IoT Platform

Developer Portal User Manual

Version: 0.0.4

Copyright

IoT Platform

Developer Portal User Manual

Version: 0.0.4

iottac@xpand.asia

Axiata Business Services Sdn Bhd
Axiata Tower, 9 Jalan Stesen Sentral 5,
Kuala Lumpur Sentral, 50490
Kuala Lumpur
Malaysia.
Tel: (+60) 3 2263 8888
web: www.xpand.asia

Contents

Table of Figures

Preface

Overview of the manual

The Developer Portal User Manual is a technical communication document intended to give assistance to the IOT Developers and IOT device vendors, to effectively utilize the XPAND IOT Developer Portal functionalities. This document provides a comprehensive guide to each and every aspect of the portal.

1. Introduction

The Introductory chapter provides an overview of the portal which will be very beneficial for a user to get a basic understanding on the portal functionalities. Moreover, it explains the procedure to start and exit the portal.

2. Getting Started

This Chapter describes the complete process of getting started with the portal. The registration process and the login process for a new user is explained step by step as well as graphically in this chapter.

3. Developer portal

The processes after successful authentication to the portal are described in this chapter. The processes such as adding device brands, adding device types, adding vendor devices, specifying particular tasks to the devices are explained thoroughly in this chapter.

Therefore, as a whole this document provides a complete guide to the developer portal users for successful implementation of custom devices capable of catering XPAND IOT Services.

Acronyms and Abbreviations

This section includes the related acronyms and abbreviations.

- IoT – Internet of Things
- SDK – Software Development Kit
- JSON – Java Script Object Notation
- MAC – Media Access Control
- HTTP – Hypertext Transfer Protocol
- MQTT – Message Queuing Telemetry Transport
- TCP - Transmission Control Protocol
- URL – Uniform Resource Locator
- SCE – Service Creation Environment
- API – Application Programming Interface

1 Introduction

1.1. IoT Developer Portal Overview

The objective of this portal is to facilitate the developers and Internet of Things (IoT) device vendors to successfully onboard their devices. The device vendors and Software Development Kit (SDK) developers can directly interact with the platform, and by utilizing the portal they are able to test and customize their devices as needed. Through the portal the developers can check the compatibility of their IoT devices with the XPAND IoT platform.

The below flow chart depicts the flow of processes in the developer portal.

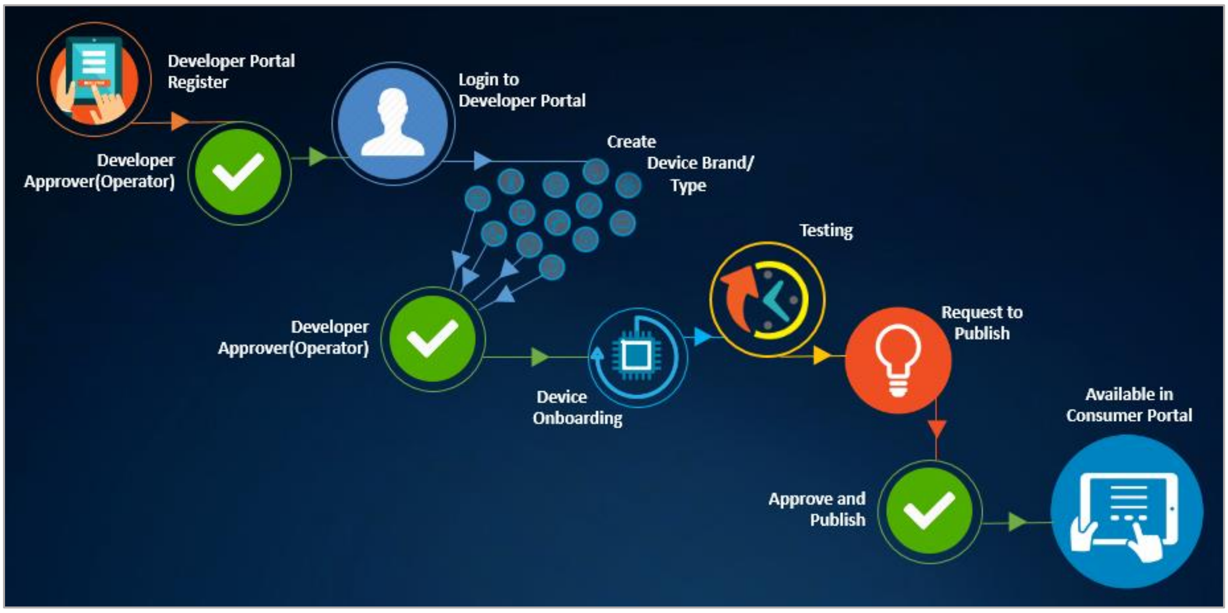

Developer Portal Register
Developer Approver(Operator)
Login to Developer Portal
Create Device Brand/ Type
Developer Approver(Operator)
Device Onboarding
Testing
Request to Publish
Available in Consumer Portal
Approve and Publish

Figure 1- Portal process flow

1.2. Starting the Portal

To access the portal a user should follow these steps;

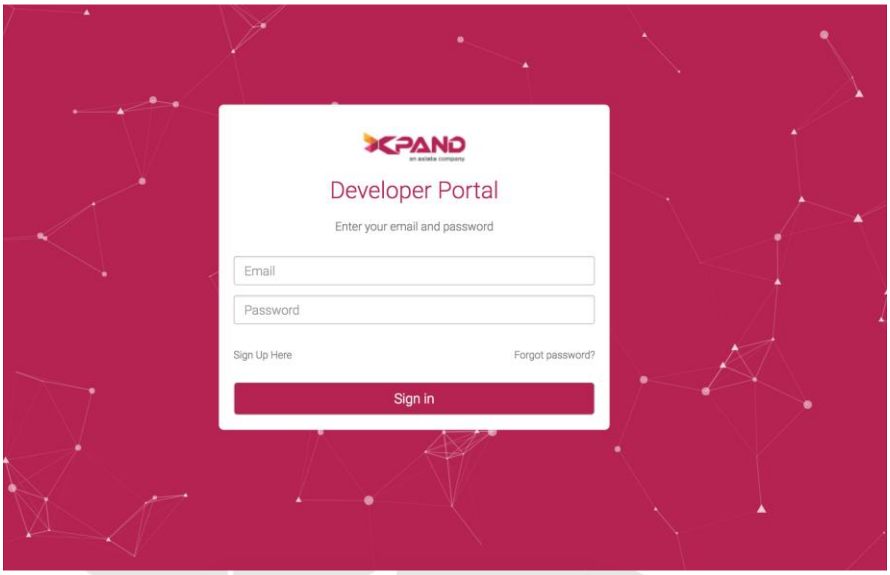
- Type this URL in the web browser and connect to it

  http://iot.xpand.asia

  Developer portal is accessible via

  https://iot.xpand.asia/developer

On successful navigation to the developer portal the below webpage will be visible.



Developer Portal

Enter your email and password

Email

Password

Sign Up Here

Forgot password?

Sign in

Figure 2 - Portal landing page

After accessing the portal's landing page;

- A new user should select the "**Sign Up Here**" option.
- An existing portal user should select the "**Sign in**" option after filling out the two credential fields.

Further instructions on getting started with the portal will be explained in the "Getting Started" section.

1.3. Exit from the Portal

To exit the portal, select on the “Log out” option of the portal menu. This will let the user exit the portal and will direct the user back to the developer portal Sign In page.

