

SubwayTraffic Platform v0.3.0接口文档

登录

URL:

```
1 /system/v1/login
```

请求类型: POST

请求body示例:

```
1 {
2   "username": "admin",
3   "password": "sdfadew&2"
4 }
```

| 属性名称 | 属性类型 | 属性说明 |
|----------|--------|------|
| username | string | 用户名 |
| password | string | 密码 |

返回值:

1、200

登录成功。

```
1 {
2   "login": true,
3   "token": "qDsJwYjz0N",
4   "code": 200
5 }
```

返回的token值随机。

2、400

下发的请求body有误。

```
1 {
2   "login": false,
3   "token": null,
4   "error": "BadRequest: Invalid username or password.",
5   "code": 400
6 }
```

3、403

场景一、服务器理解登录请求，但用户已登录成功，不必重复登录。

```
1 {  
2   "login": false,  
3   "token": null,  
4   "error": "User logged in.",  
5   "code": 403  
6 }
```

场景二、下发的用户名或密码错误。

```
1 {  
2   "login": false,  
3   "token": null,  
4   "error": "Wrong username or password.",  
5   "code": 403  
6 }
```

4、404

请求登录的用户名不存在。

```
1 {  
2   "login": false,  
3   "token": null,  
4   "error": "can not find user demo2",  
5   "code": 404  
6 }
```

5、405

使用错误的请求类型。

6、406

请求格式有误，不能body体不能被解码。

```
1 {  
2   "error": "invalid POST request: JSON decode failed.",  
3   "code": 406  
4 }
```

6、500

服务器拒绝访问请求，服务器发生内部错误。

退出登录

URL:

```
1 /system/v1/logout
```

请求类型: GET

请求键值对:

| key | value | 说明 |
|----------|-------|---------------------------------------|
| username | admin | 请求退出登录的用户名，这里以admin做例。value值是string类型 |

header:

| key | value | 说明 |
|-------|------------|---------------|
| token | sdfrgersa2 | 用户此次登录的token值 |

返回值:

1、200

用户退出登录成功。

```
1 {
2   "logout": true,
3   "code": 200
4 }
```

2、400

请求键值对格式有误。

```
1 {
2   "logout": false,
3   "error": "BadRequest: Invalid username.",
4   "code": 400
5 }
```

3、401

没有操作权限，服务器拒绝访问请求。

```
1 {
2   "logout": false,
3   "error": "limited authority",
4   "code": 401
5 }
```

4、405

使用错误的请求类型。

4、500

服务器拒绝访问请求，服务器发生内部错误。

获取版本号

URL：

```
1 /system/v1/version
```

请求类型：GET

请求键值对：无

header：

| key | value | 说明 |
|-------|------------|---------------|
| token | sdfrgersa2 | 用户此次登录的token值 |

返回值：

1、200

用户获取版本号成功。

```
1 {
2   "version": "SubwayTraffic Platform v0.1.0.B020",
3   "code": 200
4 }
```

具体根据版本返回的version值有变化。

2、401

没有操作权限，服务器拒绝访问请求。

用户登录成功后权限放开。

```
1 {
2   "version": nil,
3   "error": "limited authority",
4   "code": 401
5 }
```

3、500

服务器拒绝访问请求，服务器发生内部错误。

4、503

服务器内部配置文件被破坏，无法完成访问请求。

```
1 {
2   "version": nil,
3   "can not found file: stp.version",
4   "code": 503
5 }
```

获取用户列表

URL

```
1 /user/v1/users
```

请求类型 GET

请求键值对 无

header

| key | value | 说明 |
|-------|------------|---------------|
| token | sdfrgersa2 | 用户此次登录的token值 |

返回值

1、200

获取用户列表成功。

```
1 {
2   "users": [
3     {
4       "username": "demo",
5       "token": "qw2fvo96se",
6       "create": "2019-11-15 10:20:23"
7     }
8   ]
9 }
```

2、401

没有操作权限，服务器拒绝访问请求。

仅对成功登录的admin用户拥有权限。

```
1 {
```

```
2   "users": [],
3   "error": "limited authority"
4 }
```

