# Neato SmartApp API Documentation

Raja Software

Last updated: 03/25/2014

|  |  |  |
| --- | --- | --- |
| Date | Version | Author/Comments |
| 03/24/2014 | 0.1 | First draft of the document that captures all the available APIs. |
| 03/25/2014 | 0.2 | Formatted the JSON response and added section on Web Service Test Console. |
| 03/25/2014 | 0.3 | Added more description for SetRobotProfileDetail3 and IsVirtuallyOnline |
| 03/26/2014 | 0.4 | Document review changes |

# 

# Summary

This document gives an overview of all the APIs that are used by SmartApp. This document explains the request/response for each of these APIs in the success as well as in the error condition.

# General Guidelines

Each of these APIs accepts the parameter in HTTP POST and would not work with HTTP GET. Also each API expects a valid API Key that should be passed in the api\_key parameter. If you do not send API key with each call, you would get an error response like:

{

"status": -1,

"message": "Method call failed the API Authentication",

"error": {

"code": "-174",

"message": "Method call failed the API Authentication"

}

}

If a required parameter is not passed, a response like below is returned:

{

"status": -1,

"message": "Missing parameter app\_id in method user.check\_for\_upgrades",

"error": {

"code": "-102",

"message": "Missing parameter in method call"

}

}

Some APIs require auth\_token. You can get the auth\_token from the GetUserAuthToken API. If you do not pass the auth\_token, wherever it is required, you would be getting an error message like:

{

"status": -1,

"message": "Method call failed the API Authentication",

"error": {

"code": -101,

"message": "Method call failed the API Authentication."

}

}

API call response returns a JSON response. Each API response has a status key. status = 0 indicates that the API call succeeded and status =-1 indicates that the API call failed. If an API call was successful, the result is returned in the result key of the response. A response also has a message key but that should be ignored, as in the later versions of the API, a more structured error key is returned that has code and message keys in it. For handling the errors, please use error key instead of the message.

# Web Service Test Console

You can see these web services in action by going to this URL:

<http://neatostaging.rajatogo.com/wstest/>

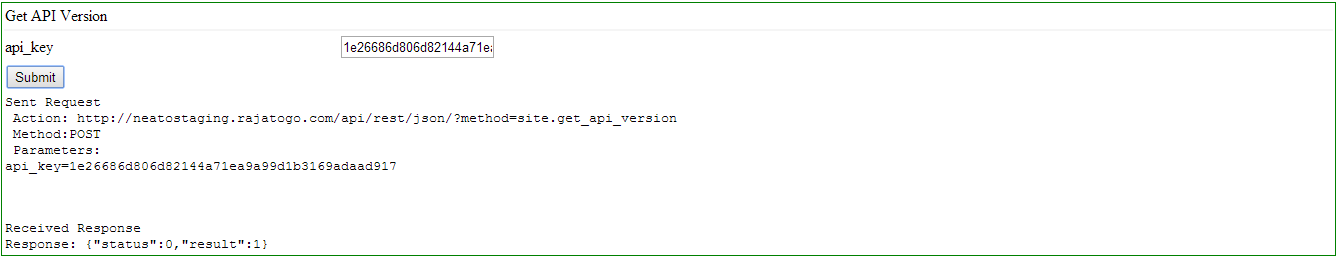
This web service test console is split into 5 web service sections and 1 utility section. Each web service section, lists out the web services related to a specific component like User, Robot etc.

Utility section is added to give out easy navigation to all the utility methods that would come handy to make web service calls. As of now it has just 1 method, that converts an image into base\_64 encoded string, which can be used to test out the map related web services.

This test console acts as a web consumer of the web services. You can pass on the parameters as mentioned in the UI and it would invoke the relevant API.

If you want to see more details about the APIs, click on the "Expand all" button at the top right corner and it would show you details about the web services. Also if you want to try out the APIs with another API key, you can specify it at the top of each page and click on "Apply API Key" button.

A sample web service call from this test console looks like:



## User APIs

This section explains the APIs related to the User creation/update and login.

#### GetAPIVersion

This is a heartbeat API to check if the backend server is alive or not.

<http://neatostaging.rajatogo.com/api/rest/json?method=site.get_api_version>

**Parameters**

api\_key, your API key

**Success Response**

{

"status": 0,

"result": "1"

}

#### CheckForUpgrades

This API is used to get latest “eligible” SmartApp versions,based on the OS, OS version and the current version number that the SmartApp is on.

<http://neatostaging.rajatogo.com/api/rest/json?method=user.check_for_upgrades>

**Parameters**api\_key :Your API Key

app\_id :Application ID

current\_appversion :Application version on device (Optional)

os\_type :Operating system on device (Optional)

os\_version :Operating system version on device (Optional)

**Success Responses**

{

"status": 0,

"result": {

"current\_app\_version": "1.0.0.1",

"latest\_version": "0.5.1.00",

"latest\_version\_url": "http://rajatogo.com/public\_shared/NeatoSmartApp.apk",

"upgrade\_status": "0"

}

}

**Failure Responses**

*If Application details not found for given ID:*

{

"status": -1,

"message": "App Id does not exist.",

"error": {

"code": "-103",

"message": "App Id does not exist."

}

}

#### CreateUser (Deprecated)

This API is used to create a user.This API supports option to create user with social networking information

<http://neatostaging.rajatogo.com/api/rest/json?method=user.create>

**Parameters**

api\_key, your API key

name, Name of the user

email, Email of the user

password, Password of the user. It does not need to be unique.

account\_type, Native OR Facebook (OR Google etc)

external\_social\_id, External Social ID (e.g. Facebook ID (numeric value) that is returned by the Facebook). This is required ONLY when the account type is NOT Native.

**Success Response**

{

"status": 0,

"result": {

"success": true,

"guid": 1074,

"user\_handle": "d8828e4ef9596dd0be3b8c4cf0de9502",

"validation\_status": 0

}

}

*If email exists but the social information does not exist*

{

"status": 0,

"result": {

"success": true,

"guid": 55,

"message": "Merged user",

"user\_handle": "ce475c5c9b84938f368efe99100b2a11",

"validation\_status": 0

}

}

Validationstatuses that you will get in response:

* validation\_status = 0 , means that the account has been validated.
* validation\_status = -1, means that the user has not been validated, but user is still within the grace period.
* validation\_status = -2 indicates that the email address has not been validated and user has exceeded the grace period too.

**Failure Responses**

*If unsupported account type is passed:*

{

"status": -1,

"message": "Method call failed the API Authentication",

"error": {

"code": -103,

"message": "Account Type is NOT supported."

}

}

*If email is not valid*

{

"status": -1,

"message": "The email address you provided does not appear to be a valid email address.",

"error": {

"code": -105,

"message": "The email address you provided does not appear to be a valid email address."

}

}

*If email already exists and account type is native*

{

"status": -1,

"message": "This email address has already been registered.",

"error": {

"code": -106,

"message": "This email address has already been registered."

}

}

*If Social information exists and the account type is Facebook*

{

"status": -1,

"message": "This social information already exists.",

"error": {

"code": -107,

"message": "This social information already exists."

}

}

*If Jabber service is not able to create chat user*

{

"status": -1,

"message": "User could not be created because jabber service in not responding.",

"error": {

"code": -108,

"message": "User could not be created because jabber service in not responding."

