API Document

[Changelog](#_7ehfljdgote)

[Introduction](#_8s3epa1tbxfp)

[Common Request](#_67jc52505puk)

[Common parameters](#_6n5nxeaqwfca)

[Common Reponse](#_yczx2t9yl4x7)

[Tools](#_puh9pjwm0oso)

[API entry points](#_qzs854yarjro)

[API Keys](#_lkmhab6b5j4d)

[API return codes](#_mjoti542jbpe)

[API class - Authentication](#_yjwnj0vq6exm)

[/user/login/](#_9hq6qagxqchr)

[/user/sign\_up/](#_wg6hog4rnfa4)

[API class - User](#_lt5bh0l2e3kp)

[/user/update\_photo/](#_ryjnzcdmpvx7)

[/user/profile/](#_4cngpkke36o)

[API class - Picture](#_o7d3aj7yzmd6)

[/picture/upload/](#_qcokeaktzzem)

# **Changelog**

|  |  |  |
| --- | --- | --- |
| Date | Revisor | Description |
| 2016/08/25 | Leo | 新增  1.[/user/login/](#_9hq6qagxqchr) 使用者登入  2.[/user/sign\_up/](#_wg6hog4rnfa4) 使用者註冊  3.[/picture/upload/](#_qcokeaktzzem) 使用者更新頭像 |

# **Introduction**

This document is to explain the API. We will introduce our deisgn and further go through each APIs.

Our design is to follow RESTful API which is very common these years. Client needs to send HTTP request to our API entry points and get returned JSON data.

## Common Request

API supports different HTTP verbs which depends on each API. We will briefly describe supported HTTP verbs by each API.

For Content-Type which will be used in POST like reuqest, we support two kinds:

1. Traditional form POST:

Content-Type: application/x-www-form-urlencoded

sample

POST /api/v1/test HTTP/1.1

Content-Length: 37

Host: localhost:31337

Content-Type: application/x-www-form-urlencoded; charset=utf-8

apikey=aaabbb&param1=val1&param2=val2

Content-Type: multipart/form-data

(For file upload)

Please do urlencode for each parameters accordingly.

1. JSON:

Content-Type: application/json

sample

POST /api/v1/test HTTP/1.1

Content-Length: 56

Accept: application/json

Host: localhost:31337

Content-Type: application/json; charset=utf-8

{"apikey": "aaabbb", "param1": "val1", "param2": "val2"}

And our suggestion is to use JSON.

### Common parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Required/Optional | Description |
| apikey | Required | Specify a API key to differentiate different client app.  Please contact Server to request a API key. |
| fields | Optional | For mobile app, specifing a fields to filter unused response data could reduce mobile traffic. With this, mobile app can benefit especially in 3G network.  fields value use ‘,’ to sperate multiple fields and use ‘.’ dot notation to filter dictionary.  For example, if the reponse JSON is:  {  "code": 0,  "data": {  "key1": "1",  "key2": {  "kk1": 1,  "kk2": 2  }  },  "msg": "Success"  }  Specifing fields as ‘key1,key2’ will be the same.  Specifing fields as ‘key1’ will only return key1’s value.  Specifing fields as ‘key2.kk1’ will be:  {  "code": 0,  "data": {  "key2": {  "kk1": 1  }  },  "msg": "Success"  } |

## 

## Common Reponse

The API response will always have

Content-Type: application/json; charset=UTF-8

The Common response will like that:

HTTP/1.1 200 OK

Content-Length: 52

Content-Type: application/json; charset=UTF-8

Date: Fri, 26 Apr 2013 10:07:16 GMT

{

"code": 0,

"data": {

....

}

},

"msg": "Success"

}

For HTTP status code, please check this table:

|  |  |
| --- | --- |
| HTTP status code | Note |
| 200 | API sucessful |
| 401 | Try to access private data without right permissions.  E.g. without specify session\_token |
| 500 | API server internal errors |

There is also a ‘code’ and ‘msg’ returned from API.

**code is for your code, msg is for human.**

**The successful code is code=0, msg=’Success’**

Otherwise, it should be error.

We will later to list all error codes.

There is a ‘data’ dictionary reutrned from API which attaching the returned value. Its value depends on each API.

## Tools

We will use [httpie](https://github.com/jkbr/httpie) to show sample HTTP request/response for each API.

