**Reportal**

Web Service Structure

First Draft For Api Document

Last updated on - 11/14/17

Objective

The document describes the required API’s for Reportal.

**Requests**

This API service receives HTTP requests. Template of the request:

<METHOD> <Resource – Path> HTTP/1.1

HOST: <Api-host>

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

Content-Length: <Param- String-Length>

User-Agent: <Api-Client-Name>

**Method** GET, POST method.

**Resource path** is the string with URI and query

**API host** is the domain name or IP address of the service server.

**Secret:**

**Base URL :** http://ec2-52-37-224-72.us-west-2.compute.amazonaws.com/

**Responses**

|  |  |
| --- | --- |
| **Label** | **Explanation** |
| Success | The request was fulfilled. |
| Error | The server has not found anything matching the URI given. |

|  |  |  |
| --- | --- | --- |
| **Function – login, Method: Get** | | |
| **Description** | **Takes** | **Returns** |
| This function is used verify login.  User\_type is to identify is the user is maintenance person or regular user or admin.  User\_type: 1 – Regular user  User\_type: 2 – Maintenance person  User\_type: 3 – Admin  User\_category is to identify the department maintenance person is assigned to.  Category: 1 – Infrastructure  Category: 2 – Services  Category: 3 –  Electronic  Category: 0 – Other | user\_email,  user\_password (encrypted) | Return success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage : Login successfully | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: Credentials not correct. | | |
| **Response Example(Success)** | | |
| {  "user": [  {  "user\_id": "",  "user\_fname": "",  "user\_lname": "",  "user\_type": "",  "user\_category":""  }  ],  "categories": [  {  "category\_id": "",  "category\_name": ""  },  {  "category\_id": "",  "category\_name": ""  }  ],  "priorities": [  {  "priority\_id": "",  "priority\_name": ""  },  {  "priority\_id": "",  "priority\_name": ""  }  ]  } | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
| **Note**: All parameters are mandatory | | |

|  |  |  |
| --- | --- | --- |
| **function – register, Method: Post** | | |
| **Description** | **Takes** | **Returns** |
| This function is used to register user | user\_email,  user\_password,  user\_fname,  user\_lname,  user\_category (optional, only on web portal to register account for maintenance person)  user\_type | Return success message if user is registered successfully otherwise return error message. |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: User registered successfully. | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: User not registered. | | |
| Response Example | | |
|  | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
| **Note**: | | |

|  |  |  |
| --- | --- | --- |
| **function – post\_issue** | Post | |
| **Description** | **Takes** | **Returns** |
| This function is used to update user profile. | userId,  issue\_picpath,  issue\_picture\_name(Not in DB),  lat,  lon,  issue\_time,  issue\_description,  issue\_category\_id,  issue\_priority\_id | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: Issue posted successfully. | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: Issue post unsuccessful. | | |
| Response Example | | |
|  | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
| **Note**: | | |

|  |  |  |
| --- | --- | --- |
| **function – getIssueList** | For status, we can set some pre-defined status and with corresponding details:  status\_id, status\_name, status\_updatetime, status\_updatedby  Status: pending, assigned to, completed  Status id: 1, 2, 3  Status update time: whenever the status for issue is updated  Status updated by: user\_id of user who updated the status of issue  Method: Get | |
| **Description** | **Takes** | **Returns** |
| This function is used to add new location | userId(optional),  page\_no | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: Issue list send successfully. | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: Issue list send unsuccessful. | | |
| Response Example | | |
| {  "Issues": [  {  "issue\_id": "",  "issue\_picpath": "",  "issue\_location": {  "lat": "",  "lon": ""  },  "issue\_status": "",  "issue\_assignedTo": {  "user\_id": "",  "user\_fname": "",  "user\_lname": ""  },  "issue\_time": "",  "issue\_postedBy": {  "user\_id": "",  "user\_fname": "",  "user\_lname": ""  },  "issue\_description": "",  "issue\_category": "",  "issue\_priority": ""  },  {  "issue\_id": "",  "issue\_picpath": "",  "issue\_location": {  "lat": "",  "lon": ""  },  "issue\_status": "",  "issue\_assignedTo": {  "user\_id": "",  "user\_fname": "",  "user\_lname": ""  },  "issue\_time": "",  "issue\_postedBy": {  "user\_id": "",  "user\_fname": "",  "user\_lname": ""  },  "issue\_description": "",  "issue\_category": "",  "issue\_priority": ""  }  ]  } | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |

|  |  |  |
| --- | --- | --- |
| **function – forgot\_password** | Get | |
| **Description** | **Takes** | **Returns** |
| This function is used to show all user list | emailId | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: Instruction to reset password sent successfully to email address. | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: Email send failed. | | |
| Response Example | | |
|  | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
| **Note**: | | |

|  |  |  |
| --- | --- | --- |
| **function – reset\_password** | Put | |
| **Description** | **Takes** | **Returns** |
| This function is used to show user friend list. | Token,  password(encrypted) | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: Password reset successful. | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: Password reset unsucessful. | | |
| Response Example | | |
|  | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
| **Note**: | | |