}

}

#### CreateUser2 (Deprecated)

This is a variation of CreateUser API. This API additionally accepts a new parameter called alternate\_email.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=user.create2>

**Parameters**

api\_key, your API key

name, Name of the user

email, Email of the user

alternate\_email, Alternate Email of the user

password, Password of the user. It does not need to be unique.

account\_type, Native OR Facebook (OR Google etc)

external\_social\_id, External Social ID (e.g. Facebook ID (numeric value) that is returned by the Facebook). This is required ONLY when the account type is NOT Native

Consideration for validation\_status which you will get in response

* validation\_status: 0 -> Validated - this means that the account has been validated.
* validation\_status: -1 -> NotValidatedButInGracePeriod - the user has not been validated, but user is still within the grace period.
* validation\_status: -2 -> NotValidated - this email address has not been validated.

**Success Responses**

{

"status": 0,

"result": {

"success": true,

"guid": 1074,

"user\_handle": "d8828e4ef9596dd0be3b8c4cf0de9502",

"validation\_status": -1

}

}

*If email exists but the social information does not exist*

{

"status": 0,

"result": {

"success": true,

"guid": 55,

"message": "Merged user",

"user\_handle": "ce475c5c9b84938f368efe99100b2a11",

"validation\_status": 0

}

}

**Failure Responses**

*If unsupported account type is passed*

{

"status": -1,

"message": "Account Type is NOT supported.",

"error": {

"code": -103,

"message": "Account Type is NOT supported."

}

}*If password is less than 6 character*

{

"status": -1,

"message": "Password length should be 6 character",

"error": {

"code": "-186",

"message": "Password length should be 6 character."

}

}

*If email is not valid*

{

"status": -1,

"message": "The email address you provided does not appear to be a valid email address.",

"error": {

"code": -105,

"message": "The email address you provided does not appear to be a valid email address."

}

}

*If alternate email is not valid*

{

"status": -1,

"message": "The alternate email address you provided does not appear to be a valid email address.",

"error": {

"code": "-115",

"message": "The alternate email address you provided does not appear to be a valid email address."

}

}

*If email already exists and account type is native*

{

"status": -1,

"message": "This email address has already been registered.",

"error": {

"code": -106,

"message": "This email address has already been registered."

}

}

*If Social information exists and the account type is Facebook*

{

"status": -1,

"message": "This social information already exists.",

"error": {

"code": "-107",

"message": "This social information already exists."

}

}

*If Jabber service does not able to create chat user*

{

"status": -1,

"message": "User could not be created because jabber service in not responding.",

"error": {

"code": -108,

"message": "User could not be created because jabber service in not responding."

}

}

*If Social information is missing and the account type is Facebook*

{

"status": -1,

"message": "Missing parameter external\_social\_id in method user.create2",

"error": {

"code": "-102",

"message": "Missing parameter in method call"

}

}

#### CreateUser3

This API is another variation of Create User API that accepts extra parameters like country code and opt in preferences.

<http://neatostaging.rajatogo.com/api/rest/json?method=user.create3>

**Parameters**

api\_key, your API key

name, Name of the user

email, Email of the user

alternate\_email, Alternate Email of the user

password, Password of the user. It does not need to be unique.

account\_type, Native OR Facebook (OR Google etc)

external\_social\_id, External Social ID (e.g. Facebook ID (numeric value) that is returned by the Facebook). This is required ONLY when the account type is NOT Native.

extra\_param, Extra Information of User should be in Json format(e.g. {"country\_code":"IN","opt\_in":"true"})

Consideration for validation\_status which you will get in response

* validation\_status: 0 -> Validated - this means that the account has been validated.
* validation\_status: -1 -> NotValidatedButInGracePeriod - the user has not been validated, but user is still within the grace period.
* validation\_status: -2 -> NotValidated - this email address has not been validated.

**Success Response**

{

"status": 0,

"result": {

"success": true,

"guid": 1074,

"user\_handle": "d8828e4ef9596dd0be3b8c4cf0de9502",

"validation\_status": -1

}

}

**Failure Responses**

*If unsupported account type is passed*

{

"status": -1,

"message": "Account Type is NOT supported.",

"error": {

"code": -103,

"message": "Account Type is NOT supported."

}

}

*If email is not valid*

{

"status": -1,

"message": "The email address you provided does not appear to be a valid email address.",

"error": {

"code": -105,

"message": "The email address you provided does not appear to be a valid email address."

}

}

*If Alternate Email is not valid*

{

"status": -1,

"message": "The alternate email address you provided does not appear to be a valid email address.",

"error": {

"code": "-115",

"message": "The alternate email address you provided does not appear to be a valid email address."

}

}

*If email already exists and account type is native*

{

"status": -1,

"message": "This email address has already been registered.",

"error": {

"code": -106,

"message": "This email address has already been registered."

}

}

*If Social information exists and the account type is Facebook*

{

"status": -1,

"message": "This social information already exists.",

"error": {

"code": "-107",

"message": "This social information already exists."

}

}

*If Jabber service does not able to create chat user*

{

"status": -1,

"message": "User could not be created because jabber service in not responding.",

"error": {

"code": -108,

"message": "User could not be created because jabber service in not responding."

}

}

*If Social information is missing and the account type is Facebook*

{

"status": -1,

"message": "Missing parameter external\_social\_id in method user.create3",

"error": {

"code": "-102",

"message": "Missing parameter in method call"

}

}

*If extra\_param is not in json format*

{

"status": -1,

"message": "The JSON Object you have provided does not appear to be a valid.",

"error": {

"code": "-126",

"message": "The JSON Object you have provided does not appear to be a valid."

}

}

#### GetUserAuthToken

This API returns the user auth token for a user. This auth token is used across the API calls, wherever a user login is required to perform certain actions. You can get the auth token by either sending Native as account type and pass on both email and password or select Facebook and pass on social external id (Facebook ID).

<http://neatostaging.rajatogo.com/api/rest/json?method=auth.get_user_auth_token>

**Parameters**

api\_key, your API key

account\_type, Account type (you can pass either Native or Facebook)

If account type is native, provide, email and password

If account type is Facebook, provide external\_social\_id

**Success Response**

*If user is within grace period and inactive*

{

"status": 0,

"result": "eb9cf0c04512903aac873697b33f45cc7e6f8328",

"extra\_params": {

"validation\_status": -1,

"message": "Please activate your account, your account still inactive"

}

}

*If user is exceed grace period and inactive*

{

"status": 0,

"result": null,

"extra\_params": {

"validation\_status": -2,

"message": "Sorry, Please validate your email first and then login again."

}

}

*If user is active*

{

"status": 0,

"result": "eb9cf0c04512903aac873697b33f45cc7e6f8328",

"extra\_params": {

"validation\_status": 0,

"message": ""

}

#### }

#### Failure Responses

*If incorrect email/password combination and account type is Native*

{

"status": -1,

"message": "Method call failed the API Authentication",

"error": {

"code": -101,

"message": "Method call failed the API Authentication."

}

}

*If incorrect external social ID and account type is Facebook*

{

"status": -1,

"message": "User could not be authenticated",

"error": {

"code": "-145",

"message": "Social id does not exist. "

}

}

*If unsupported account type is passed (e.g. Google, for the time being)*

{

"status": -1,

"message": "Account Type is not supported",

"error": {

"code": -103,

"message": "Account Type is not supported."

