

# BZLine-App 数据交互接口说明

---

## 目录

- 认证接口
- 业务数据
- 其他说明

## I. 认证接口

### - 注册

1, 请求接口地址: `http://app.bzline.cn/account/manager/`

2, 请求方法 POST 数据到以上地址, 数据内容以JSON格式封装

客户端 **post** 内容:

```
{'action': 'register',  
  'email': 'user email address', //用户输入的邮箱地址  
  'password': 'password', //注册密码  
  'infomation': {'type': 'vaaa'....}' //infomation的值是一个json格式的可选择数据,  
                包括用户信息, 将来要用  
}
```

服务器端 **response** 内容:

```
{'status': '0', //服务器端返回的状态, 如果为0, 表示注册成功, 从tokenid字段获取  
  凭证token。返回 '-1'表示失败, 具体错误消息参考下面的error_no字  
  段
```

```
'error_no': '401', //401临时性错误, 请重试; 402, 未知错误, 请报告管理员
```

```
    //403 数据为空或者数据格式错误
```

```
    //404 邮件字段不存在,
```

```
    //405 密码字段不存在
```

```
    //406 密码格式错误
```

```
    //407 邮件格式错误
```

```
    //501 该邮箱地址已经被注册, 请重新选择邮箱
```

```
'message': '返回的显示信息', //用于显示给用户的信息, 可选择
```

2015年10月15日 星期四

'tokenid': 'token string (32)' //注册成功后(status 返回0), 返回令牌, 用于用户直接

//登录不用再输入用户名和密码

//令牌固定长度为32位字节的字符串组成

}

使用场景: 新用户点击注册, 进入注册流程的时候需要调用本接口

测试用例: 终端上用curl模拟client提交数据

1, curl -I -H "Content-type: application/json" -X POST -d  
'{"action": "register", "email": "stonebzline.cn", "password": "test123"}' http://  
app.bzline.cn/account/manager/

2, 服务器response:

{"status": "-1", "error\_no": "407", "message": "register fail!"}

-----

1, curl -I -H "Content-type: application/json" -X POST -d  
'{"action": "register", "email": "stone@bzline.cn", "password": "test123"}' http://  
app.bzline.cn/account/manager/

2, 服务器response:

{"status": "0"}

-----

1, curl -I -H "Content-type: application/json" -X POST -d  
'{"action": "register", "email": "stone@bzline.cn", "password": "test123"}' http://  
app.bzline.cn/account/manager/

2, 服务器response:

{"status": "-1", "error\_no": "501", "message": "register fail!"}

## - 登录

1, 请求接口地址: <http://app.bzline.cn/account/manager/>

2, 请求方法Post数据, 数据json封装

客户端请求**post**内容:

本地不存在tokenid的用户(包括第一次登陆的新用户), 需要通过用户名和密码进行认证

```
{'action': 'login',  
  'email': 'user email',  
  'password': 'password'  
}
```

服务器端**response**内容:

{'status': '0', //登录认证成功 , -1 认证失败

'error\_no': '401', //401临时错误, 请重试; 402, 未知错误, 请报告管理员

//404 邮件字段不存在,

//405 密码字段不存在

//406 密码格式错误

//407 邮件格式错误

//501 用户名或者密码错误

```
'message': 'response msg', //服务器端返回的消息，显示给用户
'tokenid': 'token string(32)' //当状态为0的时候，返回tokenid凭证
}
```

认证成功后，本地保存tokenid 和email字段。服务器端通信的时候通过这2者来验证客户端合法性。

使用场景：新用户登陆，或者本地**tokenid** 过期的时候，会进入登陆界面，此时需要调用本接口

```
curl -I -H "Content-type: application/json" -X POST -d
'{"action": "login", "email": "stone@bzline.cn", "password": "test123"}' http://
app.bzline.cn/account/manager/
{"status": "0", "tokenid": "9bf84fda0009b18550124d0dcfddd6ab"}
```

```
curl -I -H "Content-type: application/json" -X POST -d
'{"action": "login", "email": "stone@bzline.cn", "password": "test123asd"}' http://
app.bzline.cn/account/manager/
{"status": "-1", "error_no": "501", "message": "auth failed"}
```

#### - 验证**tokenid**有效性

请求接口地址: <http://app.bzline.cn/account/manager/>

POST数据:

```
{'action': 'verify_tokenid',
'oauth_ower': 'W' //如果是第三方的登陆，则需要有本字段，否则不用本字段
'userid': 'uid string ', //这里的userid 就是用户注册和登陆用的邮件地址
```

