**API REFERENCE**

Playground

Table of Contents

[1.0 Authorization 3](#_Toc480280495)

[1.1 Access Token 3](#_Toc480280496)

[1.1.1 Process flow for Password Grant 3](#_Toc480280497)

[1.2 Web Service Request 4](#_Toc480280498)

[2.0 Specifications 5](#_Toc480280499)

[2.1 Users 5](#_Toc480280500)

[2.2 Register 5](#_Toc480280501)

[2.3 Password 6](#_Toc480280502)

[2.4 Password Reset 6](#_Toc480280503)

# Authorization

The web service adheres to the OAuth2 (2 legged) authorization protocol. Application will need to acquire an access token from the Service, and thereafter use the returned access token to obtain access to protected resources in the system.

## Access Token

To acquire an access token, Application will need to perform a POST request to the endpoint ‘v1/identity/token’.

An authorization header is required to be sent along with the request.

|  |
| --- |
| **Format:**  Authorization: Basic {AUTH\_STRING}  **Note:**  The authentication string should base64 encode the client\_id and client\_secret in the format: “{CLIENT\_ID}:{CLIENT\_SECRET}” |

The following describe the grants supported by the service:

### Process flow for Password Grant

1. Application makes request to Service:

* grant\_type = ‘password’
* username
* password

1. Service responds with Access Token:

* access\_token
* expires\_in
* token\_type

## Web Service Request

The request to the Web Service will adhere to the Representational State Transfer (REST) architectural style. The data format will be in JSON.

The Access Token acquired via the OAuth2 protocol will be used to submit request to the Web Service. An authorization header is required to be sent along with each request.

|  |
| --- |
| **Format:**  Authorization: Bearer {ACCESS\_TOKEN}  **Available REST Methods:**  HEAD, GET, POST, PUT, DELETE, OPTIONS |

# Specifications

## Users

Display information on active user.

|  |  |  |  |
| --- | --- | --- | --- |
| URI | Method | Description | Available |
| users | GET | List all models by page | Yes |
| users/{id} | GET | Return the details of model {id} | No |
| users | POST | Create a new model | No |
| users/{id} | PUT | Update the model {id}  Attributes  name, varchar(255)\*  mobile\_no, varchar(20)\* | Yes |
| users/{id} | DELETE | Delete the model {id} | No |

1. *Reference for identity/users*

## Register

Register user to the application. Implements web controller. Only HTTP Basic authentication is required.

|  |  |  |  |
| --- | --- | --- | --- |
| URI | Method | Description | Available |
| register | POST | Web Controller  Attributes  email, varchar(255)\*  password, varchar(50)\*  name, varchar(255)\*  mobile\_no, varchar(20)\* | Yes |

1. *Reference for register*

## Password

Handles password update for user. Implements web controller.

|  |  |  |  |
| --- | --- | --- | --- |
| URI | Method | Description | Available |
| password | POST | Web Controller  Attributes  password, varchar(20)\*  old\_password, varchar(20)\* | Yes |

1. *Reference for identity/password*

## Password Reset

Generate a password reset request. Entry point via frontend web interface.

|  |  |  |  |
| --- | --- | --- | --- |
| URI | Method | Description | Available |
| site/request-password-reset | POST | Frontend Web Controller  Attributes  email, varchar(255)\* | Yes |

1. *Reference for site/request-password-reset*