}

#### }

#### SetUserAccountDetails (Deprecated)

This API is used to set the user’s account and profile details. This API accepts a JSON of profile key/values.

http://neatostaging.rajatogo.com/api/rest/json?method=user.set\_account\_details

**Parameters**

api\_key, your API key

email, User's email (Optional)

auth\_token, User's auth token

profile, Map of key=>value pairs, e.g. profile{'name'=>'james bond', 'facebook\_external\_social\_id'=>'12312111', 'country\_code'=>'US', 'opt\_in' => 'true'}

**Success Response**

{

"status": 0,

"result": "1"

}

**Failure Responses**

*If add detail key is invalid*

{

"status": -1,

"message": "Invalid value for key www.",

"error": {

"code": "-109",

"message": "Invalid value for key"

}

}

*If add detail key is set and country\_code is invalid*

{

"status": -1,

"message": "invalid country\_code",

"error": {

"code": "-187",

"message": "Country code is invalid."

}

}

*If add detail key is set and opt\_in is invalid*

{

"status": -1,

"message": "Invalid opt in flag. It should be true or false.",

"error": {

"code": "-188",

"message": "opt\_in flag is invalid. It should be true or false"

}

}

#### GetCountryCodeDetails

This API returns the country code for a passed country name.

<http://neatostaging.rajatogo.com/api/rest/json?method=user.get_country_code>

**Parameters**

api\_key your API key

country\_name Enter Country Name

**Success Response**

{

"status": 0,

"result": {

"success": true,

"country\_name": "United States",

"country\_code": "US"

}

}

**Failure Response**

*If country name does not exist*

{

"status": -1,

"message": "Please Enter valid Country Name",

"error": {

"code": "-187",

"message": "Provided country name is not valid."

}

}

#### GetAttributes

This API is used to get user's attributes like device type and app version.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=user.get_attributes>

**Parameters**

api\_key your API key

auth\_token User's auth token

**Success Response**

{

"status": 0,

"result": {

"success": true,

"user\_attributes": {

"name": "mac",

"operating\_system": "",

"version": ""

}

}

}

**Failure Responses**

*If Attributes are not set:*

{

"status": -1,

"message": "Attributes not found for this user",

"error": {

"code": "-158",

"message": "No attribute found for this user"

}

}

#### ForgotPassword

This API is used to enable user who has lost his password, to log back in to the SmartApp. This API sends an email to the primary email address and this email contains a temporary password.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=user.forget_password>

**Parameters**

api\_key, your API key

email, User's email address

**Success Response**

{

"status": 0,

"result": {

"success": true,

"message": "New password is sent to your email."

}

}

**Failure Responses**

*If email address not found in database*

{

"status": -1,

"message": "Email does not exist.",

"error": {

"code": "-112",

"message": "Email does not exist."

}

}

#### ChangePassword

This API is used to change the password of the logged in user.

<http://neatostaging.rajatogo.com/api/rest/json?method=user.change_password>

**Parameters**

api\_key, your API key

auth\_token, User's auth\_token

password\_old, User's old password

password\_new, User's new password

#### Success Responses

{

"status": 0,

"result": {

"success": true,

"message": "Your password is changed successfully."

}

#### }

#### Failure Responses

*If old password does not match with user's existing password*

{

"status": -1,

"message": "Old password does not match with user password.",

"error": {

"code": "-110",

"message": "Old password does not match with user password."

}

}

*If new password is empty or has only spaces*

{

"status": -1,

"message": "Password should contain atleast one character.",

"error": {

"code": "-102",

"message": "Missing parameter in method call"

}

#### }

#### GetUserAccountDetails

API to get the User's Details  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=user.get_user_account_details>

**Parameters**

api\_key, your API key

email, User's email (Optional)

auth\_token, User's auth token

**Success Responses**

*If social information does not exist*

{

"status": 0,

"result": {

"id": 367,

"name": "john37",

"email": "john378@gmail.com",

"chat\_id": "1350922773\_user@rajatogo",

"chat\_pwd": "1350922773\_user",

"social\_networks": [],

"validation\_status": 0,

"alternate\_email": "xyz@abc.com",

"extra\_param": {

"country\_code": "IN",

"opt\_in": "true"

}

}

}

*If social information does exist*

{

"status": 0,

"result": {

"id": 357,

"name": "john3",

"email": "john3@gmail.com",

"chat\_id": "1350911036\_user@rajatogo",

"chat\_pwd": "1350911036\_user",

"social\_networks": [

{

"provider": "Facebook"

},

{

"external\_social\_id": "123456789"

}

],

"validation\_status": 0,

"alternate\_email": "xyz@abc.com",

"extra\_param": {

"country\_code": "IN",

"opt\_in": "true"

}

}

}

*If social information does not exist and robot association exists*

{

"status": 0,

"result": {

"id": 542,

"name": "john",

"email": "john@gmail.com",

"chat\_id": "1351499916\_user@rajatogo",

"chat\_pwd": "1351499916\_user",

"social\_networks": [],

"robots": [

{

"id": "68",

"name": "room cleaner1",

"serial\_number": "robo5",

"chat\_id": "1350987452\_robot@rajatogo"

},

{

"id": "69",

"name": "desk cleaner60",

"serial\_number": "robo6",

"chat\_id": "1350991375\_robot@rajatogo"

}

],

"validation\_status": 0,

"alternate\_email": "xyz@abc.com",

"extra\_param": {

"country\_code": "IN",

"opt\_in": "true"

}

}

}

*If both social information and robot association do not exist*

{

"status": 0,

"result": {

"id": 543,

"name": "john",

"email": "john1@gmail.com",

"chat\_id": "1351500158\_user@rajatogo",

"chat\_pwd": "1351500158\_user",

"social\_networks": [],

"robots": [],

"validation\_status": 0,

"alternate\_email": "xyz@abc.com",

"extra\_param": {

"country\_code": "IN",

"opt\_in": "true"

}

}

#### }

#### GetAssociatedRobots

This method returns the associated robots for a user.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=user.get_associated_robots>

**Parameters**

api\_key, your API key

email, User's email (Optional)

auth\_token, User's auth token

**Success Responses**

*If robot association exists*

{

"status": 0,

"result": [

{

"id": "68",

"name": "room cleaner1",

"serial\_number": "robo5",

"chat\_id": "1350987452\_robot@rajatogo"

},

{

"id": "69",

"name": "desk cleaner60",

"serial\_number": "robo6",

"chat\_id": "1350991375\_robot@rajatogo"

}

]

}

*If everything goes fine and robot association does not exist*

{

"status": 0,

"result": []

}

#### UpdateUserAuthTokenExpiry

This method updates the User Auth token expiry by a configurable duration. This method is used by SmartApp frequently to ensure that the user auth token does not expire.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=user.update_auth_token_expiry>

**Parameters**

api\_key, your API key

auth\_token, User's auth token

**Success Response**

{

"status": 0,

"result": {

"success": true,

"message": "You are successfully updated auth token expiry date."