|  |  |  |
| --- | --- | --- |
| **function – update\_user\_settings** | Method: Put | |
| **Description** | **Takes** | **Returns** |
| This function is used to search added location. | user\_id,  user\_email (optional),  user\_password (optional),  user\_fname (optional),  user\_lname (optional),  user\_category (optional, only on web portal to register account for maintenance person)  user\_status (optional, only on web portal to manage the status of user as active or deactivate) | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: User details updated successfully. | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: User details update unsuccessful. | | |
| Response Example | | |
|  | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
| **Note**: | | |

|  |  |  |
| --- | --- | --- |
| **function – get\_user\_list** | Method: Get | |
| **Description** | **Takes** | **Returns** |
| This function is used to set profile pic | user\_type (optional),  page\_number | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: User list sent successfully. | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: User list sent unsuccessful. | | |
| Response Example | | |
| {  "user": [  {  "user\_id": "",  "user\_fname": "",  "user\_lname": "",  "user\_type": "",  "user\_category":"",  "user\_email":"",  “user\_status”:””  },  {  "user\_id": "",  "user\_fname": "",  "user\_lname": "",  "user\_type": "",  "user\_category":"",  "user\_email":"".  “user\_status”:””  }  ]  } | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
| **Note**: 20 results will be send per page. | | |

|  |  |  |
| --- | --- | --- |
| **function – get\_user** | Method: get | |
| **Description** | **Takes** | **Returns** |
| This function is used to show following list. | user\_id | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: User details sent successfully. | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: User details send unsuccessful. | | |
| Response Example | | |
| {  "user\_id": "",  "user\_fname": "",  "user\_lname": "",  "user\_type": "",  "user\_category":"",  "user\_email":"",  “user\_status”:””  } | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **function – getIssue** | Method: Get | |
| **Description** | **Takes** | **Returns** |
| This function is used to show following list. | issue\_id | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: Issue details sent successfully(Get) | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: Issue details send unsuccessful(Get) | | |
| Response Example | | |
| {  "issue\_id": "",  "issue\_picpath": "",  "issue\_location": {  "lat": "",  "lon": ""  },  "issue\_status": "",  "issue\_assignedTo": {  "user\_id": "",  "user\_fname": "",  "user\_lname": ""  },  "issue\_time": "",  "issue\_postedBy": {  "user\_id": "",  "user\_fname": "",  "user\_lname": ""  },  "issue\_description": "",  "issue\_category": "",  "issue\_priority": ""  } | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
|  | | |

issue\_status will have three values:

0 – Open

1 – In progress

2 – Closed

|  |  |  |
| --- | --- | --- |
| **function – updtateIssue** | Method: Put | |
| **Description** | **Takes** | **Returns** |
| This function is used to show following list. | issue\_id,  issue\_status,  issue\_assigned\_to,  issue\_update\_time,  issue\_updated\_by | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: Issue updated successfully(Put). | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: Issue update failed(Put). | | |
| Response Example | | |
|  | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
|  | | |

Status for each user, category, priority:

0 – Deactivate

1 – Activate

Default Status will be Activate for Category and Priority.

|  |  |  |
| --- | --- | --- |
| **function – category** | Method: Post, Put | |
| **Description** | **Takes** | **Returns** |
| This function is used to show following list. | category\_id (optional, only required in put),  category\_name,  category\_status | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: Category added/Update successfully. | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: Category add/Update unsuccessful. | | |
| Response Example | | |
|  | | |
| **Cipher:** md5 (secret. “sequence in which you pass parameters”) | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **function – priority** | Method: Post, Put | |
| **Description** | **Takes** | **Returns** |
| This function is used to show following list. | priority\_id (optional, only required in put),  priority\_name,  priority\_status | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: Priority added/Update successfully. | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: Priority add/Update unsuccessful. | | |
| Response Example | | |
|  | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **function – getNotification** | Method: Get | |
| **Description** | **Takes** | **Returns** |
| This function is used to show following list. | user\_id | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: Notification list sent successfully. | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: Notification list send unsuccessful. | | |
| Response Example | | |
| {  "notification": [  {  "notification\_id": "",  "notification\_title": "",  "notification\_description": ""  },  {  "notification\_id": "",  "notification\_title": "",  "notification\_description": ""  }  ]  } | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **function – deleteNotification** | Method: Delete | |
| **Description** | **Takes** | **Returns** |
| This function is used to show following list. | notification\_id (it will be an array of notification ids, in case if user wants to delete multiple notification at the same time) | Success and response data |
| **Success Response Example in header (Success)** | | |
| responseCode : 200  responseMessage: Notification deleted successfully. | | |
| **Error Response Example** | | |
| responseCode : 400  responseMessage: Notification delete unsuccessful. | | |
| Response Example | | |
|  | | |
| **Cipher:** md5 (secret. ”sequence in which you pass parameters”) | | |
|  | | |

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ End of Document***