//如果是通过OAuth第三方登陆的，这里就是第三方

//的user id

```
'tokenid','token string(32)'  
}
```

服务器端返回数据：

{'status':'0', //验证成功，-1验证失败

'error\_no':'401', //401临时错误，请重试；402，未知错误，请报告管理员

//403 tokenid字段不存在

//404 userid字段不存在

//405 tokenid 格式错误

//501 tokenid 无效

```
'message':'session过期'  
}
```

使用场景：用户从手机上打开应用的时候，如果检查到本地存在**tokenid**，此时调用本接口进行验证，而不需要进入登陆界面。

或者其他关键操作，需要和服务器通信的时候，需要调用本接口进行合法性验证。（暂时不用）

```
curl -I -H "Content-type: application/json" -X POST -d  
'{"action":"verify_tokenid","userid":"stone@bzline.cna","tokenid":"9bf84fda0009b  
18550124d0dcfddd6ab"}' http://app.bzline.cn/account/manager/  
{ "status": "-1", "error_no": "501", "message": "tokenid is invalid" }
```

-----

```
curl -I -H "Content-type: application/json" -X POST -d
'{"action":"verify_tokenid","userid":"stone@bzline.cn","tokenid":"9bf84fda0009b1
8550124d0dcfddd6ab"}' http://app.bzline.cn/account/manager/
```

```
{"status": "0", "tokenid": "auth success"}
```

-----

```
curl -I -H "Content-type: application/json" -X POST -d
'{"action":"verify_tokenid","oauth_ower":"W","userid":"stone@qq.com","tokenid"
:"6ad70c8f9fd32678c96d847cbae72556"}' http://app.bzline.cn/account/manager/
```

```
{"status": "-1", "error_no": "501", "message": "tokenid is invalid"}
```

-----

```
curl -I -H "Content-type: application/json" -X POST -d
'{"action":"verify_tokenid","oauth_ower":"W","userid":"stone@qq.com","tokenid"
:"0ea09261af6169d9b9e5cc42e8ee59e2"}' http://app.bzline.cn/account/manager/
```

```
{"status": "0", "tokenid": "auth success"}
```

## - 社交账号登陆

1,通过各个社交平台的开放接口进行认证过程，认证通过后获得 社交平台颁发的 access\_token

2, 通过access\_token 获得社交平台的用户id

3, 请求调用接口:http://app.bzline.cn/account/manager/

post 以下json:

```
{
```

```

'action':'oauth',
'oauth_ower': 'W', //W =weibo, Q = qq, X =weixin
'access_token': RSA(token string)' //第三方颁发的token 凭证被加密后的字符串，该字符串是用非对称RSA算法的公钥加密后的内容
//服务器端用私钥解密，确保该凭证的有效性，防止模仿客户端登陆。
//测试服务器暂时未用RSA加密，用明文测试即可

'access_id': 'user id' ,//该用户在第三方的用户名称
}

```

服务器response 以下json:

```

{'status': '0', //0 成功, -1 失败
'error_no': '401', //401 临时错误，请重新试； 402 意外错误，请联系管理员
//501 凭证解密失败，可能该凭证是假的
'message': 'this is message to user',
'tokenid': 'token string(32)' //颁发的正式凭证
}

```

应用场景： 当使用社交账号登陆的时候，需要调用本接口流程。

```

curl -I -H "Content-type: application/json" -X POST -d
'{"action": "oauth", "oauth_ower": "W", "access_token": "thrid token", "access_id": "stone@qq.com"}' http://app.bzline.cn/account/manager/
{"status": "0", "tokenid": "851eed069bbe4ec38faf477cb2701b1"}

```

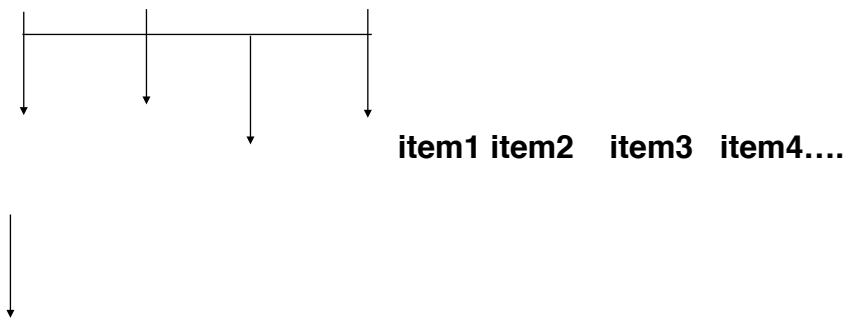
## II. 业务数据

业务数据的层次结构为：





类别数据 {类别名称, 类别图标}



| item 名称 | 数值 | 参考数值 | 标记 |
|---------|----|------|----|
|         |    |      |    |
|         |    |      |    |
|         |    |      |    |