3、500

服务器拒绝访问请求，服务器发生内部错误。

获取session

URL

```
1 /system/v1/session
```

请求类型 GET

请求键值对 无

header

| key | value | 说明 |
|-------|------------|---------------|
| token | sdfrgersa2 | 用户此次登录的token值 |

返回值

1、200

获取用户列表成功。

```
1 {
2   "session": {
3     "_permanent": true,
4     "tuEI0tFq": "admin"
5   },
6   "code": 200
7 }
```

2、401

没有操作权限，服务器拒绝访问请求。

```
1 {
2   "error": "limited authority",
3   "code": 401
4 }
```

3、500

服务器拒绝访问请求，服务器发生内部错误。

用户注册

URL

```
1 /user/v1/register
```

请求类型 POST

请求body示例：

```
1 {
2   "username": "demo",
3   "password": "sdfadew&2"
4 }
```

| 属性名称 | 属性类型 | 属性说明 |
|----------|--------|------|
| username | string | 用户名 |
| password | string | 密码 |

返回值

1、200

用户注册成功。

```
1 {
2   "success": "registered user demo success"
3 }
```

2、400

下发参数有误。

```
1 {"error": "Badrequest: Invalid param"}
```

3、403

服务器理解客户端请求，但拒绝执行。原因包括：用户名已存在导致注册失败；密码长度不大于8。

```
1 {"error": "The username admin already exist"}
```

或

```
1 {"error": "Password length must more than 8."}
```

4、406

请求格式有误，不能body体不能被解码。

```
1 {"error": "invalid POST request: JSON decode failed."}
```

4、500

服务器内部错误，无法执行请求。

5、503

数据库插入数据错误。

```
1 {"error": "Register user ERROR: xxx"}
```

删除用户

URL

```
1 /user/v1/delete
```

请求类型 **DELETE**

请求body示例：

```
1 {
2   "username": "demo"
3 }
```

| 属性名称 | 属性类型 | 属性说明 |
|----------|--------|------|
| username | string | 用户名 |

header

| key | value | 说明 |
|-------|------------|---------------|
| token | sdfrgersa2 | 用户此次登录的token值 |

返回值

1、200

删除成功

```
1 {"success": "delete user demo success"}
```

2、400

下发的删除请求参数不正确。

```
1 {"error": "Badrequest: Invalid param"}
```


3、401

没有权限。删除操作必须由admin用户完成。

```
1 {"error": "limited authority"}
```

或

不能对admin账户进行删除。

```
1 {"error": "can not delete admin user"}
```

4、404

删除一个不存在的用户。

```
1 {"error": "can not find user admin"}
```

4、405

请求类型错误。

5、406

请求的body体格式有误，无法解析。

```
1 {"error": "invalid DELETE request: JSON decode failed."}
```

5、500

服务器内部报错，无法完成请求。

修改用户密码

URL

```
1 /user/v1/modify/password
```

请求类型 PUT

请求body示例：

```
1 {
2   "username": "demo",
3   "password": "demo2479",
4   "new_password": "wfmuf."
5 }
```

| 属性名称 | 属性类型 | 属性说明 |
|------|------|------|
|------|------|------|

| | | |
|--------------|--------|---------|
| username | string | 用户名 |
| password | string | 用户密码 |
| new_password | string | 要修改的新密码 |

返回值

1、200

修改密码成功

```
1  {"success": "change password success."}
```

2、400

请求参数有误。

```
1  {"error": "BadRequest: Invalid param"}
```

3、403

下发的用户名或密码错误。

```
1  {"error": "Wrong username or password."}
```

下发的新密码过短，小于8位长度。

```
1  {"error": "Password length must more than 8."}
```

4、405

请求类型错误。

5、406

请求的body体格式有误，无法解析。

```
1  {  
2    "error": "invalid PUT request: JSON decode failed.",  
3  }
```