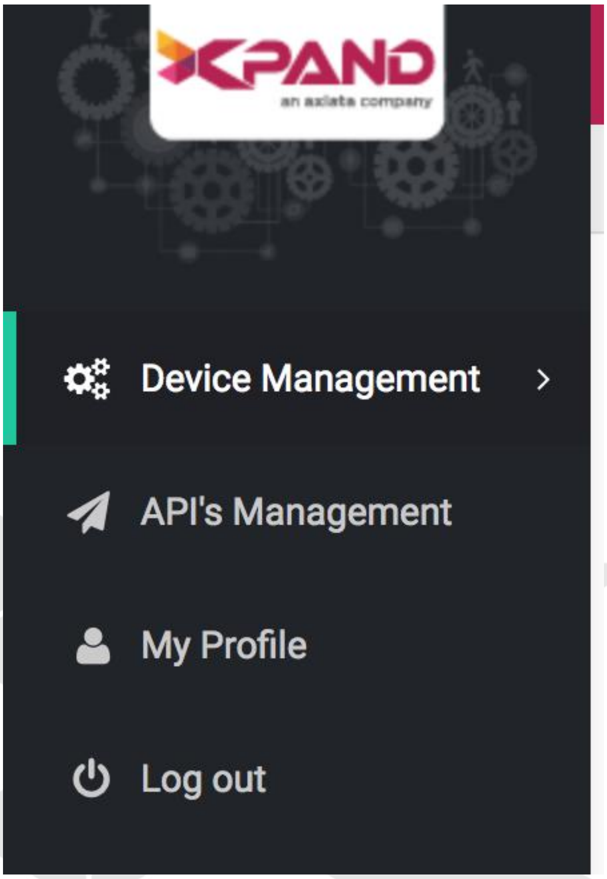


Figure 3 - Portal menu panel

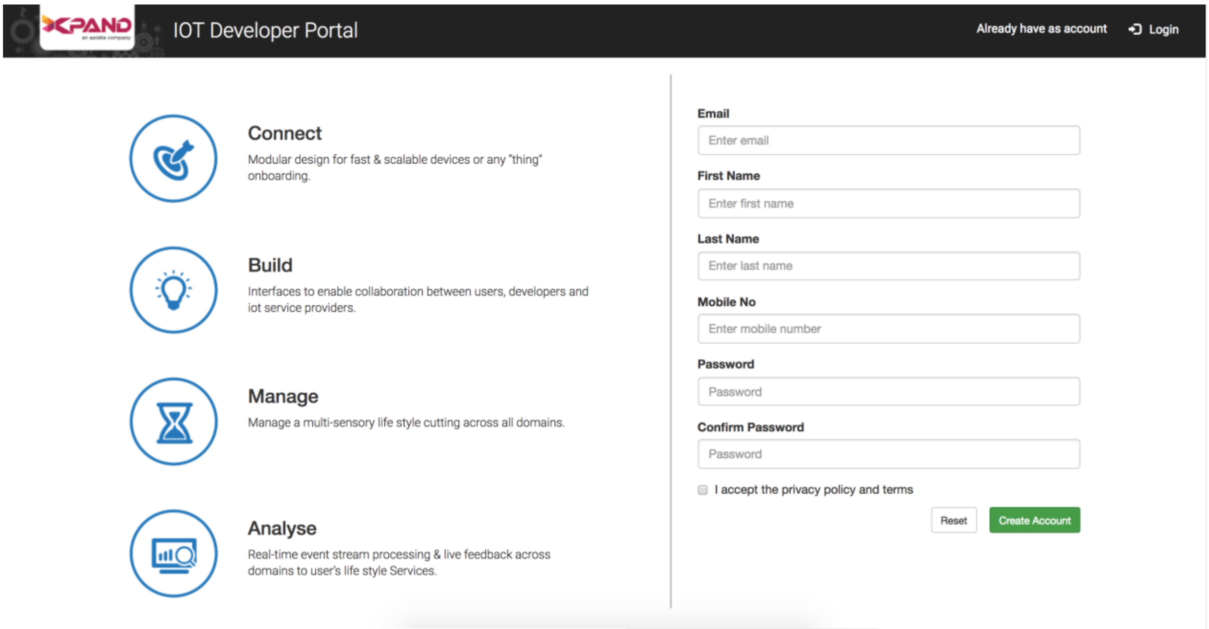
The portal exiting procedure is further explained towards the end of the user manual.

2 Getting Started

2.1 Registering a New User

On successful navigation to the landing page of the portal, a new user has to register him/herself as a new developer to utilize the developer portal functionalities. To register yourself as a new user, the below steps should be followed in the portal login page;

- Click on the “**Sign Up Here**” button to register as a new user.
- Upon clicking the sign up option the user reaches the Sign up page shown below.



IOT Developer Portal

Connect
Modular design for fast & scalable devices or any “thing” onboarding.

Build
Interfaces to enable collaboration between users, developers and iot service providers.

Manage
Manage a multi-sensory life style cutting across all domains.

Analyse
Real-time event stream processing & live feedback across domains to user's life style Services.

Figure 4 – Sign up page

All fields in the sign-up page are mandatory. Therefore, a new user has to fill in all the information fields and follow the steps specified below;

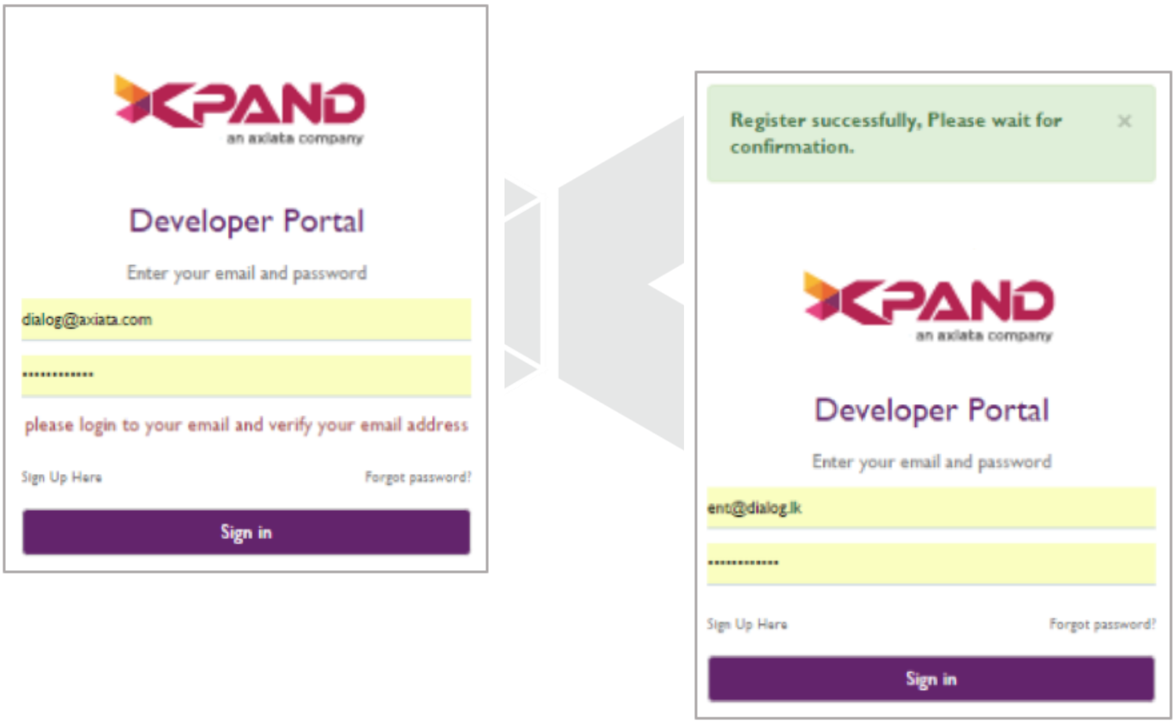
- **Email** – User’s email address
- **First Name** – User’s first name
- **Last Name** – User’s last name
- **Mobile No** – User’s mobile number
- **Password** – A strong password as user prefers
- **Confirm Password** – Re-enter the password entered at the “**Password**” field
- Tick the “ **I accept the privacy policy and terms**” box
- Click on the “**Create Account**” button
- Click on the “**Reset**” button to start filling the fields again, if the user wishes to change the added information.

After implementing the above steps, the new user information will be forwarded to the admins for approval. For a new registration to be successful, below two criteria should be fulfilled;

1. User information approval by the admin
2. Provided user email account confirmation – Login to the email account entered at the registration and verify by clicking the "Email verification URL".

When these two criteria are fulfilled a new user is able to login to the portal for further implementations.

If the criteria are not satisfied, notifying messages as shown below are displayed in the login page.



Developer Portal

Enter your email and password

dialog@axiata.com

•••••••••••••

please login to your email and verify your email address

Register successfully, Please wait for confirmation.

Developer Portal

Enter your email and password

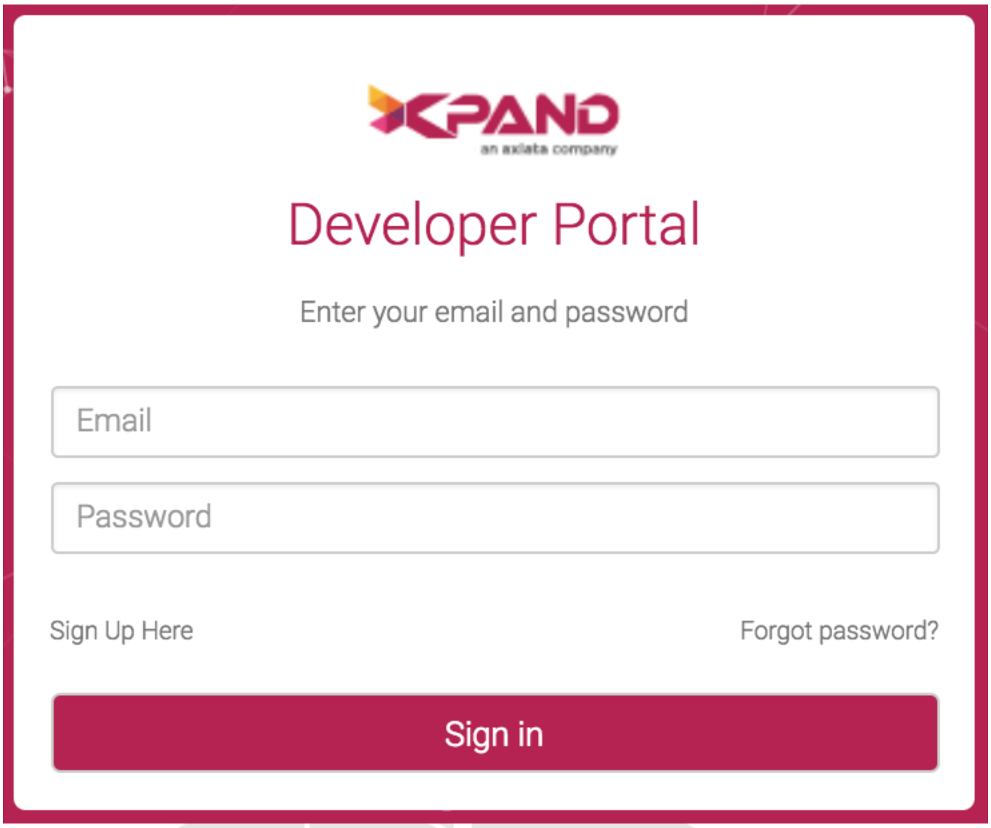
ent@dialog.lk

•••••••••••••

Figure 6 - Wait for admin approval alert

2.2 Logging into the Portal

On successful registration, the user is directed to the login page shown below. The user can use the credentials specified at the registration to login.