存在2种数据：

- 1, 类别数据, 只包含类别名称 和 类别图标
- 2, 条目数据 (item), 包含一个表格数据和其他说明数据
- 3, 一个类别数据由很多item组成

接口说明：

- 获取类别数据

接口地址： <http://app.bzline.cn/uassay/managerCategory/>

客户端post请求数据，json格式：

```
{  
  'action': 'get_category',  
  'tokenid': 'token string(32)',  
  'fetch_count': 0, //这2个字段用来处理分页，大数据用，目前暂时不用  
  'start_offset': 0 ,  
  'userid': 'user id',  
  'oauth_ower': 'W'  
}
```

tokenid 和userid 是认证成功后保存在本地的凭证。

服务器response数据，json格式：

```
{'status': '0', //0成功返回， -1失败  
  'error_no': '401', //401临时错误， 501 凭证验证失败  
  'category_count': 2, //类别数量  
  'datas': [ //类别具体内容， 一个json数组  
    {  
      'id': 1,  
      'name': 'name string',  
      'imageid': 'image_png_01'  
    }, //1条数据  
    {  
      'id': 10, //category id  
      'name': 'name string', //category name  
      'imageid': 'image_png_01' //category image name  
    }, //第2条数据  
  ]  
}
```

应用场景：点击（进入）app首页的时候，显示用户当前的所有类别数据，需要调用本接口

```
curl -I -H "Content-type: application/json" -X POST -d
 '{"action": "get_category", "oauth_ower": "W", "tokenid": "0ea09261af6169d9b9e5cc42e8ee59e2", "userid": "stone@qq.com"}' http://app.bzline.cn/uassay/managerCategory/

{"status": "0", "category_count": 2, "datas": "[{"imageid": "imageNameld", "id": 1, "name": "categoryName"}, {"imageid": "imageNameld", "id": 2, "name": "categoryName1"}]"}

```

#### - 添加类别接口

接口地址： <http://app.bzline.cn/uassay/managerCategory/>

客户端post请求数据，json格式：

```
{
  'action': 'create_category',
  'tokenid': 'token string(32)',
  'userid': 'user id',
  'oauth_ower': 'W' //如果是第三方登陆，本字段用来指明是那个第三方, W = 'WEIBO'
                    //X = 'WEIXIN', Q = 'QQ'
  'category_name': 'category name', //用户需要创建的类别名称
  'category_image_id': 'category image id', //用户选择的图标id
}
```

tokenid 和userid 是认证成功后保存在本地的凭证。

服务器response数据，json格式：

```
{'status': '0', //0成功返回， -1失败
  'error_no': '401', //401临时错误， 501 凭证验证失败
  'name': 'category_name', //创建的类别名称
  'id': 'category id', //创建的类别id
  'imageid': 'category image id', 创建的类别图标id
}
```

服务器返回很多提交时候的数据内容，方便支持异步操作。

应用场景：添加类别的时候需要调用本接口

测试用例：

```
curl -I -H "Content-type: application/json" -X POST -d
 '{"action": "create_category", "oauth_ower": "W", "tokenid": "xxxx", "userid": "stone@qq.com",
  "category_name": "categoryName", "category_image_id": "imageNameld"}' http://
app.bzline.cn/uassay/managerCategory/
```

```
{"status": "-1", "error_no": "405", "message": "token id is invalid"}
```

=====

```
curl -I -H "Content-type: application/json" -X POST -d
 '{"action": "create_category", "oauth_ower": "W", "tokenid": "0ea09261af6169d9b9e5cc42e8
ee59e2", "userid": "stone@qq.com", "category_name": "categoryName1", "category_image
_id": "imageNameld"}' http://app.bzline.cn/uassay/managerCategory/
```

```
{"status": "0", "category_image_id": "imageNameld", "category_id": 2,
  "category_name": "categoryName1"}
```

=====

```
curl -I -H "Content-type: application/json" -X POST -d
 '{"action": "create_category", "oauth_ower": "W", "tokenid": "0ea092c42e8ee59e2", "userid":
stone@qq.com", "category_name": "categoryName", "category_image_id": "imageNameld"
}' http://app.bzline.cn/uassay/managerCategory/
```

```
{"status": "-1", "error_no": "504", "message": "The category name already exists"}
```

## 条目(item)相关接口

### - 创建item 过程

- 1, 用户在**App (ios)**上点开照相机, 拍照, 然后提交照片 (客户端提交操作)
- 2, 服务器返回成功或者失败 (接收图片) 的消息, 并在数据库中增加一条**item**记录
- 3, 客户端利用 ‘获取指定类别下的所有条目(**item**)名称’ 接口刷新客户端的**Item**列表界面