# **API entry points**

|  |  |
| --- | --- |
| Environment | Entry Point |
| Development |  |
| Staging |  |
| Production |  |

# **API Keys**

|  |  |
| --- | --- |
| Name | Value |
| Android | fjijfiowe84@#)@#$\*jiewfj==-3 |

# **API return codes**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | **Code** | **Description** | **Referenced UI wording** |
| SUCCESS | 0 | Success |  |
| INVALID\_PARAMETERS | 1 | Invalid parameters |  |
| MISSING\_REQUIRED\_PARAMETERS | 2 | Missing required parameters |  |
| INVALID\_APIKEY | 3 | Invalid API key |  |
| INTERNAL\_ERRORS | 4 | Internal errors | 內部發生錯誤 |
| GENERIC\_ERROR | 5 | Generic error | 發生錯誤 |
| ERROR\_NOT\_FOUND | 6 | Resource not found | 找不到這個物件 |
| ACCESS\_DENIED | 7 | Access denied | 使用者權限不足 |
| AUTH.INVALID\_AUTH\_INFO | 1001 | Invalid account or password | 帳號或密碼錯誤 |

Error (\*) means the error message may be dynamically changed.

# **API class - Authentication**

## /user/login/

Support HTTP methods: POST [**TOP**](#h24r8kvr8632)

Information : 登入

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Required/Optional | Note |
| email | Required | email |
| password | Required | user password |

Special return codes

|  |  |
| --- | --- |
| Codes | Description |
| 1001 | Invalid account or password |

Reponse

|  |  |
| --- | --- |
| Name | Description |
| session\_token | The session token which can be later to do furture operations. |

Sample

{

* "apikey":"jj9j9c2-FKfjwe=/fewfjweFEJIwmelwf",
* "email":"aaa@gmail.com",
* "password":"aabb1122"

}

return:

{

* "code":0,
* "session\_token":"eL/BmZsuuNW4h4ST2oazTGO+c44="

}

## /user/sign\_up/

Support HTTP methods: POST [**TOP**](#h24r8kvr8632)

Information : 註冊

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Required/Optional | Note |
| email | Required | email |
| password | Required | 密碼 |
| name | Required | 名稱 |

Special return codes

|  |  |
| --- | --- |
| Codes | Description |
|  |  |

Reponse

|  |  |
| --- | --- |
| Name | Description |
| session\_token | The session token which can be later to do furture operations. |

Sample

{

* "apikey":"jj9j9c2-FKfjwe=/fewfjweFEJIwmelwf",
* "email":"aaa@gmail.com",
* "password":"aabb1122",
* "name":"Leo"

}

return:

{

* "code":0

}

# **API class - User**

## /user/update\_photo/

Support HTTP methods: POST [**TOP**](#h24r8kvr8632)

Information : 更新頭像

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Required/Optional | Note |
| photo | Required | 檔案名稱 |

Reponse

|  |  |
| --- | --- |
| Name | Description |
|  |  |

Sample

{

* "apikey":"jj9j9c2-FKfjwe=/fewfjweFEJIwmelwf",
* "photo":"12\_124141312312.jpg",
* "session\_token":"eL/BmZsuuNW4h4ST2oazTGO+c44="

}

return:

{

* "code":0

}

## /user/profile/

Support HTTP methods: POST [**TOP**](#h24r8kvr8632)

Information : 取得使用者資料

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Required/Optional | Note |
|  |  |  |

Reponse

|  |  |
| --- | --- |
| Name | Description |
| name | 使用者名稱 |
| email | email |
| photo | 頭像網址 |

Sample

{

* "apikey":"jj9j9c2-FKfjwe=/fewfjweFEJIwmelwf",
* "session\_token":"eL/BmZsuuNW4h4ST2oazTGO+c44="

}

return:

{

* "code":0,
* "name":"Leo",
* "email":"aaa@gmail.com",
* "photo":"12\_242348923.jpg"

}

# **API class - Picture**

## /picture/upload/

Support HTTP methods: POST [**TOP**](https://docs.google.com/document/d/1A_Qefhm0BueZrguN8xRXAv8CKdujgKnIUqtZNk4wURs/edit#bookmark=id.h24r8kvr8632)

Information :更新使用者頭像

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Required/Optional | Note |
| session\_token | Required | session\_token |
| picture | Optional | 圖片 (ByteArrayInputStream) |

Special return codes

|  |  |
| --- | --- |
| Codes | Description |
|  |  |

Reponse

|  |  |
| --- | --- |
| Name | Description |
| picture\_name | 圖片暫存名稱 |

Sample

{

* "apikey":"jj9j9c2-FKfjwe=/fewfjweFEJIwmelwf",
* "session\_token":"eL/BmZsuuNW4h4ST2oazTGO+c44=",
* "picture":ByteArrayInputStream

}

return:

{

* "code":0
* "picture\_name":"picturename043.jpg"

}