Developer Portal

Enter your email and password

Figure 7 - Login Page

- Enter the email address mentioned at the registration in the “**Email**” field.
- Enter the password specified at the registration in the “**Password**” field.
- Click the “**Sign in**” button.

A successful login directs the user to the XPAND IoT homepage shown below.

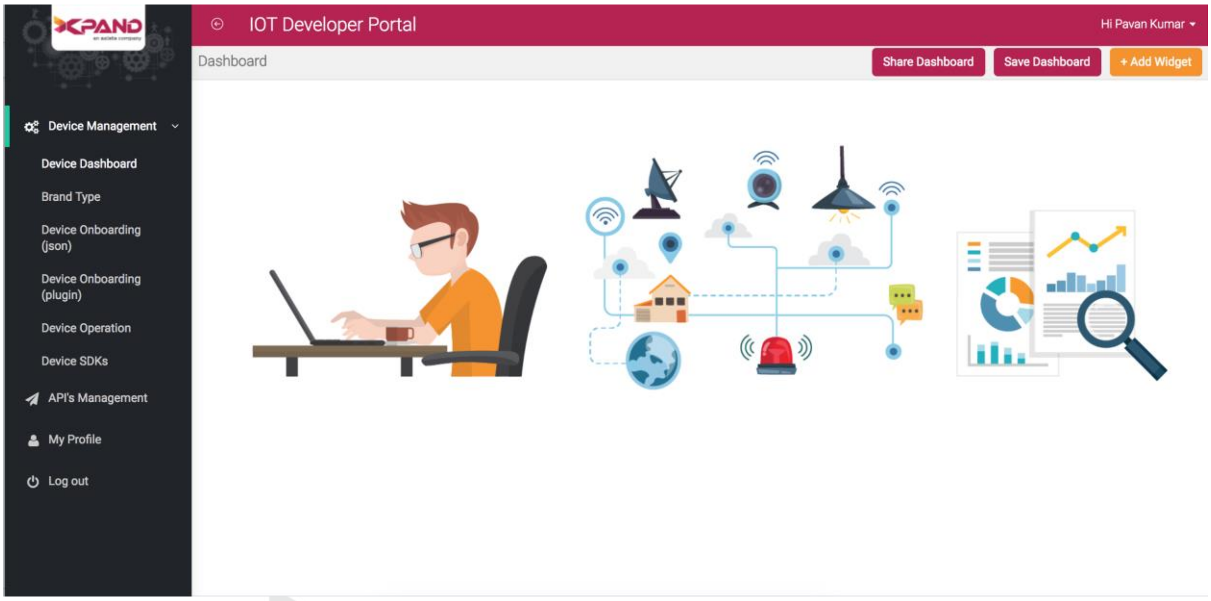


Figure 8 - Portal homepage

In the homepage;

- **A** : the user’s name is shown
- **B** : the portal menu panel is shown
- the body of the page displays the developer’s device dashboard

2.3 View Your Account Details

To view the account details of the user, the “My Profile” option in the portal menu can be used.

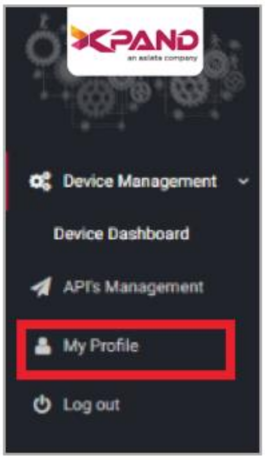
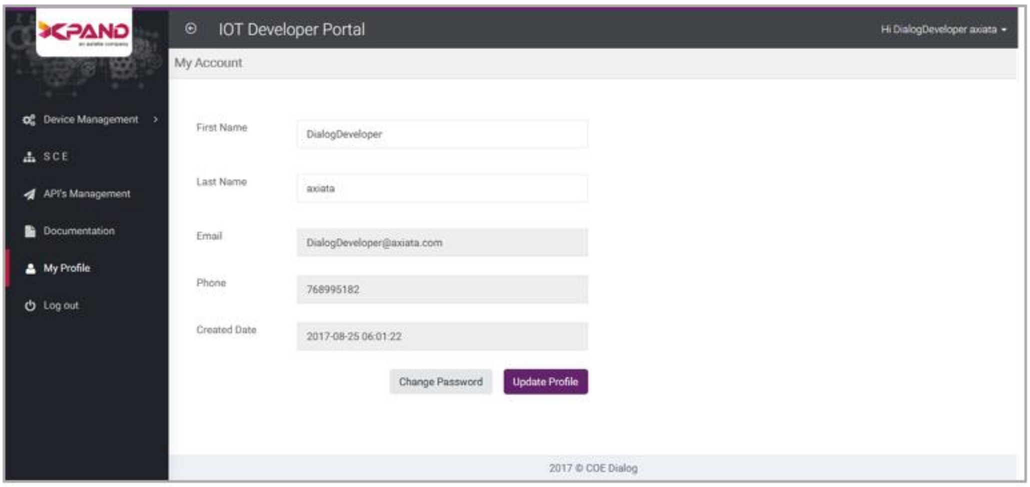


Figure 9 - Menu panel (My Profile)

When the “My Profile” option is clicked the particular user’s account details will be visible as shown below.



My Account

First Name DialogDeveloper

Last Name axiata

Email DialogDeveloper@axiata.com

Phone 768995182

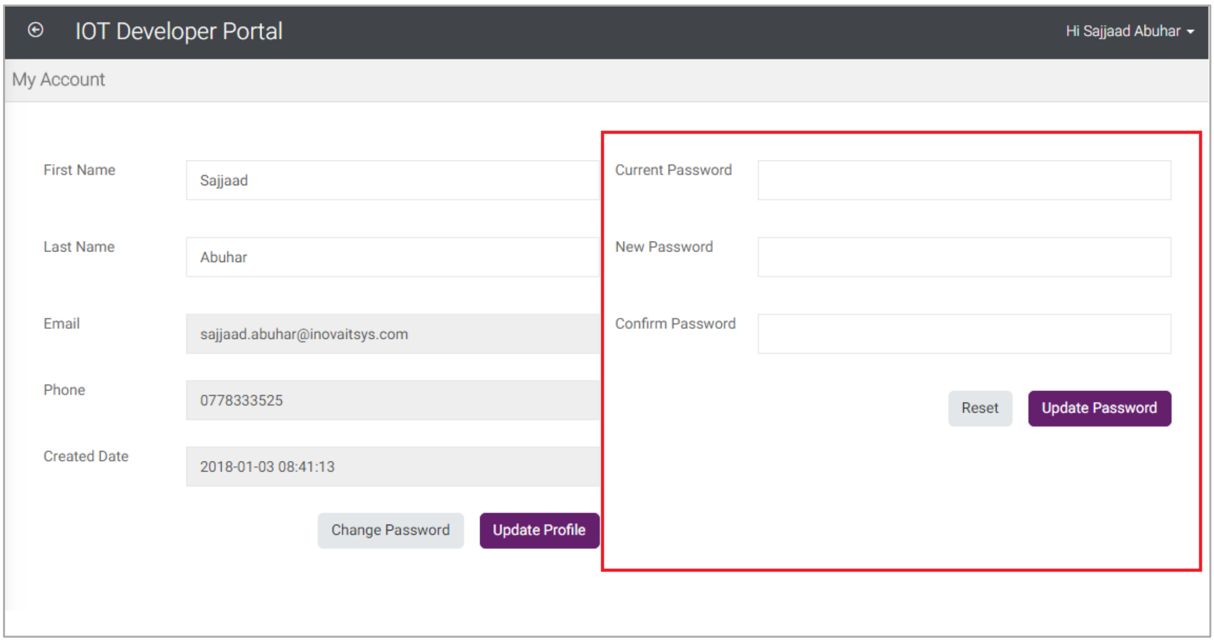
Created Date 2017-08-25 06:01:22

Figure 10 - My profile page

Moreover this option enables the user, to update the profile information and change passwords if necessary. To update the profile details simply do the necessary changes and click the “**Update Profile**” button.