}

#### }

#### LogoutUserAuthToken

This API is used to delete a specific Auth token.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=user.logout_auth_token>

**Parameters**

api\_key, your API key

email, User's email (Optional)

auth\_token, User's auth token

**Success Response**

{

"status": 0,

"result": {

"success": true,

"message": "You are successfully logged out."

}

}

#### DisassociateUserFromRobot

This API is used to disassociate User from Robot or Robots  
  
[http://neatostaging.rajatogo.com/api/rest/json?method=user.disassociate\_robot](http://neatodev.rajatogo.com/api/rest/json?method=user.disassociate_robot)

**Parameters**

api\_key, your API key

email, User's email

serial\_number, Serial Number of robot (If this field is empty, it would delete all the robot associated with this particular user)

**Success Responses**

*If everything goes fine, robot serial number provided and user robot association exist*

{

"status": 0,

"result": {

"success": true,

"message": "User Robot association removed successfully."

}

}

*If everything goes fine, robot serial number not provided and user-robot association exist*

{

"status": 0,

"result": {

"success": true,

"message": "User association with all robot removed successfully."

}

}

*If everything goes fine and robot association does not exist*

{

"status": 0,

"result": {

"success": true,

"message": "There is no association between provided user and robot"

}

}

**Failure Responses**

*If email does not exist*

{

"status": -1,

"message": "Email does not exist.",

"error": {

"code": "-112",

"message": "Email does not exist."

}

}

*If serial number does not exist*

{

"status": -1,

"message": "Serial number does not exist",

"error": {

"code": "-114",

"message": "Serial number does not exist."

}

}

#### IsUserValidated

This method is used to check whether a user's email is validated or not   
  
<http://neatostaging.rajatogo.com/api/rest/json?method=user.IsUserValidated>

**Parameters**

api\_key, your API key

email, Email of the user

**Success Responses**

*If email is active*

{

"status": 0,

"result": {

"validation\_status": 0,

"message": "The email address you have provided is Active"

}

}

*If email is inactive and within grace period*

{

"status": 0,

"result": {

"validation\_status": -1,

"message": "The email address you have provided does not appear to be a validated. Please validate it."

}

}

*If email is inactive and cross grace period*

{

"status": 0,

"result": {

"validation\_status": -2,

"message": "Sorry, You must validate your account to proceed."

}

}

**Failure Responses**

*If email is missing or invalid*

{

"status": -1,

"message": "The email address you have provided does not appear to be a valid email address.",

"error": {

"code": "-105",

"message": "The email address you provided does not appear to be a valid email address."

}

}

*If provided email does not exist in the database*

{

"status": -1,

"message": "The email address you have provided does not exist in our system.",

"error": {

"code": "-112",

"message": "Email does not exist."

}

#### }

#### ResendValidationEmail

This API is used to resend validation email to a given email address.

[http://neatostaging.rajatogo.com/api/rest/json?method=user.ResendValidationEmail](http://neatodev.rajatogo.com/api/rest/json?method=user.ResendValidationEmail)

**Parameters**

api\_key, your API key

email, Email of the user

**Success Responses**

{

"status": 0,

"result": {

"success": true,

"message": "We have resent validation email"

}

}

**Failure Responses**

*If email is missing or invalid*

{

"status": -1,

"message": "The email address you have provided does not appear to be a valid email address.",

"error": {

"code": "-105",

"message": "The email address you provided does not appear to be a valid email address."

}

}

*If provided email does not exist in database*

{

"status": -1,

"message": "The email address you have provided does not exist in our system.",

"error": {

"code": "-112",

"message": "Email does not exist."

}

}

*If resend limit exceeded*

{

"status": -1,

"message": "Sorry, You crossed resend validation email limit.",

"error": {

"code": "-116",

"message": "Sorry, You crossed resend validation email limit."

}

}

*If provided email already activated*

{

"status": -1,

"message": "The email address you have provided is already activated.",

"error": {

"code": "-117",

"message": "The email address you have provided is already activated."

}

}

#### GetErrorDetailByErrorCode

This API is used to get the error details for a given error code.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=user.get_error_code>

**Parameters**

api\_key, your API key

error\_code, Enter Error Code

**Success Response**

{

"status": 0,

"result": {

"-174": "Method call failed the API Authentication"

}

}

**Failure Responses**

*If error code does not exist*

{

"status": -1,

"message": "-189",

"error": {

"code": "-175",

"message": "Provided error code does not exist."

}

#### }

# Robot APIs

This section covers all the APIs related to Robot

#### CreateRobot

This API is to create the robot.   
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.create>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of the robot

name, Name of the robot

**Success Response**

{

"status": 0,

"result": {

"success": true,

"message": "Robot created successfully."

}

}

**Failure Responses**

*If Robot serial number is duplicate*

{

"status": -1,

"message": "This robot serial number already exists.",

"error": {

"code": "-118",

"message": "This robot serial number already exists."

}

}

*If Jabber service is not able to create chat user*

{

"status": -1,

"message": "Robot could not be created because jabber service in not responding.",

"error": {

"code": -108,

"message": "Robot could not be created because jabber service in not responding."

}

#### }

#### CreateRobot2

This API is a variation of Create Robot that accepts Robot type.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.create2>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of the robot

name, Name of the robot

robot\_type, 100 OR 101 (100 for Basic and 101 for Advanced)

**Success Response**

{

"status": 0,

"result": {

"success": true,

"message": "Robot created successfully."

}

}

**Failure Responses**

*If Robot serial number is duplicate*

{

"status": -1,

"message": "This robot serial number already exists.",

"error": {

"code": "-118",

"message": "This robot serial number already exists."

}

}

*If robot\_type is invalid*

{

"status": -1,

"message": "'Robot Type is not valid'",

"error": {

"code": "-142",

"message": "Robot Type is not valid"

}

}

*If Jabber service is not able to create chat user*

{

"status": -1,

"message": "Robot could not be created because jabber service in not responding.",

"error": {

"code": -108,

"message": "Robot could not be created because jabber service in not responding."

}

}

#### CheckIfRobotIsOnline

This API is used to check if a robot is online.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.is_online>

**Parameters**

api\_key, Your API is Key

serial\_number, Serial Number of the robot

**Success Response**

*If robot is online*

{

"status": 0,

"result": {

"online": true,

"message": "Robot 1234 is online."

}

}

*If robot is offline*

{

"status": 0,

"result": {

"online": false,

"message": "Robot 1234 is offline."

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Serial number does not exist",

"error": {

"code": "-114",

"message": "Serial number does not exist."

}

}

#### SetRobotProfileDetails3

This API is used to set robot's profile details. This API is the most critical piece as this enables the "real time" communication between Robots and SmartApp Users. This method is invoked by both Robots and SmartApp users to indicate that a Robot status change has happened. This results in an XMPP message being sent to all the users/robots. A sample XMPP message, which is an XML, looks like this:

<?xml version="1.0" encoding="UTF-8"?>

<packet>

   <header>

      <version>1</version>

      <signature>0xcafebabe</signature>

   </header>

   <payload>

      <request>

         <command>5001</command>

         <requestId>59054</requestId>

         <timeStamp>1395731229</timeStamp>

         <retryCount>0</retryCount>

         <responseNeeded>false</responseNeeded>

         <distributionMode>2</distributionMode>

         <params>

            <robotId>N1</robotId>

            <causeAgentId />

         </params>

      </request>

   </payload>

</packet>

The receiver of this message then parses this message and then acts upon it.

