需要已登录才可使用。

请求参数如下表所示。

字段	含义	备注
id	活动 ID	
name	活动名称	已发布的活动不可修改
place	活动地点	已发布的活动不可修改
description	活动描述	
picUrl	活动配图 URL	
startTime	活动开始时间	时间戳，精确到秒 活动结束后不可修改
endTime	活动结束时间	时间戳，精确到秒 活动结束后不可修改
bookStart	抢票开始时间	时间戳，精确到秒 已发布的活动不可修改
bookend	抢票结束时间	时间戳，精确到秒 活动开始后不可修改
totalTickets	总票数	抢票开始后不可修改
status	暂存或发布	0 代表暂存，1 代表发布。 已发布的活动只能立即发布而不能暂存。

若修改成功，返回 code=0，否则返回 code 非 0。

3.5 微信抢票菜单调整

相关页面：/a/activity/menu

接口 11. [/api/a/activity/menu](#)

GET: 获取当前微信抢票菜单

该接口需要已登录才可使用。

请求参数为空即可。

返回数据为当前所有可加入菜单的活动对象数组，与接口 6 不同的是，活动对象的字段有所差异，具体如下表所示。

字段	含义	备注
id	活动 ID	
name	活动名称	
menuIndex	该活动当前在抢票菜单中的位置	0 代表不在菜单中，正整数代表具体位置

POST: 修改微信抢票菜单

该接口需要已登录才可使用。

请求参数为加入菜单的活动 id 的数组。

若修改成功，返回 code=0，否则返回 code 非 0。

3.6 检票

相关页面: /a/activity/checkin

URL 参数: id 为活动 ID。

接口 12. /api/a/activity/checkin

POST: 检票

该接口需要已登录才可使用。

请求参数如下表所示。

字段	含义	备注
actId	活动 ID	
ticket	电子票的唯一 ID	与 studentId 只提交一个
studentId	学号	与 ticket 只提交一个

若检票成功，返回电子票信息如下表所示，否则返回 code 非 0。

字段	含义	备注
ticket	电子票的唯一 ID	
studentId	学号	