To change the password the below procedure should be followed.

- Click the “**Change Password**” button and the below screen appears.



My Account

First Name	Sajjaad
Last Name	Abuhar
Email	sajjaad.abuhar@inovaitsys.com
Phone	0778333525
Created Date	2018-01-03 08:41:13

Current Password	
New Password	
Confirm Password	

Figure 11 - Change Password

- Fill the three information fields.
    - **Current Password** – Type the user's current password
    - **New password** – Type the new password
    - **Confirm password** – Re-type the password mentioned at "**New Password**"
- After filling the fields click the "**Update Password**" button.
- Click "**Reset**" if you need to change what you mentioned.

3 Developer Portal

3.1 Device Management

After getting started with the portal accordingly, the user is able to perform the developer functionalities, as to prosper the IoT based devices.

The device related functions are available under the “Device Management” option of the portal menu. On clicking the “Device Management” option, a drop down of all device management functions are displayed as shown below.

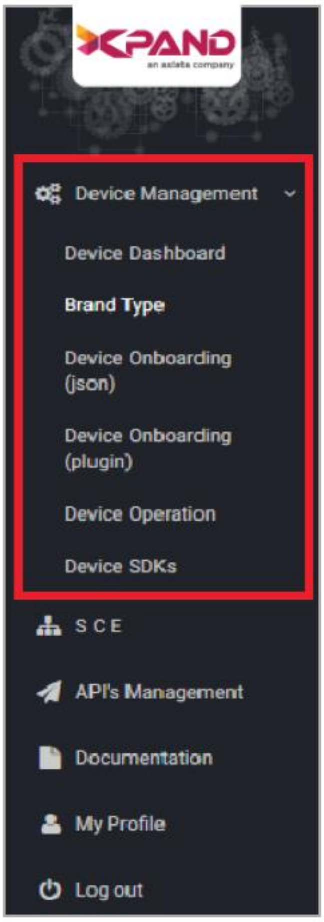


Figure 12 - Device management menu panel

The methods of implementing the above shown device management functions will be clarified further in this chapter.

3.1.1 Device Dashboard

In this portal, the homepage is the Device Dashboard. Therefore, any user on logging into the portal gets to see their Device Dashboard first. The device information added by the developer are displayed in the dashboard.

A new user will see a device dashboard as shown below.

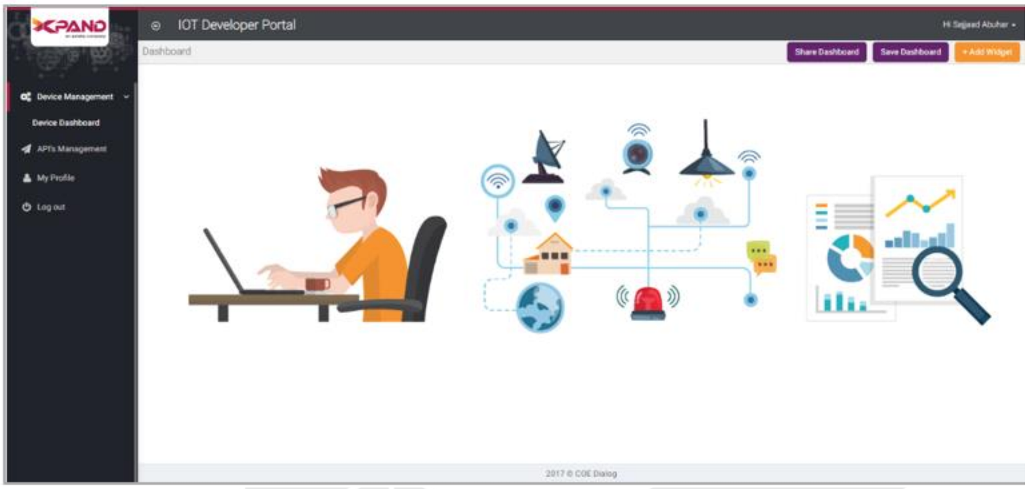


Figure 13 - Device dashboard

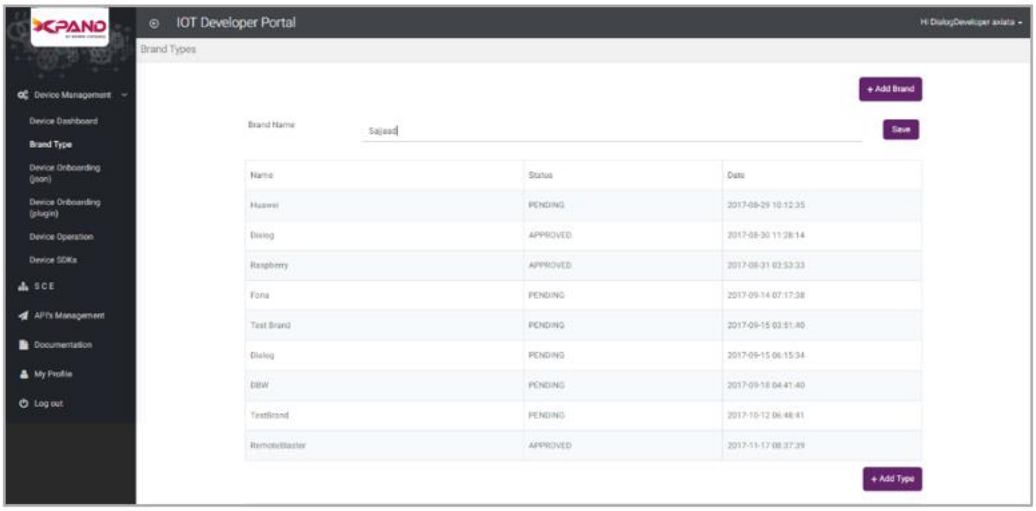
An existing user will be able to see the added device history information in his dashboard.

3.1.2 Brand and Type

The initial step of device management is to add the device brand and type. “Brand Type” function enables the developer to add the brand and type of the device, the user needs to add to the XPAND IoT platform. The vital fact is that, when a brand is added and the device type is specified, **both of them need the admin approval to be recognized as a XPAND IoT accepted device brand and type.**

- Device Brand - the name given by the manufacturer for his range of products, or rather the manufacturer’s trademark.
- Device Type - the device category.

On selecting the “Brand Type” option of the portal menu, the user is able to see the below screen.



Brand Name Sajaad

Name	Status	Date
Huawei	PENDING	2017-08-29 10:12:35
Dialog	APPROVED	2017-08-30 11:28:14
Raspberry	APPROVED	2017-08-31 03:53:33
Fona	PENDING	2017-09-14 07:17:38
Test Brand	PENDING	2017-09-15 03:51:40
Dialog	PENDING	2017-09-15 06:15:34
DBW	PENDING	2017-09-18 04:41:40
TestBrand	PENDING	2017-10-12 06:48:41
RemoteMaster	APPROVED	2017-11-17 08:37:39

Figure 14 - Brand and Type page

3.1.2.1 Add Brand

The process of adding a brand should be done as follows;

Note: Use above figure to refer the steps mentioned below.

– Click on the “**+Add Brand**” button

– Specify the brand name

– Click on “**Save**”

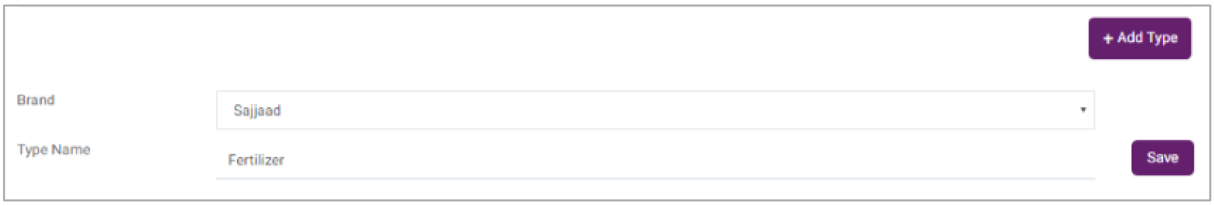
After successfully saving the brand name it is forwarded to the admin for approval. Until the brand is approved, the brand status will be shown as “PENDING”. After approval, the status changes to “APPROVED”.

3.1.2.2 Add Type

Once the brand is approved the user can start adding the device types for the particular brand.

The steps to add a device type is mentioned below;

- Click on “**+Add Type**” button and then the below screen will be visible.



Brand	Sajjaad
Type Name	Fertilizer

Figure 15 - Add device type details

- Choose the device’s brand from the brands dropdown.
- Specify the device type name in the “**Type Name**” field.
- Finally click “**Save**” to successfully add the device type.

The device type status at the “Brand Type” table (Figure 13) will be shown “PENDING” until approved by the admin and after approval the status will change to “APPROVED”.

3.1.3 Device Onboarding

The next step in the process of adding a device to the XPAND IoT platform is, device onboarding. Onboarding is the process of integrating a new device into the XPAND IoT framework. Each device type has to be on-boarded separately as to provide IoT services. In XPAND IOT framework, device onboarding can be done in two methods based on the communication technique of the device.