<http://neatostaging.rajatogo.com/api/rest/json?method=robot.set_profile_details3>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of robot

source\_serial\_number, if sent from robot, contains the robot's serial id (can be empty)

source\_smartapp\_id, if sent from the SmartApp, contains the user email (can be empty)

cause\_agent\_id, Unique sender ID

value\_extra, an optional JSON string

notification\_flag, Notification Flag (Default value is true)

profile, Map of key=>value pairs, e.g. profile{'name'=>'room cleaner'}

**Success Response**

{

"status": 0,

"result": 1,

"extra\_params": {

"expected\_time": 1,

"timestamp": 1369377928

}

}

**Failure Responses**

*If serial\_number is not provided*

{

"status": -1,

"message": "Missing parameter serial\_number in method robot.set\_profile\_details3",

"error": {

"code": "-102",

"message": "Missing parameter in method call"

}

}

*If source\_serial\_number or source\_smartapp\_id is missing*

{

"status": -1,

"message": "Please provide atleast one source(source\_serial\_number or source\_smartapp\_id)",

"error": {

"code": "-148",

"message": "Missing parameter source serial number or source smartapp id in method call"

}

}

*If source\_smartapp\_id is invalid*

{

"status": -1,

"message": "Please enter valid email address in field source\_smartapp\_id.",

"error": {

"code": "-149",

"message": "Please enter valid email id in the field source smartapp id"

}

}If source\_smartapp\_id does not exist

{

"status": -1,

"message": "Sorry, Provided source\_smartapp\_id(email) does not exist in our system.",

"error": {

"code": "-150",

"message": "Sorry, Provided source\_smartapp\_id(email) does not exist in our system"

}

}

*If source\_smartapp\_idis not associated with given robot:*

{

"status": -1,

"message": "Sorry, Provided source\_smartapp\_id(email) is not associated with given robot",

"error": {

"code": "-151",

"message": "Sorry, Provided source\_smartapp\_id(email) is not associated with given robot"

}

}

*If profile key is not added*

{

"status": -1,

"message": "Missing parameter profile in method robot.set\_profile\_details3",

"error": {

"code": "-102",

"message": "Missing parameter in method call"

}

#### }

#### GetRobotProfileDetails (Deprecated)

This API returns the robot's profile details.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.get_profile_details>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of robot

key, Key (Optional)

**Success Response**

{

"status": 0,

"result": {

"success": true,

"profile\_details": {

"name": "robot 1",

"serial\_number": "1",

"dark": "knight"

}

}

}

**Failure Responses**

*If serial\_number is invalid*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": -114,

"message": "Robot serial number does not exist"

}

}

*If key is invalid*

{

"status": -1,

"message": "Sorry, entered key is invalid",

"error": {

"code": "-152",

"message": "Sorry, entered key does not match with serial number."

}

}

#### GetRobotProfileDetails2

This API is variation of the GetRobotProfileDetails. It accepts a key and returns the value for this key

<http://neatostaging.rajatogo.com/api/rest/json?method=robot.get_profile_details>

Parameters

api\_key, your API key

serial\_number, Serial Number of robot

key, Key

**Success Response**

*If enter serial number is 1*

{

"status": 0,

"result": {

"success": true,

"profile\_details": {

"name": {

"value": "robot 1",

"timestamp": 0

},

"serial\_number": {

"value": "1",

"timestamp": 0

},

"dark": {

"value": "knight",

"timestamp": "1368090063"

}

}

}

}

**Failure Responses**

*If serial\_number is invalid*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": -114,

"message": "Robot serial number does not exist"

}

}*If key is invalid*

{

"status": -1,

"message": "Sorry, entered key is invalid",

"error": {

"code": "-152",

"message": "Sorry, entered key does not match with serial number."

}

}

*If source\_serial\_number or source\_smartapp\_id is missing*

{

"status": -1,

"message": "Please provide atleast one source(source\_serial\_number or source\_smartapp\_id)",

"error": {

"code": "-148",

"message": "Missing parameter source serial number or source smartapp id in method call"

}

}*If source smartapp id does not exist*

{

"status": -1,

"message": "Sorry, Provided source\_smartapp\_id(email) does not exist in our system.",

"error": {

"code": "-150",

"message": "Sorry, Provided source\_smartapp\_id(email) does not exist in our system"

}

}

*If source smartapp id is incorrect*

{

"status": -1,

"message": "Please enter valid email address in field source\_smartapp\_id.",

"error": {

"code": "-149",

"message": "Please enter valid email id in the field source smartapp id"

}

}

*If source smartapp id is not associated with any robot*

{

"status": -1,

"message": "Sorry, Provided source\_smartapp\_id(email) is not associated with given robot",

"error": {

"code": "-151",

"message": "Sorry, Provided source\_smartapp\_id(email) is not associated with given robot"

}

#### }

#### DeleteRobotProfileKey2

This method deletes a robot's profile key  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.delete_robot_profile_key2>

Parameters

api\_key, your API key

serial\_number, Serial Number of robot

key, Key

cause\_agent\_id, Cause Agent Id

source\_serial\_number, Source Serial Number

source\_smartapp\_id, Source Smartapp Id

notification\_flag, Notification Flag (Default value is true). This flag decides whether to notify the other users regarding the profile being updated. If set to true, a data changed notification will be sent to the associated users (like setRobotProfileDetails3)

**Success Response**

{

"status": 0,

"result": {

"success": true

}

}

**Failure Responses**

*If serial\_number is invalid*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": -114,

"message": "Robot serial number does not exist"

}

}

*If key is invalid*

{

"status": -1,

"message": "Sorry, entered key is invalid",

"error": {

"code": "-152",

"message": "Sorry, entered key does not match with serial number."

}

}

*If source smartapp id is incorrect*

{

"status": -1,

"message": "Please enter valid email address in field source\_smartapp\_id.",

"error": {

"code": "-149",

"message": "Please enter valid email id in the field source smartapp id"

}

}

*If source smartapp id does not exist*

{

"status": -1,

"message": "Sorry, Provided source\_smartapp\_id(email) does not exist in our system.",

"error": {

"code": "-150",

"message": "Sorry, Provided source\_smartapp\_id(email) does not exist in our system"

}

}

*If Provided source\_smartapp\_idis not associated with given robot*

{

"status": -1,

"message": "Sorry, Provided source\_smartapp\_id(email) is not associated with given robot",

"error": {

"code": "-151",

"message": "Sorry, Provided source\_smartapp\_id(email) is not associated with given robot"

}

#### }

#### GetRobotDetails

This API returns the details of Robot.

