User Service API

Server Information:

* Host: 52.187.62.107
* Port: 10300
* Base URI: /authen

Data type:

* Integer
* String
* DateTime
* Predefined permission

Predefined Permission:

User Service API provides an action-level permission checking. An Object-level permission checking is provided only an action concern with user account.

|  |  |
| --- | --- |
| Code name | Name |
| view\_user | Can view user |
| change\_user | Can change user |
| delete\_user | Can delete user |
| add\_menu | Can add menu |
| view\_menu | Can view menu |
| change\_menu | Can change menu |
| delete\_menu | Can delete menu |
| add\_order | Can add order |
| view\_ order | Can view order |
| change\_ order | Can change order |
| delete\_ order | Can delete order |

User Permission:

* Staff: Can access all permissions.
* Standard user: view\_user, change\_user, delete\_user, view\_menu, add\_order, view\_ order, and delete\_ order.

\*\* Note that standard user can use predefined permission change\_user and delete\_user

with only itself.

APIs:

* Sign up API (/sign\_up)
  + - * POST:
  + Detail: An API for sign up which requires a username, a password, and an email. This API will return a user data and a token (used for authorization) after success.
  + Request:

Header:

* Content-Type: application/json

Body:

{

"username": "enceladus",

“first\_name”: “ice\_ngai”,

“last\_name”: “jakaila”,

“address”: “more chit”,

"email": "ice@ice.com",

"password": "12345678",

"confirm\_password": "12345678"

}

* Response (application/json):
* 200 (OK):

{

"success": true,

"payload": {

"user\_data": {

"id": 1,

"username": "enceladus",

“first\_name”: “ice\_ngai”,

“last\_name”: “jakaila”,

“address”: “more chit”,

"email": "ice@ice.com",

"is\_superuser": true,

"is\_staff": true,

"is\_active": true,

"last\_login": null,

"date\_joined": "2017-02-25

10:24:30"

},

"token": "vdlfvdl43t34t34gdflbmf6545df"

}

}

* 406 (NOT ACCEPTABLE):

{

"success": false,

"error": {

"code": 406,

"message": "Username and email may already exist,

or there are incomplete required fields.

Please use another one or try to fill out

the form again.”

}

}

* Login API (/login)
  + - * POST:
* Detail: This API is used for getting a user data and its token.
* Request:

Header:

* Content-Type: application/json

Body:

{

"username": "enceladus",

"password": "12345678",

}

* Response (application/json)
* 200 (OK):

{

"success": true,

"payload": {

"user\_data": {

"id": 1,

"username": "enceladus",

“first\_name”: “ice\_ngai”,

“last\_name”: “jakaila”,

“address”: “more chit”,

"email": "ice@ice.com",

"is\_superuser": true,

"is\_staff": true,

"is\_active": true,

"last\_login": null,

"date\_joined": "2017-02-25

10:24:30"

},

"token": "vdlfvdl43t34t34gdflbmf6545df"

}

}

* 404 (NOT FOUND):

{

"success": false,

"error": {

"code": 404,

"message": "Username or password is incorrect.

Please try again.”

}

}

* Logout API (/logout)
  + - * GET:
* Detail: This API is used for logging out an unexpected session login that server did not tend to use. (The server uses token-based authentication.).
* Request:

Header:

* Content-Type: application/json
* Authorization: Token sdvdfsdvsd5fbdlkb34534
* Response (application/json):
* 200 (OK):

{

"success": true,

}

* 500 (INTERNAL SERVER ERROR):

{

"success": false,

"error": {

"code": 500,

"message": "Logout failed

}

}

* User Information API (/user\_info)
  + - * GET:
* Detail: Provide public access information of any user. Owner can access full information of itself.
* Request:

Header:

* Content-Type: application/json
* Authorization: Token sdvdfsdvsd5fbdlkb34534

Query Parameters:

* id: integer
* Response (application/json)
* 200 (OK):

{

"success": true,

"payload": {

"user\_data": {

"id": 1,

"username": "enceladus",

"email": "ice@ice.com",

"is\_active": true,

}

}

}

* 404 (NOT FOUND):

{

"success": false,

"error": {

"code": 404,

"message": "User data not found.

Please try again.”

}

}

* + - * PUT:
* Detail: An API for editing user information.
* Request:

Header:

* Content-Type: application/json
* Authorization: Token sdvdfsdvsd5fbdlkb34534

Body:

{

"id": 1,

"password": "12345678",

“email”: “ice2@ice.com”

}

* Response (application/json)
* 200 (OK):

{

"success": true,

"payload": {

"user\_data": {

"id": 1,

"username": "enceladus",

"email": "ice2@ice.com",

"is\_superuser": true,

"is\_staff": true,

"is\_active": true,

"last\_login": null,

"date\_joined": "2017-02-25

10:24:30"

}

}

}

* 406 (NOT ACCEPTABLE):

{

"success": false,

"error": {

"code": 406,

"message": "Input data is invalid. Please correct

and try again.”

}

}

* 403 (FORBIDDEN):

{

"success": false,

"error": {

"code": 403,

"message": "Permission not allowed”

}

}

* 404 (NOT FOUND):

{

"success": false,

"error": {

"code": 404,

"message": "User data not found”

}

}

* + - * DELETE:
* Detail: Delete account.
* Request:

Header:

* Content-Type: application/json
* Authorization: Token sdvdfsdvsd5fbdlkb34534

Body:

{

“id”: 1

}

* Response (application/json)
* 200 (OK):

{

"success": true,

}

* 403 (FORBIDDEN):

{

"success": false,

"error": {

"code": 403,

"message": "Permission not allowed”

}

}

* 500 (INTERNAL SERVER ERROR):

{

"success": false,

"error": {

"code": 500,

"message": "Logout failed

}

}

* Permission Checking API (/check\_ permission)
  + - * GET:
* Detail: An API for checking a predefined permission on authorized user account.
* Request:

Header:

* Content-Type: application/json
* Authorization: Token sdvdfsdvsd5fbdlkb34534

Query Parameters:

* permission: predefined permission code name
* Response (application/json)
* 200 (OK):

{

"success": true,

"payload": {

"is\_allowed ": true

}

}

* 500 (INTERNAL SERVER ERROR):

{

"success": false,

"error": {

"code": 505,

"message": "Permission checking failed.”

}

}