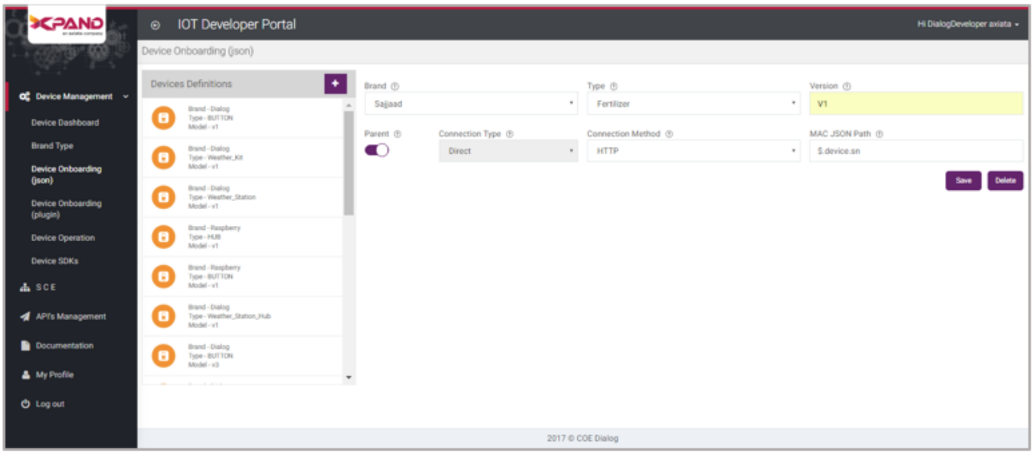
3.1.3.1 Onboarding Devices That Use JSON to Communicate

To onboard devices in this method **“Device Onboarding (JSON)”** option of the portal menu should be used. This method of onboarding can be done if the particular **device uses JSON format to communicate**. Whereas JSON requests and JSON responses are forwarded to and fro the system backend and the device.

Under this method JSON paths should be specified for the purpose of unique identification of necessary elements in the JSON structure. **JSONpath is a tool that allows to pick elements within a JSON structure**.

E.g.: A device forwards a JSON message to the system and the user needs to know the exact particular device that forwarded this JSON. At this the “MAC JSON path” can be used to correctly pick the MAC address element within the communicated JSON, as to identify the device correctly.

On choosing the “Device Onboarding (JSON)” option in the portal menu, the user navigates to the below shown webpage.



Brand - Dialog
Type - BUTTON
Model - v1

Brand - Dialog
Type - Weather_Kit
Model - v1

Brand - Dialog
Type - Weather_Station
Model - v1

Brand - Raspberry
Type - HUB
Model - v1

Brand - Raspberry
Type - BUTTON
Model - v1

Brand - Dialog
Type - Weather_Station_Hub
Model - v1

Brand - Dialog
Type - BUTTON
Model - v3

Brand: Sajjaad
Type: Fertilizer
Version: V1
Parent: ●
Connection Type: Direct
Connection Method: HTTP
MAC JSON Path: $.device.sn

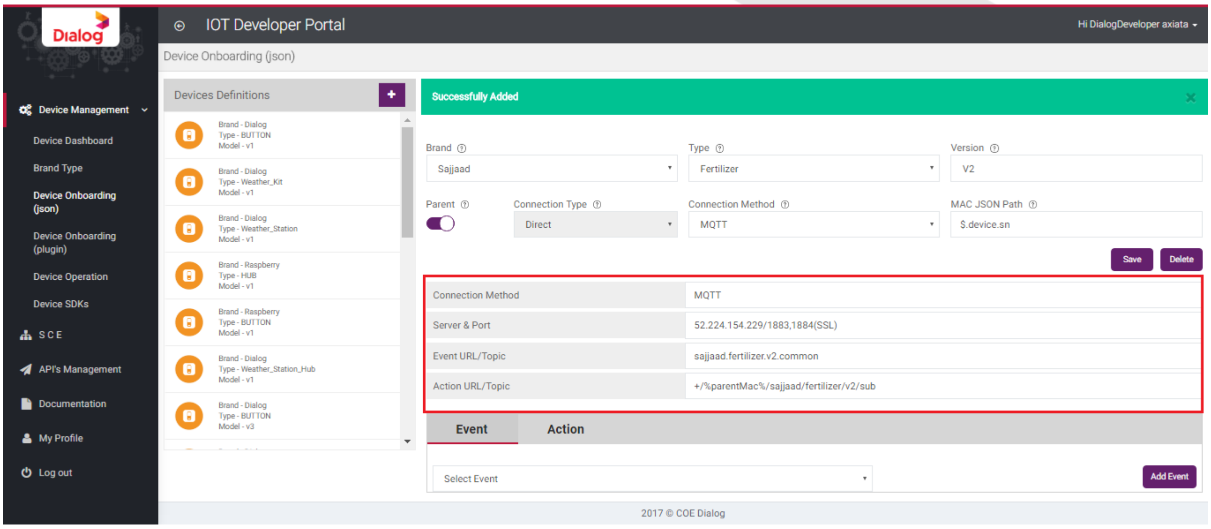
Figure 16 - Device onboarding JSON

3.1.3.1.1 Adding the Device Definitions

The process of adding the device definition in JSON onboarding is mentioned step by step below.

- To add device definitions for a new device, click on the "**+**" box in the top left.
- **Brand** – Select the Brand name you specified in the list of brands.
- **Type** – Select the device type of the particular device in the list of types.
- **Version** – Specify the device version (version should be unique and cannot be repeated again for another device under the same brand and type).
- **Parent** – Choose the "Parent" button based on the below criteria;
    - Turn ON – if the device can connect to the system directly (Parent device).
    - Turn OFF – if the device cannot connect to the system directly, but connects through a hub (Smart device).
- **Connection type** – Will be auto selected if the device is Parent. Unless select the connection type from the dropdown.
- **Connection method** – Can select either method as appropriate, from the two options MQTT or HTTP.
- **MAC JSON path** – Mention the JSON path to capture the device's MAC address.
- Click the "**Save**" button after filling all the fields.

On successfully adding the above device attributes, the below webpages are shown based on the two connection methods MQTT and HTTP.



Successfully Added

Brand: Sajjaad

Type: Fertilizer

Version: V2

Parent: ●

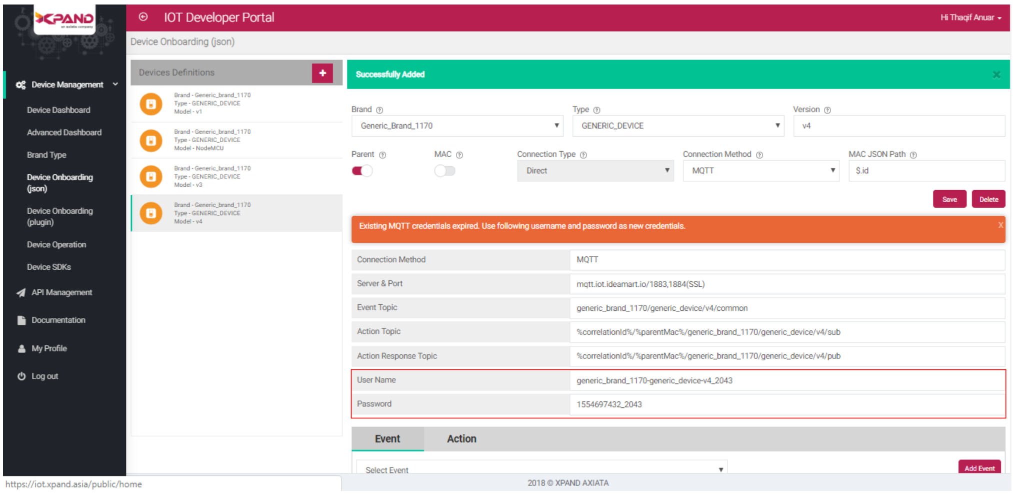
Connection Type: Direct

Connection Method: MQTT

MAC JSON Path: $.device.sn

Connection Method	MQTT
Server & Port	52.224.154.229/1883,1884(SSL)
Event URL/Topic	sajjaad.fertilizer.v2.common
Action URL/Topic	+/%parentMac%/sajjaad/fertilizer/v2/sub

Figure 17 - Device added successfully (MQTT)



Brand - Generic_brand_1170
Type - GENERIC_DEVICE
Model - v1

Brand - Generic_brand_1170
Type - GENERIC_DEVICE
Model - NodeMCU

Brand - Generic_brand_1170
Type - GENERIC_DEVICE
Model - v3

Brand - Generic_brand_1170
Type - GENERIC_DEVICE
Model - v4

Successfully Added

Brand: Generic_Brand_1170
Type: GENERIC_DEVICE
Version: v4
Parent
MAC
Connection Type: Direct
Connection Method: MQTT
MAC JSON Path: $.id

Existing MQTT credentials expired. Use following username and password as new credentials.

