# API MANUAL

01.00.00



By Mohsin Hassan on Jun 2021

# **Overview**

This section contains an overview of the data provided and the API purpose.

We built the API to be as self-documenting as possible, This article includes a complete overview of each Iterable API type, how to use them, sample requests, When submitting an API call, you can enter your API key in the headers, Administrator can provide you that API key and secret key.

The API is used to access System and perform some tasks like registration, login, change password, etc from the third party or outside the system.

For this purpose, there are some API calls and their responses we are discussing in this document.



## **Overview**

This API is created on Zend framework3 API module. To access this API we need an API user and its secret key, by which we can get a token and with that token we can access our API calls. For both calls we have to call the “getToken” API and take a token and send in the header as a bearer token.

To run this API We have to set up a Zend Framework 3 project and add that API folder inside to “Modules” folder of the project on the root.

For more details please follow the link as below:

<https://olegkrivtsov.github.io/zf3-api-reference/html/>

**Supported Documents:**

I have added Json of the calls so can get ideas about calling API endpoints. That file is included into the repository named “apicalls.json”

Also there are a file with database schema named “role\_permission\_schema.html”

You can also find in a repository file named “role\_permission.sql” for database structure.

Also if there is anything need to describe please feel free to ask.

## **Goals**

1. Keep updating all third parties that are using API.
2. This document will help developers to maintain or extend API.
3. Calls can be easily created and executed with the help of this document with no time.
4. All calls Inputs are written with key and values as well as data types of submission data.
5. Success and Error responses are written with every call so developers can set their inputs and outputs.

## **Get Token Call:**

This call is used to get access tokens. This token is a randomly generated string and valid for some time period.

This token will be used in the next API Calls for check authentication of the valid or invalid user. The responses to success or failure are written below.

Credentials example to use this call are as below for testing, live credentials will provide later:

email: youremail@yourdomain.com

password: 123456

URL is as follows:

[YOURDOMAIN/apitoken/getToken](http://52.62.3.10/sso/public/index.php/api/apitoken/getToken)

Method:

POST

Parameters JSON:

"mode": "formdata",

"data": [

{

"key": "email",

"value": "YOUR\_ADMIN\_EMAIL",

"description": "",

"type": "text"

},

{

"key": "password",

"value": "YOUR\_PASSWORD",

"description": "",

"type": "text"

},

],

This call will response as below:

Success Response:

{

"access\_token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90aW1lIjoiMjAxOS0wMS0xNyAwOTo0MjoxMSIsImVtYWlsIjoiYXBpQHFvY3Rvci5jb20uYXUiLCJuYW1lIjoiUW9jdG9yIEFwaSIsInVzZXJfaWQiOiI2IiwiaXBfYWRkcmVzcyI6IjEyMS4xMjEuODkuMTI4In0.X389yHFW7tD-qACvM1KAz82oDQl4VzfLQhFnVsQ9eGQ",

"refresh\_token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJudW1iZXIiOjEwNzQ0OTEzNzgsInVzZXJfaWQiOiI2In0.3\_MtEKNJgmgkfCg\_DuPtCmkU-fv0qc5JTwnOz98GqSk",

"action":"OK","message":"Success","expires\_in":1800

}

Error Response Example:

{

"Access\_token":"0",

"Refresh\_token":"0",

"Action":"0",

"message":"Incorrect API user or password"

}

Required Fields:

Email

password

## **Check User Permission:**

This call will return true or false as per data provided.

If User is able to access as per featureName it will return true other wise return false.

URL is as follows:

[YOURDOMAIN](http://52.62.3.10/sso/public/index.php/api/apitoken/getToken)/api/sso/getUserValidation?email=youremail@gmail.com&featureName=practitioner.manage

Method:

GET

Success Response:

{

"status": "OK",

"result": {

"canAccess": "1"

}

}

Error Response Example:

{

"status": "NOK",

"result": {

"canAccess": "0"

}

}

Required Fields:

Email

featureName

For more details please refer to “Supported Document”.

## **Grant User Permission:**

This API call will assign permission to the user.

URL is as follows:

[YOURDOMAIN](http://52.62.3.10/sso/public/index.php/api/apitoken/getToken)/api/sso/getUserValidation?email=youremail@gmail.com&featureName=practitioner.manage

Method:

POST

Success Response:

{"status":"OK","result":{"success":1}}

Error Response Example:

{"status":"NOK","result":{"success":0}}

For more details please refer to “Supported Document”.

Required Fields:

Email

featureName

## horizontal line