<http://neatostaging.rajatogo.com/api/rest/json?method=robot.get_details>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of robot

**Success Responses**

{

"status": 0,

"result": {

"id": "65",

"name": "desk cleaner59",

"serial\_number": "robo1",

"chat\_id": "1350924155\_robot@rajatogo",

"chat\_pwd": "1350924155\_robot"

}

}

*If user association already exists*

{

"status": 0,

"result": {

"id": "68",

"name": "room cleaner1",

"serial\_number": "robo5",

"chat\_id": "1350987452\_robot@rajatogo",

"chat\_pwd": "1350987452\_robot",

"users": [

{

"id": "542",

"name": "pradip",

"email": "pradip@gmail.com",

"chat\_id": "1351499916\_user@rajatogo"

},

{

"id": "543",

"name": "pradip",

"email": "pradip1@gmail.com",

"chat\_id": "1351500158\_user@rajatogo"

}

]

}

}

*If user association does not exist*

{

"status": 0,

"result": {

"id": "70",

"name": "room cleaner",

"serial\_number": "robo1",

"chat\_id": "1351501366\_robot@rajatogo",

"chat\_pwd": "1351501366\_robot",

"users": []

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Serial number does not exist",

"error": {

"code": -114,

"message": "Serial number does not exist"

}

}

#### GetAsociatedUsers

This API is used to get the associated users for a given serial number.

<http://neatostaging.rajatogo.com/api/rest/json?method=robot.get_associated_users>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of robot

**Success Responses**

*If user association exists*

{

"status": 0,

"result": [

{

"id": "542",

"name": "pradip",

"email": "pradip@gmail.com",

"chat\_id": "1351499916\_user@rajatogo"

},

{

"id": "543",

"name": "pradip",

"email": "pradip1@gmail.com",

"chat\_id": "1351500158\_user@rajatogo"

}

]

}

*If user association does not exist*

{

"status": 0,

"result": []

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Serial number does not exist",

"error": {

"code": -114,

"message": "Serial number does not exist"

}

#### }

#### SetRobotUser (Deprecated)

This API is used to associate a user with a robot.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.set_user>

**Parameters**

api\_key your API key

email User Email ID

serial\_number Serial Number of robot

**Success Responses**

*If everything goes fine*

{

"status": 0,

"result": {

"success": true,

"message": "Robot ownership established successfully."

}

}

*If ownership already exists*

{

"status": 0,

"result": {

"success": true,

"message": "This robot ownership relation already exists."

}

}

**Failure Responses**

*If email does not exist*

{

"status": -1,

"message": "Email does not exist",

"error": {

"code": "-112",

"message": "Email does not exist."

}

}

*If serial number does not exist*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": "-114",

"message": "Serial number does not exist."

}

#### }

#### DisassociateRobotFromUser

This API is used to Disassociate Robot from User or Users  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.disassociate_user>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of robot

email, User's Email (If this field is empty, it will delete all user association for this particular robot)

**Success Responses**

*If user email is provided and robot user association already exists*

{

"status": 0,

"result": {

"success": true,

"message": "Robot User association removed successfully."

}

}

*If user email is not provided and robot user association exist*

{

"status": 0,

"result": {

"success": true,

"message": "Robot association with all user removed successfully."

}

}

*If robot user association does not exist*

{

"status": 0,

"result": {

"success": true,

"message": "There is no association between provided robot and user"

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": "-114",

"message": "Serial number does not exist."

}

}

*If email is invalid*

{

"status": -1,

"message": "Email does not exist.",

"error": {

"code": "-112",

"message": "Email does not exist."

}

}

#### DeleteRobot

This APIs used to delete a robot. This also cleans up the user/robot association and removes all the entries in the map and the schedule table.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.delete>

**Parameters**

api\_key, your API keys

serial\_number, Serial Number of robot

**Success Responses**

{

"status": 0,

"result": {

"success": true,

"message": "You have deleted robot 123 successfully"

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": -114,

"message": "Robot serial number does not exist"

}

}

#### GetRobotPresenceStatus

This API is used to get robot presence status  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.get_robot_presence_status>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of the robot

**Success Response**

*If robot is online*

{

"status": 0,

"result": {

"online": true,

"message": "Robot 1234 is online."

}

}

*If robot is offline*

{

"status": 0,

"result": {

"online": false,

"message": "Robot 1234 is offline."

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": -114,

"message": "Robot serial number does not exist"

}

#### }

#### PingFromRobot

This API is used by Robot to register it’s presence.

<http://neatostaging.rajatogo.com/api/rest/json?method=robot.ping_from_robot>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of the robot

status, status of robot(optional)

**Success Response**

{

"status": 0,

"result": {

"success": true,

"message": "robot ping have been recorded"

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": -114,

"message": "Robot serial number does not exist"

}

}

#### IsRobotOnlineVirtual

While designing the system, we have taken care to support 2 cases for robot's Wifi connectivity. First, when a Robot maintains a persistent connection with XMPP and second when robot comes online and might go offline for few minutes. The first mode is called “always on” and second mode is called "timed mode". To reflect support for these 2 cases, there are 2 types of presence statuses supported by the system, online and virtual online. If a robot is connected to XMPP server, it is considered online. If a robot is not online but was spotted by the XMPP some time ago, it is considered "virtually online".  What is the time interval in which a robot is considered virtually online can be configured as parameters (sleep time and awake time) at robot level.

This API is used to check whether a robot is virtually online or not.   
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.is_robot_online_virtual>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of the robot

**Success Response**

*If robot is online*

{

"status": 0,

"result": {

"online": true,

"message": "Robot 1 is online.",

"expected\_time": 12

}

}

*If robot is offline*

{

"status": 0,

"result": {

"online": false,

"message": "Robot 1 is offline.",

"expected\_time": -12

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": -114,

"message": "Robot serial number does not exist"

}

}

#### GetRobotTypeMetadataUsingRobotType

This API is used to get robot type metadata using robot type.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.get_robot_type_metadata_using_type>

**Parameters**

api\_key, your API key

robot\_type,Your Robot's Type

**Success Response**

{

"status": 0,

"result": {

"success": true,

"robot\_metadata": {

"type": "1",

"metadata": {

"sleep\_time": "120",

"lag\_time": "15"

}

}

}

}

**Failure Responses**

*If entered robot\_type is invalid*

{

"status": -1,

"message": "Provided robot type is not valid",

"error": {

"code": "-142",

"message": "Robot Type is not valid"

}

}

#### GetRobotTypeMetadataUsingRobotId

This API is used to get robot type metadata using robot Serial number.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.get_robot_type_metadata_using_id>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of the robot

**Success Response**

{

"status": 0,

"result": {

"success": true,

"robot\_metadata": {

"type": "1",

"metadata": {

"sleep\_time": "120",

"lag\_time": "15"

}

}

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": -114,

"message": "Robot serial number does not exist"

}

}

#### SetRobotConfiguration

This API is used to set robot configuration. With this sleep time, wakeup time can be set.   
  
http://neatostaging.rajatogo.com/api/rest/json?method=robot.set\_robot\_configuration

**Parameters**

api\_key your API key

serial\_number Serial Number of the robot

sleep\_time Sleep Time of the robot in seconds

wakeup\_time Wakeup time of the robot in seconds

config\_key\_value Robot Configuration Key Value (Optional)

**Success Response**

{

"status": 0,

"result": {

"success": true,

"timestamp": 1370422277

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Missing parameter sleep\_time in method robot.set\_robot\_configuration",

"error": {

"code": "-114",

"message": "Serial number does not exist."