Connection Method	MQTT
Server & Port	mqtt.iot.ideamart.io/1883,1884(SSL)
Event Topic	generic_brand_1170/generic_device/v4/common
Action Topic	%correlationId%/%parentMac%/generic_brand_1170/generic_device/v4/sub
Action Response Topic	%correlationId%/%parentMac%/generic_brand_1170/generic_device/v4/pub
User Name	generic_brand_1170-generic_device-v4_2043
Password	1554697432_2043

Figure 18 - Device added successfully (MQTT)

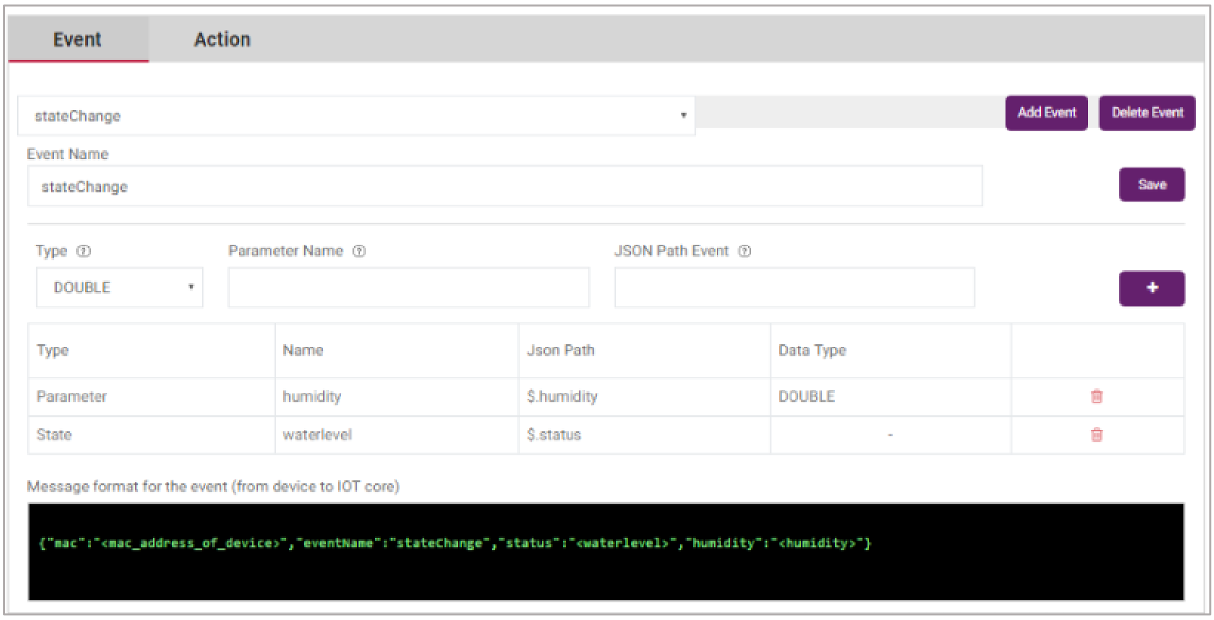
After saving the device attributes, the device configuration information will be visible as seen in the four info fields (within the red boxes) shown in the above screens. To properly work in the framework, the device should be configured using this information.

- **Connection Method** – Displays the device's connection method the user specified before.
- **Server & Port** – Displays the server and port address for communications. Both HTTP & MQTT methods communicate over Transmission Control Protocol (TCP). Therefore, the Server & Port addresses should be added to the device for it to communicate with the framework properly.
- **Event URL/Topic** – Displays the auto generated event URL (if HTTP) or the event topic (if MQTT) specific for the device. Add this to the device.
- **Action URL/Topic** – Displays the auto generated action URL (if HTTP) or the action topic (if MQTT) specific for the device. Add this to the device.

3.1.3.1.2 Adding Device Specific Events

The next step of device onboarding (JSON) is specifying the "Events" for the particular device. Currently IoT devices are augmented with sensors and actuators. Therefore, these devices sense their surroundings and they communicate the sensed data with the system and other devices, to assist the related users in automating the intended processes. Accordingly, if a change in the environment is sensed, the device is made to notify the system. Such a notification on a sensed state change is referred to as an **"Event"**.

The developer should define all the Events specific for the particular IoT device. The process of Event creation is explained step by step below.



stateChange

Event Name: stateChange

Type: DOUBLE

Parameter Name

JSON Path Event

Type	Name	Json Path	Data Type
Parameter	humidity	$.humidity	DOUBLE
State	waterlevel	$.status	-

Message format for the event (from device to IOT core)

```
{"mac":"<mac_address_of_device>","eventName":"stateChange","status":"<waterlevel>","humidity":"<humidity>"}
```

Figure 19 - Add event (JSON)

Upon clicking the “Event” tab on the top, the above screen will be displayed. To add an Event, click the “Add Event” button. Thereafter fill in the shown fields as instructed below.

- **Event Name** – Specify the event name as preferred.
- Click on “**Save**” to save the event name. All saved event names will be visible in the dropdown list.
- **Type** – Specify the event type from the dropdown list.
- **Parameter Name** – Define the parameter name for the event.
- **JSON Path Event** – Specify the JSON path. The specified JSON path is used to correctly capture the event information component from the message.
- Click on “**+**” button to add the specified information on the new event.

On successfully implementing the above procedure, the information the user added on the device will be displayed in a table as shown in the figure above.

The black area at the bottom shows the message format for the event.

3.1.3.1.3 Adding Device Specific Actions

The next step of the device onboarding process is, adding actions.

An “Action” is the process of performing some act by a device, as instructed from the IoT core or rather the developer.

Actions for the particular device can be added in the below mentioned manner;

- Click “**Add Action**” and start filling the information fields.
- **Action Name** – Specify the action name as preferred (Name referred to the system backend)

- **Display Name** – Specify the display name as preferred (Name referred to the front interface)
- **Visibility** – Controls the visibility of the action information to the device end user. An end user can access the action information through the Consumer Portal.
    - Turn ON – If the action should be visible to the end user.
    - Turn OFF – if the action should not be visible to the end user.
- Save the action added by clicking "**Save**". Upon saving, two new information fields will be displayed as shown below. Moreover, the added action will be visible in the "Select Action" dropdown and in the actions table as well.

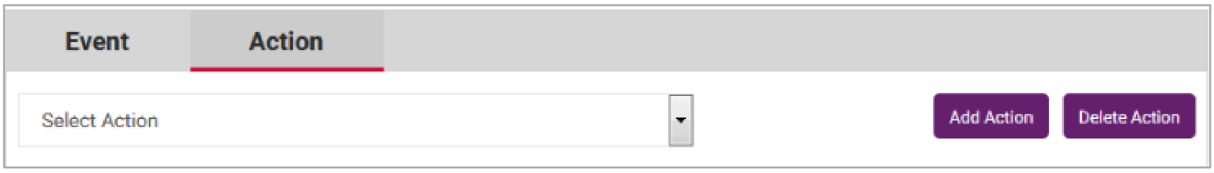
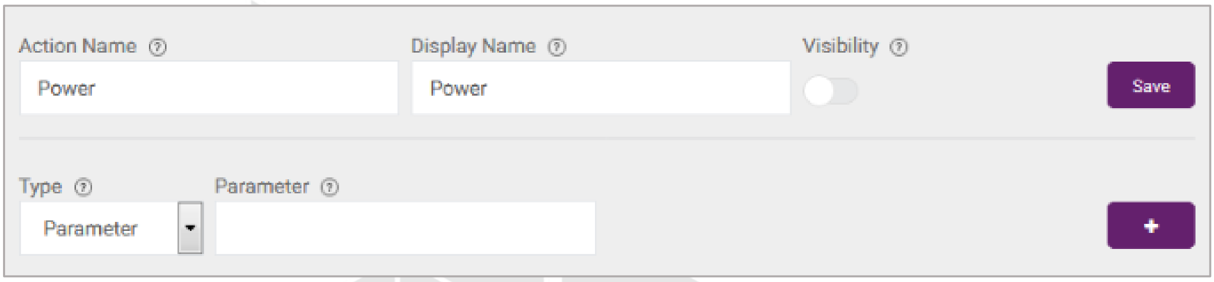