5、500

服务器内部错误。

增加地铁线路

URL

```
1 /line/v1/add
```

请求类型 POST

请求body示例：

```
1 {
2   "name": "line-1"
3 }
```

| 属性名称 | 属性类型 | 属性说明 |
|------|--------|-------|
| name | string | 地铁线路名 |

返回值

1、200

增加地铁线路成功。

```
1 {
2   "success": "add subway line line-1 success.",
3   "code": 200
4 }
```

2、400

请求参数有误。

```
1 {
2   "error": "BadRequest: Invalid subway name.",
3   "code": 400
4 }
```

3、401

用户没有登录或不是admin用户。

```
1 {
2   "error": "limited authority",
3   "code": 401
4 }
```

4、403

添加的地铁线路名称已存在，服务器拒绝再次添加。

```
1 {
2   "error": "subway line line-1 has been exist.",
3   "code": 403
4 }
```

4、405

请求类型有误。

5、406

请求的body体格式有误，无法解析。

```
1 {
2     "error": "invalid POST request: JSON decode failed.",
3     "code": 406
4 }
```

6、500

服务器内部错误。

删除地铁线路

URL

```
1 /line/v1/delete
```

请求类型 DELETE

请求body示例：

```
1 {
2     "uuid": "2W1R01UI-5JUR-BWM3-PGT2D3XS"
3 }
```

| 属性名称 | 属性类型 | 属性说明 |
|------|--------|----------|
| uuid | string | 地铁线路uuid |

返回值

1、200

删除地铁线路成功。

```
1 {
2     "success": "delete subway line CPYQKS79-ULD9-7QAK-ZFBWF35E success.",
3     "code": 200
4 }
```

2、400

请求参数有误。

```
1 {  
2   "error": "BadRequest: Invalid subway uuid.",  
3   "code": 400  
4 }
```

3、401

用户没有登录或不是admin用户。

```
1 {  
2   "error": "limited authority",  
3   "code": 401  
4 }
```

4、404

所要删除的uuid不存在。

```
1 {  
2   "error": "subway CPYQKS79-ULD9-7QAK-ZFBWF35E has not exist.",  
3   "code": 404  
4 }
```

4、405

请求类型有误。

5、406

请求的body体格式有误，无法解析。

```
1 {  
2   "error": "invalid DELETE request: JSON decode failed.",  
3   "code": 406  
4 }
```

6、500

服务器内部错误。

修改地铁线路名称

URL

```
1 /line/v1/modify/name
```

请求类型 PUT

请求body示例：

```
1 {
2   "uuid": "2W1R01UI-5JUR-BWM3-PGT2D3XS",
3   "name": "line-3"
4 }
```

| 属性名称 | 属性类型 | 属性说明 |
|------|--------|----------|
| uuid | string | 地铁线路uuid |
| name | string | 地铁线路的新名称 |

返回值

1、200

修改地铁线路名称成功。

```
1 {
2   "success": "update subway line JLBXYAH7-OT25-7MLH-G66N3GHD success.",
3   "code": 200
4 }
```

2、400

请求参数有误。

```
1 {
2   "error": "BadRequest: Invalid param.",
3   "code": 400
4 }
```

3、401

用户没有登录或不是admin用户。

```
1 {
2   "error": "limited authority",
3   "code": 401
4 }
```

4、404

所要修改地铁线路的uuid不存在。

```
1 {}
2   "error": "subway JLBXYAH7-OT25-7MLH-G66N3GHDS has not exist.",
```

```
3   "code": 404
4 }
```

4、405

请求类型有误。

5、406

请求的body体格式有误，无法解析。

```
1 {
2   "error": "invalid PUT request: JSON decode failed.",
3   "code": 406
4 }
```

6、500

服务器内部错误。

获取地铁线路列表

URL:

```
1 /line/v1/list
```

请求类型: GET

请求键值对: 无

返回值:

1、200

用户获取地铁线路列表成功。

```
1 {
2   "line": [
3     {
4       "uuid": "JLBYXAH7-OT25-7MLH-G66N3GHD",
5       "name": "line-2"
6     }
7   ],
8   "code": 200
9 }
```

2、500

服务器拒绝访问请求，服务器发生内部错误。