

Contents

1 Introduction:..... 2

2 Register User ..... 2

    1. Functional Specification: ..... 2

    2. Service Operations: ..... 2

3 Login User ..... 3

    1. Functional Specification: ..... 3

    2. Service Operations: ..... 3

4 Register Device ..... 4

    1. Functional Specification: ..... 4

    2. Service Operations: ..... 4

5 List All Devices ..... 4

    1. Functional Specification: ..... 4

    2. Service Operations: ..... 5

6 Authentication & Authorization ..... 5

[Type here]

## 1 Introduction:

This service aims to provide register electronic devices to system. In order to register electronic devices user must have register itself to a system first.

## 2 Register User

### 1. Functional Specification:

Service Name	Register User
Service Description	This service allow the user to registers to the system
HTTP METHOD	Post
Endpoint	http://localhost:8081/auth/register

### 2. Service Operations:

#### 1. INPUT PARAMETERS

Name	Type	Length	Optional	Description
name	String		NO	Name Of user
email	String		NO	Email address
password	String		NO	Password

#### 2. Sample input

```
{
  "name": "Terminus Group",
  "email": "test1@gmail.com",
  "password": "test123"
}
```

#### 3. OUTPUT JSON

Result will be JSON objects.

##### Operation Response

```
{
  "status": 200,
  "message": "user created successfully",
  "code": "successful",
  "data": "Terminus Group"
}
```

**Note: - Basic authentication is required to use this service (Refer to Authentication and Authorization)**

[Type here]

### 3 Login User

#### 1. Functional Specification:

Service Name	Login User
Service Description	This service allow the user to login to the system. The response will be token which will be used to call another API throughout the service.
HTTP METHOD	POST
Endpoint	http://localhost:8081/oauth/token

#### 2. Service Operations:

##### 1 INPUT PARAMETERS

Name	Type	Length	Optional	Description
grant_type _	String		NO	-
username	String		NO	Email of user
password	String		NO	Password of user

##### 2 Sample input

###### Form Data

grant\_type password  
username test1@gmail.com  
password 123456

##### 3 OUTPUT JSON

Result will be JSON objects.

###### Operation Response

```
{  
  "access_token": "ed740626-edbc-43fb-b405-dfd34db72a86",  
  "token_type": "bearer",  
  "refresh_token": "55967444-e79c-4129-981d-1d962d482427",  
  "expires_in": 147643,  
  "scope": "read write"  
}
```

**Note: - 1.Basic authentication is required to use this service (Refer to Authentication and Authorization)**  
**2. Input parameters must be a form data**

[Type here]

## 4 Register Device

### 1. Functional Specification:

Service Name	Register User
Service Description	This service allow logged in user to register the device to system
HTTP METHOD	POST
Endpoint	http://localhost:8081/api/v2.0/devices/register

### 2. Service Operations:

#### 1 INPUT PARAMETERS

Name	Type	Length	Optional	Description
imei	String		NO	Imei of device
sim	String		NO	Sim of device

#### 2 Sample input

```
{
  "imei": "12345678",
  "sim": "0565697710"
}
```

#### 3 OUTPUT JSON

Result will be JSON objects.

```
{
  "status": 200,
  "message": "Device Register successfully in database",
  "code": "successful",
  "data": "1"
}
```

**Note: - Authorization token is required to use this service (Refer to Authentication and Authorization)**

## 5 List All Devices

### 1. Functional Specification:

Service Name	Get all registered devices
Service Description	This service will list all device that are registered to the system
HTTP METHOD	POST
Endpoint	http://localhost:8081/api/v2.0/devices/listalldevices

[Type here]

[Type here]

## 2. Service Operations:

### 1. INPUT PARAMETERS

No input parameters are required for this service

### 2. OUTPUT JSON

Result will be the list of JSON objects.

```
[
  {
    "id": 1,
    "imei": "123456789",
    "sim": "0565697710",
    "enable": true,
    "configured": false
  },
  {
    "id": 2,
    "imei": "12345678",
    "sim": "0565697710",
    "enable": true,
    "configured": false
  }
]
```

**Note: - Authorization token is required to use this service (Refer to Authentication and Authorization)**

## 6 Authentication & Authorization

**Register user** and **Login user** API must include the user id and password with a colon ":" separated base64 encoded and send this information in the request header as shown below.

For example: user id is john and password is abc123 then encode the string "john:abc123" to base64

Requests include the header name "Authorization" with value starts with " Basic " and concatenate with encoded string in the header as shown below.

```
Authorization: Basic bG9sOnNlY3VyZQ==
```

Other than login and register all API must include authorization token shown below in request header.

```
Authorization: Bearer ed740626-edbc-43fb-b405-dfd34db72a86
```

[Type here]

[Type here]

[Type here]