Figure 20 – Action Screen


Action Name: Power

Display Name: Power

Visibility

Type: Parameter

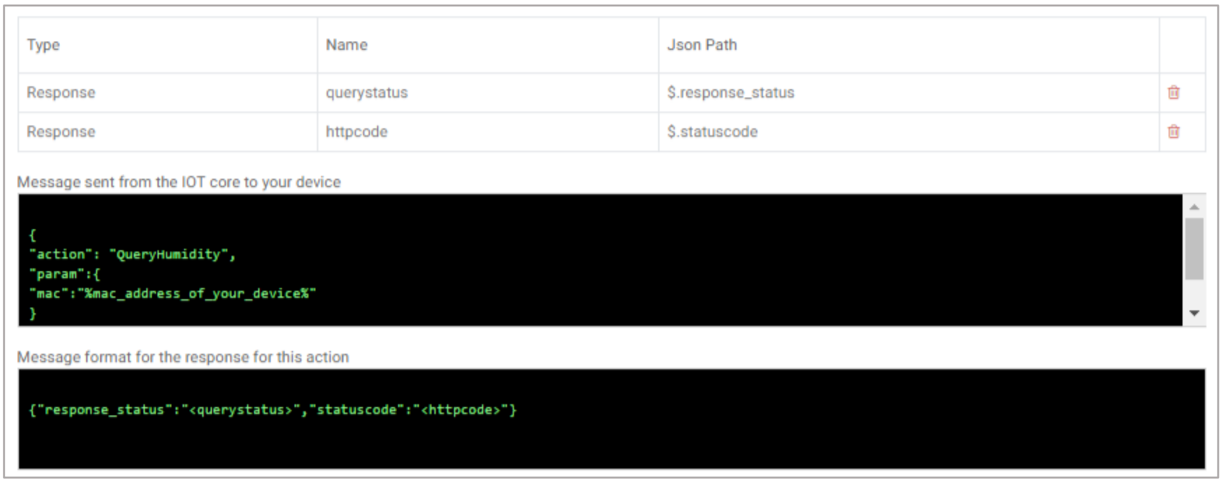
Parameter

Figure 21 - Add action (JSON)

- **Type** – Specify the action type from the list of action types.
- **Parameter** – Specify the action parameter.
- Click "**+**" to successfully finish adding the action parameter.

  Note: For a single action, multiple parameters can be specified. Specify the parameter you need and click "**+**" to add parameters for the action.

On successfully adding the action, a message will be sent from the IoT core to the device, and the message format of the response for the action will be displayed. In the below screen, the two black sections at the bottom depicts these message codes respectively.



Type	Name	Json Path	
Response	querystatus	$.response_status	🗑
Response	httpcode	$.statuscode	🗑

Message sent from the IOT core to your device

```
{
"action": "QueryHumidity",
"param":{
"mac":"%mac_address_of_your_device%"
}
```

Message format for the response for this action

```
{"response_status":"<querystatus>","statuscode":"<httpcode>"}
```

Figure 22 - Actions added successfully (JSON)

3.1.3.2 Onboarding Devices That Don’t Use JSON to Communicate

In order to onboard a particular device that doesn’t use JSON for communication, Plugin device onboarding methodology is implemented.

3.1.3.2.1 Adding the Device Definition

On selecting the “Device Onboarding (Plugin)” option of the portal menu, the user navigates to the below page.

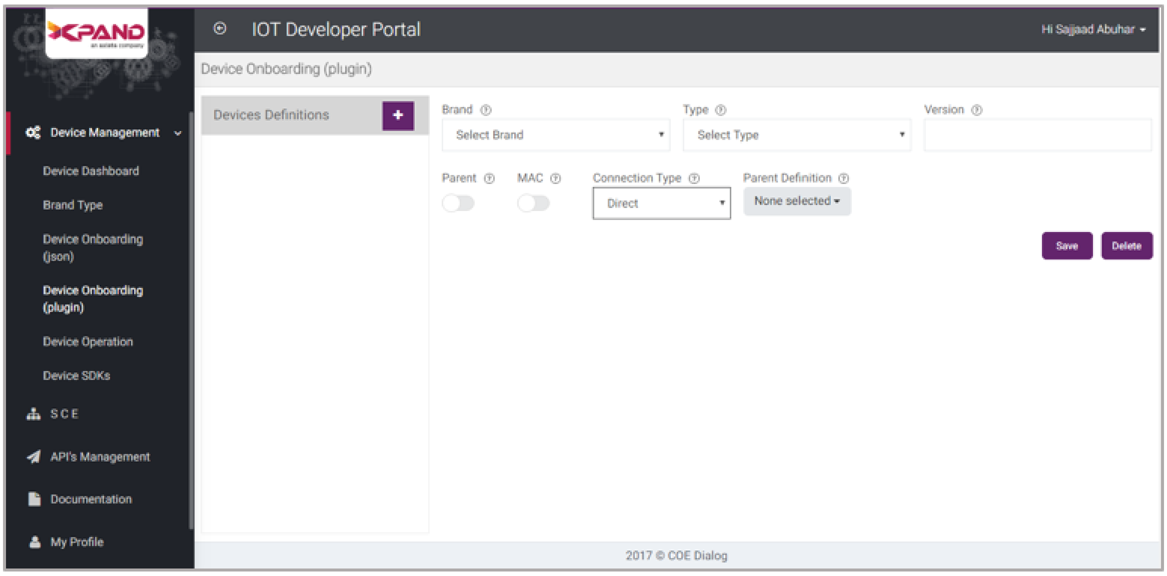
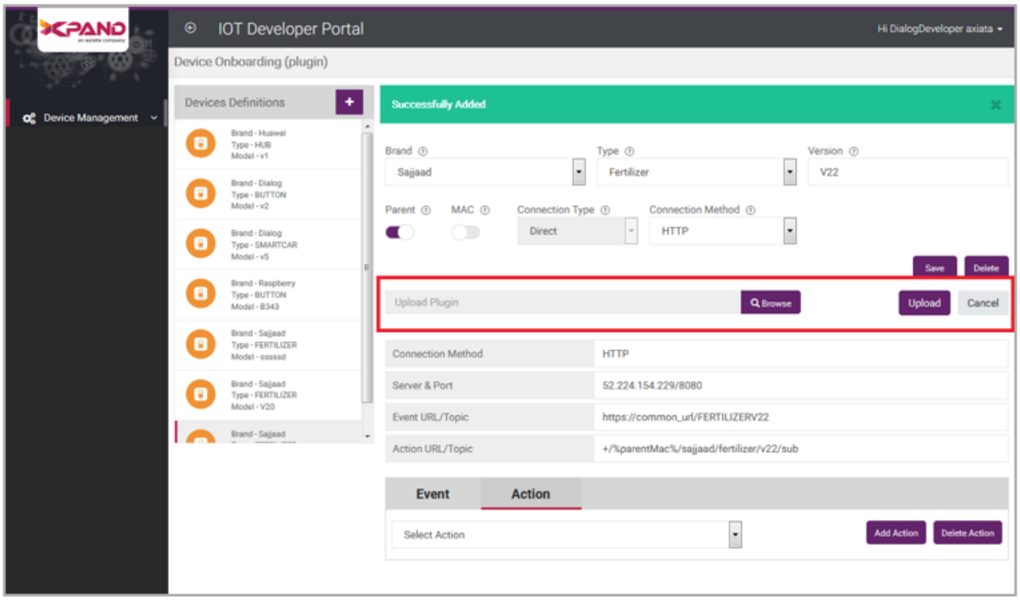


Figure 23 - Device onboarding (plugin)

To add a device, the device definition fields should be filled following the same procedure as in [JSON onboarding.](#)
After saving the device definitions, plugin fields will be visible as shown below.



Brand - Huawei
Type - HUB
Model - v1

Brand - Dialog
Type - BUTTON
Model - v2

Brand - Dialog
Type - SMARTCAR
Model - v5

Brand - Raspberry
Type - BUTTON
Model - B343

Brand - Sajjaad
Type - FERTILIZER
Model - sssssd

Brand - Sajjaad
Type - FERTILIZER
Model - V20

Brand - Sajjaad

Brand: Sajjaad
Type: Fertilizer
Version: V22
Parent
MAC
Connection Type: Direct
Connection Method: HTTP

Connection Method	HTTP
Server & Port	52.224.154.229/8080
Event URL/Topic	https://common_url/FERTILIZERV22
Action URL/Topic	+/%parentMac%/sajjaad/fertilizer/v22/sub

Figure 24 - Add plugin

The Plugin should be created referring the “Plugin Document”. After successfully creating the plugin, it should be uploaded to the portal as to specify for the device. Uploading the created plugin is done as instructed below.

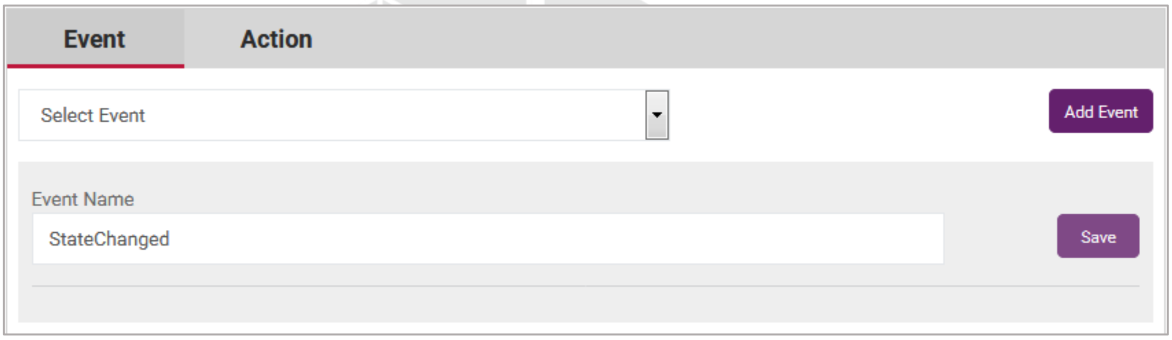
- Click the “**Browse**” button and select the created plugin file.
- Click the “**Upload**” button to successfully upload the plugin.