}

}

*If provided sleep time or wakeup time is invalid*

{

"status": -1,

"message": "Please enter valid sleep time or wakeup time",

"error": {

"code": "-156",

"message": "Please enter valid sleep time or wakeup time"

}

}

#### SetRobotConfiguration2

This API is used to set the robot configuration data.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.set_robot_configuration2>

Parameters

api\_key, your API key

serial\_number, Serial Number of the robot

sleep\_time, Sleep Time of the robot in seconds

wakeup\_time, Wakeup time of the robot in seconds

robot\_type, 100 OR 101 (100 for Basic and 101 for Advanced)

config\_key\_value,Robot Configuration Key Value (Optional)

**Success Response**

{

"status": 0,

"result": {

"success": true,

"timestamp": 1370422277

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Missing parameter sleep\_time in method robot.set\_robot\_configuration2",

"error": {

"code": "-114",

"message": "Serial number does not exist."

}

}

*If provided sleep time or wakeup time is invalid*

{

"status": -1,

"message": "Please enter valid sleep time or wakeup time",

"error": {

"code": "-156",

"message": "Please enter valid sleep time or wakeup time"

}

}

*If provided robot\_type is invalid*

{

"status": -1,

"message": "'Robot Type is not valid'",

"error": {

"code": "-142",

"message": "Robot Type is not valid"

}

}

#### GetRobotConfiguration

This API returns the robot configuration  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.get_robot_configuration>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of the robot

**Success Response**

{

"status": 0,

"result": {

"success": true,

"serial\_number": "1",

"sleep\_time": "1",

"wakeup\_time": "100",

"timestamp": "1370527227",

"config\_key\_value": {

"real": "mask",

"real1": "steal1",

"k1": "v1",

"k2": "v2"

}

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": -114,

"message": "Robot serial number does not exist"

}

}

#### ClearRobotData

This API Clears the Robot Data completely.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.clear_robot_association>

**Parameters**

api\_key your API key

serial\_number Serial Number of the robot

email Valid email address

is\_delete 1 for delete the robot and 0 for clear the robot data

**Success Response**

{

"status": 0,

"result": {

"success": true,

"message": "Robot is Cleaned."

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": "-114",

"message": "Serial number does not exist."

}

}

*If email does not exist in system*

{

"status": -1,

"message": "Email does not exist in system",

"error": {

"code": "-112",

"message": "Email does not exist."

}

}

*If Invalid is\_delete Field*

{

"status": -1,

"message": "Please enter 1 for delete robot data and 0 for clear the robot data.",

"error": {

"code": "-183",

"message": "Please enter 1 for delete robot data and 0 for clear the robot data."

}

#### }

#### RequestLinkingCode

This API is used to request the linking code.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robot.request_link_code>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of the robot

**Success Response**

{

"status": 0,

"result": {

"success": true,

"linking\_code": "GaSE",

"expiry\_time": 300

}

}

**Failure Responses**

*If serial number does not exist*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": "-114",

"message": "Serial number does not exist."

}

}

#### LinkToRobot

This API is used to link the robot with the given email address..

<http://neatostaging.rajatogo.com/api/rest/json?method=robot.link_to_robot>

**Parameters**

api\_key, your API key

email, User email address

linking\_code, linking\_code which is get in 'Link To Robot' API

**Success Response**

{

"status": 0,

"result": {

"success": true,

"message": "Request For Robot-User association is done successfully"

}

}

**Failure Responses**

*If email does not exist*

{

"status": -1,

"message": "The email address you have provided does not appear to be a valid email address.",

"error": {

"code": "-105",

"message": "The email address you provided does not appear to be a valid email address."

}

}

*If provide linking\_code is invalid*

{

"status": -1,

"message": "Please enter valid linking\_code",

"error": {

"code": "-154",

"message": "Please enter valid linking\_code"

}

}

*If provide linking\_code has expired*

{

"status": -1,

"message": "Sorry, provided linking\_code is expired",

"error": {

"code": "-155",

"message": "Sorry, provided linking\_code is expired"

}

}

{

"status": -1,

"message": "Robot already has a user associated with it.",

"error": {

"code": "-192",

"message": "Robot already has a user associated with it."

}

}

# Message APIs

#### SendNotificationToAllUsersOfRobot2

This API sends push notifications to all users of a robot.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=message.send_notification_to_all_users_of_robot2>

**Parameters**

api\_key, your API key

serial\_number, Serial Number of Robot.

message, JSON object in '{"notifications":[{"id":"101"}]}' format.

**Success Response:**

{

"status": 0,

"result": {

"success": true

}

}

**Failure Responses**

*If JSON message is invalid*

{

"status": -1,

"message": "The json message you have provided does not appear to be a valid.",

"error": {

"code": "-126",

"message": "The JSON Object you have provided does not appear to be a valid."

}

#### }

#### RegisterForNotification

This API is used by SmartApp to register users for notifications  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=message.notification_registration>

**Parameters**

api\_key your API key

user\_email User's email for whom you are storing registration id.

registration\_id Registration Id

device\_type Device Type (1 for 'Android Device', 2 for 'iPhone Device' ).

application\_id Application Id (optional)

notification\_server\_type Notification Server Type (optional)

**Success Response**

{

"status": 0,

"result": {

"success": true,

"message": "Registered successfully"

}

}

**Failure Responses**

*If user's email not found:*

{

"status": -1,

"message": "Sorry, Provided user email address does not exist in our system.",

"error": {

"code": "-112",

"message": "Email does not exist."

}

}

*If provided an invalid email address:*

{

"status": -1,

"message": "Please enter valid email address.",

"error": {

"code": "-105",

"message": "The email address you provided does not appear to be a valid email address."

}

#### }

#### UnregisterFromNotification

This API is used to unregister user from receiving the notifications.   
  
[http://neatostaging.rajatogo.com/api/rest/json?method=message.notification\_unregistration](http://neatodev.rajatogo.com/api/rest/json?method=message.notification_unregistration)

**Parameters**

api\_key your API key

registration\_id Registration Id

**Success Response**

{

"status": 0,

"result": {

"success": true,

"message": "Unregistered successfully"

}

}

**Failure Responses**

*If registration is not found*

{

"status": -1,

"message": "Sorry, Provided registration id does not exist in our system",

"error": {

"code": "-125",

"message": "Invalid registration IDs"

}

#### }

#### Set User Push Notification Options

POST method to Set User Push Notification Options.   
  
<http://neatostaging.rajatogo.com/api/rest/json?method=message.set_user_push_notification_options>

**Parameters**

api\_key : your API key

email : email address of user

json\_object : JSON Ojbect in '{"global":"true", "notifications":[{"key":"101", "value":"true"}, {"key":"102", "value":"true"}, {"key":"103", "value":"true"}]}' format

**Success Response**

*When user enters push notification preference first time,*

{

"status": 0,

"result": {

"success": true,

"message": "Successfully saved user push notification preferences."

}

}

*When user tries to set push notification preference again,*

{

"status": 0,

"result": {

"success": true,

"message": "Successfully updated user push notification preferences."

}

}

**Failure Responses**

*If given email address is not valid*

{

"status": -1,

"message": "The email address you have provided does not appear to be a valid email address.",

"error": {

"code": -105,

"message": "The email address you have provided does not appear to be a valid email address"

}

}

*If given email address does not exist in database*

{

"status": -1,

"message": "Sorry, email address that you provided does not exist our database",

"error": {

"code": -112,

"message": "Email does not exist."

}

}

*If given json\_object is not valid*

{

"status": -1,

"message": "The JSON Object you have provided does not appear to be a valid.",

"error": {

"code": "-126",

"message": "The JSON Object you have provided does not appear to be a valid."