注意: 客户端提交图片后, 服务器只是返回是否接收到图片的消息, 不会同步返回**item**的信息。所以, 必须要调用接口进行刷新**item**列表, 而且有可能在一定时间内, 并不能刷出新的**item**(服务器处理慢...或者其他问题), 提交图片和获得**item**信息是一个异步的过程。

### - 获取指定类别下的所有条目(item)名称

接口地址: <https://www.bzline.cn/app/managerCategory>

客户端post请求数据, json格式:

```
{  
  'action': 'get_category_item_list',  
}
```

```

'tokenid': 'token string(32)'
'userid': 'user id',
'oauth_ower': '',
'category_id': 'category id', //需要获取的category id
'fetch_count': 0, //这2个字段用来处理多页, 大数据用, 目前暂时不用
'start_offset': 0,
}

```

服务器response数据, json格式:

```

{'status': '0', //0成功返回, -1失败
 'error_no': '401', //401临时错误, 501 凭证验证失败
 'item_count': 2, //条目(item)数量
 'datas': [ //条目具体内容, 一个json数组
     {'item_id': 1, 'item_name': 'name string'}, //1条数据
     {'item_id': 2, 'item_name': 'name string'}, //第2条数据
 ]
}

```

应用场景: 列出某一个类别下的所有条目列表

```

curl -I -H "Content-type: application/json" -X POST -d
{'action': 'get_category_item_list', 'oauth_ower': '', 'tokenid': '6e04addacad6322c9d8da8
f0bbb56b26', 'userid': 'aa@aa.aa', 'category_id': 3} http://app.bzline.cn/uassay/
managerItems/

```

```

{"status": "0", "datas": "[{"date": "2015-10-22 13:37:50", "notes": "test notes",
 "address": "shanghai", "id": 1, "name": "item_name_001"}, {"date": null, "notes":
 "notes 001", "address": "beij", "id": 3, "name": "item_002"}, {"date": null, "notes
 ": "notes 003", "address": "beij", "id": 4, "name": "item_003"}]", "item_count": 3}

```

```
{"status": "0", "datas": "[]", "item_count": 0}
```

```
{"status": "-1", "error_no": "405", "message": "token id is invalid"}
```

- 获取指定条目**id(item\_id)**的详细内容

接口地址：<https://www.bzline.cn/app/managerItems>

客户端post请求数据，json格式：

```
{  
  'action': 'get_item_detail',  
  'tokenid': 'token string(32)',  
  'userid': 'user id',  
  'oauth_ower': '',  
  'item_id': 'item id', //需要获取的item id  
}
```

服务器response数据，json格式：

```
{'status': '0', //0成功返回, -1失败  
  'error_no': '401', //401临时错误, 501 凭证验证失败  
  'datas': {'item_id': 1,  
            'item_name': 'name string',  
            'item_date': 'date',  
            'item_address': 'address string',  
            'item_notes': 'notes content'}
```

```
        'table_datas':[ //表格数据
        { 'field_name':'name','field_value':'value','field_advance_value':'ad value','mark':'mark
        string'},
        { 'field_name':'name','field_value':'value','field_advance_value':'ad value','mark':'mark
        string'},
        { 'field_name':'name','field_value':'value','field_advance_value':'ad value','mark':'mark
        string'},

        ]

    }

}
```

#### - 修改item内容接口

接口地址：<https://www.bzline.cn/app/managerItems>

客户端post请求数据，json格式：

```
{
    'action':'update_item_detail',
    'tokenid':'token string(32)'
    'userid':'user id',
    'item_id ':'item id', //需要修改的item id
    'datas':{ //修改后的内容
        'item_name':'name string',
        'item_date': 'date',
        'item_address':'address string',
        'item_notes': 'notes content'
        'table_datas':[ //表格数据
        { 'field_name':'name','field_value':'value','field_advance_value':'ad value','mark':'mark
        string'},
        { 'field_name':'name','field_value':'value','field_advance_value':'ad value','mark':'mark
        string'},
        { 'field_name':'name','field_value':'value','field_advance_value':'ad value','mark':'mark
        string'},

        ]

    }

}
```



```
{'field_name':'name','field_value':'value','field_advance_value':'ad value','mark':'mark
string'},
{'field_name':'name','field_value':'value','field_advance_value':'ad value','mark':'mark
string'},

    ]

    }//end datas

}
```

服务器response数据，json格式：

```
{'status': '0', //0成功返回， -1失败
  'error_no': '401', //401临时错误， 501 凭证验证失败
  'message': 'show message',
  'item_id': 'item id', //成功被修改的itemid
}
```

应用场景： 当用户编辑条目(**item**)内容，需要确认保存提交的时候调用本接口。

### III. 其他说明