}

#### }

#### GetUserPushNotificationOptions

User can set the preferences for the type of notifications that he wants to receive.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=message.get_user_push_notification_options>

**Parameters**

api\_key, your API key

email, email address of user

**Success Response**

{

"status": 0,

"result": {

"global": true,

"notifications": [

{

"key": "101",

"value": true

},

{

"key": "102",

"value": true

},

{

"key": "103",

"value": true

}

]

}

}

**Failure Responses**

*If given email address is not valid*

{

"status": -1,

"message": "The email address you have provided does not appear to be a valid email address.",

"error": {

"code": "-105",

"message": "Theemailaddressyouprovideddoesnotappeartobeavalidemailaddress."

}

}

*If given email address does not exist in database*

{

"status": -1,

"message": "Sorry, email address that you provided does not exist our database",

"error": {

"code": -112,

"message": "Email does not exist."

}

}

# Robot schedule APIs

#### PostRobotScheduleData.

This API is used to post robot’s schedule data.

There are two types of schedule data we have support for– JSON and Blob.

Currently we only use JSON schedule data and all the discussion will be surrounded around XML schedule data. All the parameters returned in the web-service calls regarding blob data can be ignored.

Note: The key for json data is still “xml\_data” as previously we supported schedule for xml data but we switched to json on the client side.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robotschedule.post_data>

**Parameters**

api\_key your API keyserial\_number Serial Number of robot

schedule\_type Basic OR Advanced

xml\_data JSON data for robot schedule (Optional)

**Scenarios**

* If JSON data is not provided
  + It would create json file with blank data

**Success Response**

{

"status": 0,

"result": {

"success": true,

"robot\_schedule\_id": "5",

"schedule\_type": "Advanced",

"xml\_data\_version": 1,

"blob\_data\_version": 1,

"schedule\_version": "1"

}

}

**Failure Responses**

*If schedule type not valid or missing*

{

"status": -1,

"message": "Robot schedule type is not valid",

"error": {

"code": -133,

"message": "Robot schedule type is not valid."

}

#### }

#### GetRobotSchedules

This API returns all the schedules for a robot.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robotschedule.get_schedules>

**Parameters**

api\_key your API key

serial\_number Serial number of robot

**Success Responses**

{

"status": 0,

"result": [

{

"id": "1",

"schedule\_type": "Advanced",

"xml\_data\_version": "1",

"blob\_data\_version": "1"

},

{

"id": "6",

"schedule\_type": "Basic",

"xml\_data\_version": "1",

"blob\_data\_version": "1"

}

]

#### }

#### GetRobotScheduleData

This API returns the Robot’s schedule data for pass schedule ID.

<http://neatostaging.rajatogo.com/api/rest/json?method=robotschedule.get_data>

**Parameters**

api\_key, your API key

robot\_schedule\_id, Robot Schedule Id

**Success Responses**

{

"status": 0,

"result": {

"schedule\_type": "Basic",

"xml\_data\_url": "http://localhost/Neato\_Server/Server\_Yii/Neato/robot\_data/34/xml/1353061075.xml",

"blob\_data\_url": "http://localhost/Neato\_Server/Server\_Yii/Neato/robot\_data/34/blob/Koala.jpg"

}

}

**Failure Responses**

If robot schedule id does not exist

{

"status": -1,

"message": "Robot schedule id does not exist",

"error": {

"code": "-134",

"message": "Robot schedule id does not exist."

}

}

#### UpdateRobotScheduleData

This API is used to update the schedule data of a robot.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robotschedule.update_data>

**Parameters**

api\_key your API key

robot\_schedule\_id Robot Schedule Id

schedule\_type Basic OR Advanced (Optional)

xml\_data\_version JSON data version

xml\_data JSON data for robot schedule (Optional)

**Scenarios**

* If XML data version provided and xml data field is blank, it would update previous xml data with blank data

**Success Responses**

*If XML data version is provided:*

{

"status": 0,

"result": {

"success": true,

"message": "You have successfully updated robot schedule data.",

"schedule\_version": "XML Data Latest Version"

}

}

**Failure Response**

*If robot schedule id does not exist:*

{

"status": -1,

"message": "Robot schedule id does not exist",

"error": {

"code": "-134",

"message": "Robot schedule id does not exist."

}

}

*If schedule type not valid or missing*

{

"status": -1,

"message": "Robot schedule type is not valid",

"error": {

"code": -133,

"message": "Robot schedule type is not valid."

}

}

*If both the data versions are missing*

{

"status": -1,

"message": "Provide at least one data version(xml or blob) or schedule type.",

"error": {

"code": "-128",

"message": "Provide at least one data version(xml or blob) or schedule type."

}

}

*If provided JSON data version does not match with the latest JSON data version*

{

"status": -1,

"message": "Version mismatch for schedule xml data.",

"error": {

"code": "-129",

"message": "Version mismatch for xml data."

}

}

#### DeleteRobotScheduleData

This API is used to delete a specific robot Schedule.  
  
<http://neatostaging.rajatogo.com/api/rest/json?method=robotschedule.delete_data>

**Parameters**

api\_key your API key

robot\_schedule\_id Robot Schedule Id

**Success Response**

{

"status": 0,

"result": {

"success": true,

"message": "You have successfully deleted robot schedule data."

}

}

**Failure Responses**

*If robot schedule does not exist:*

{

"status": -1,

"message": "Robot schedule id does not exist",

"error": {

"code": "-134",

"message": "Robot schedule id does not exist."

}

#### }

#### GetScheduleBasedOnType

This API returns either the basic or the advanced schedule for a robot.

<http://neatostaging.rajatogo.com/api/rest/json?method=robotschedule.get_schedule_based_on_type>

**Parameters**

api\_key your API key

robot\_serial\_number Serial Number of Robot

schedule\_type Robot Schedule Type (1 for 'Basic'2 for 'Advance')

**Success Response**

{

"status": 0,

"result": [

{

"schedule\_id": "68",

"schedule\_type": "Advanced",

"schedule\_version": "1",

"schedule\_data": "{\"book\":{\"@attributes\":{\"id\":\"bk101\"},\"author\":\"Gambardella, Matthew\",\"title\":\"XML Developer's Guide\",\"genre\":\"Computer\",\"price\":\"44.95\",\"publish\_date\":\"2000-10-01\",\"description\":\"An in-depth look at creating applications with XML.\"}}"

},

{

"schedule\_id": "71",

"schedule\_type": "Advanced",

"schedule\_version": "1",

"schedule\_data": "{\"book\":{\"@attributes\":{\"id\":\"bk112\"},\"author\":\"Galos, Mike\",\"title\":\"Visual Studio 7: A Comprehensive Guide\",\"genre\":\"Computer\"}}"

}

]

}

**Failure Responses**

*If robot\_serial\_number does not exist*

{

"status": -1,

"message": "Robot serial number does not exist",

"error": {

"code": "-114",

"message": "Serial number does not exist."

}

}

*If didn't get any schedule data*

{

"status": -1,

"message": "Sorry, we didn't find any schedule data for given robot serial number and schedule type",

"error": {

"code": "-159",

"message": "No schedule data found for this robot"

}

}

End.