On successfully adding the device, the device configuration information will be visible. The information fields depict the same information as in the JSON onboarding method. Use this information to configure the device accordingly.

Note: When a device is on-boarded via plugin with the HTTP connection method, the “Action URL” field of the device configuration information is shown empty. This is because, under HTTP method, communications happen only from the device to the IoT core and not the other way around. So no action is present. **But if the device manufacturer is able to assign a public IP for each device then the Action URL is present** as it can perform actions.

3.1.3.2.2 Adding Device Specific Events

In Plugin device onboarding, the user has to specify only the event name to create a new device event. On clicking the “Event” tab the below screen is shown.



Event Name: StateChanged

Figure 25 - Add event (plugin)

The process of adding an Event is described below’

- Click the “**Add Event**” button and fill the information field shown
- **Event Name** – Specify a name for the event as preferred.
- Click on “**Save**” to save the created event.

On successfully saving the event, the newly created event will be displayed in the “Select Event” dropdown.

3.1.3.2.3 Adding device specific actions

Upon selecting the "Action" tab the below screen opens.

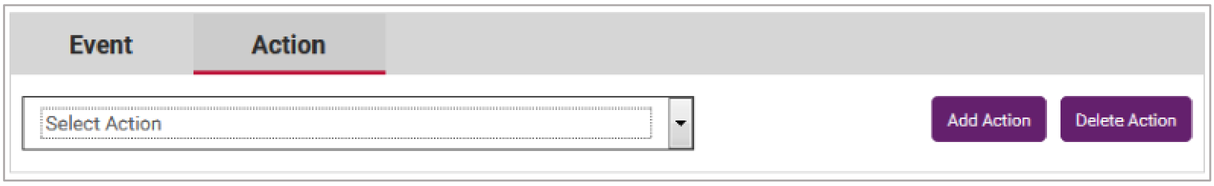
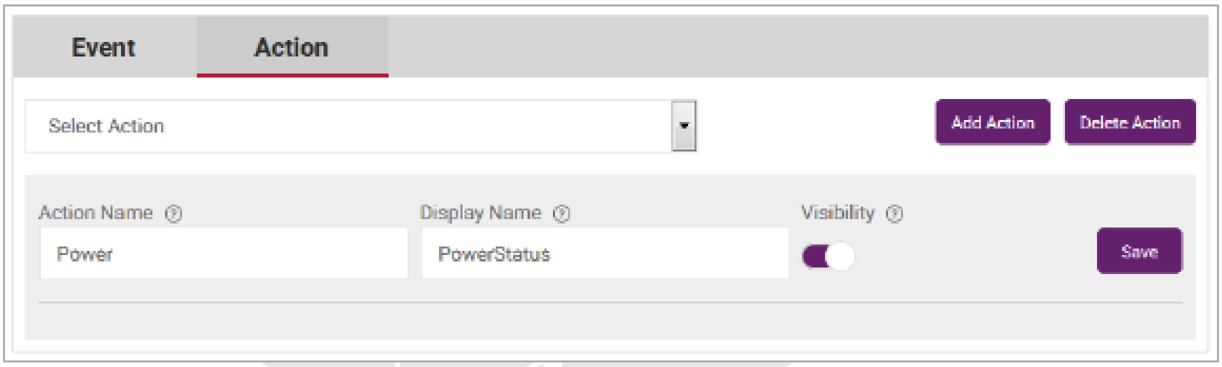


Figure 26 - Add action (plugin) 1

Actions for the particular device can be added in the below mentioned manner;

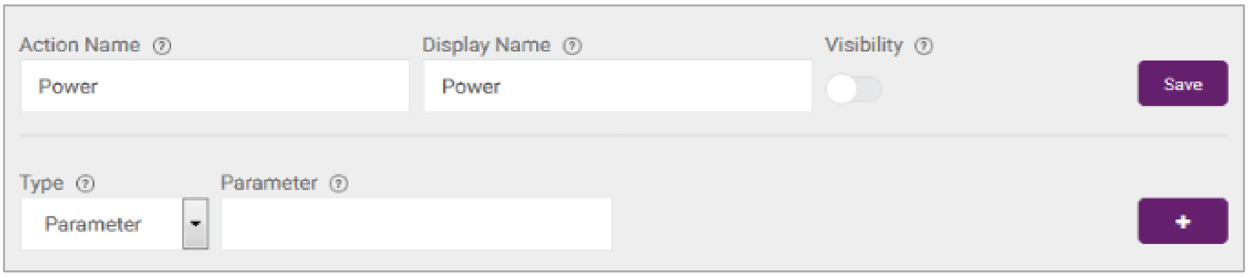
- Click "**Add Action**" and start filling the information fields shown below.



Action Name: Power
Display Name: PowerStatus
Visibility: ON

Figure 27 - Add action (plugin) 2

- **Action Name** – Specify the action name as preferred (Name referred to the system backend)
- **Display Name** – Specify the display name as preferred (Name referred to the front interface)
- **Visibility** – Controls the visibility of the action information to the device end user. An end user can access the action information through the Consumer Portal.
    - Turn ON – If the action should be visible to the end user.
    - Turn OFF – if the action should not be visible to the end user.
- Save the action added by clicking "**Save**". Upon saving, two new information fields will be displayed as shown below. Moreover the saved action will be displayed in the "Select Action" dropdown.



Action Name: Power
Display Name: Power
Visibility: OFF
Type: Parameter
Parameter:

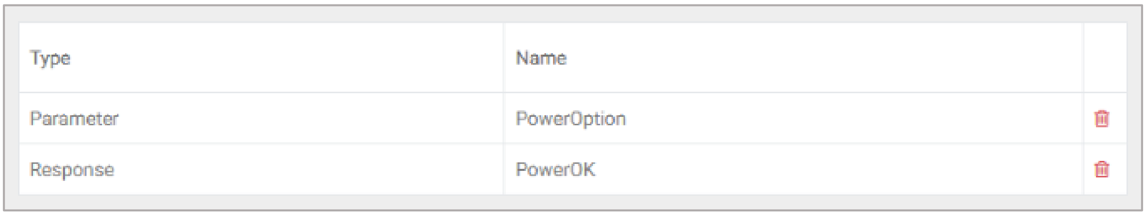
Figure 28 - Add action (plugin) 3

- **Type** – Specify the action type from the list of action types.

- **Parameter** – Specify the action parameter.
- Click “**+**” to successfully finish adding the action parameters.

  Note: For a single action, multiple parameters can be specified. Specify the parameter you need and click “**+**” to add parameters for the action.

On successfully adding the action, the action information table will be formulated and displayed as shown below.



Type	Name	
Parameter	PowerOption	🗑
Response	PowerOK	🗑

Figure 29 - Action added successfully

3.1.4 Device Operation

The “device operation” function is for developer testing purposes. Through this function the developers get the chance to check their devices’ performance, by creating a pretend environment for the device. The developer is able to add his device and specify different events and actions for it. Then he can manually implement the event and check how the system and device responds to each type of event. Therefore, this is a very beneficial testing platform for the developers.

On selecting the “Device Operation” option of the portal menu the user navigates to the below screen.

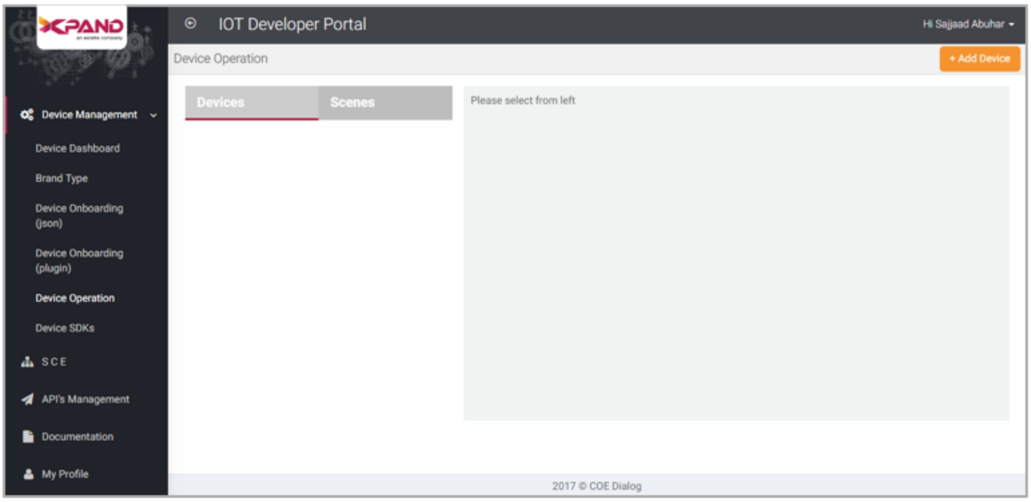
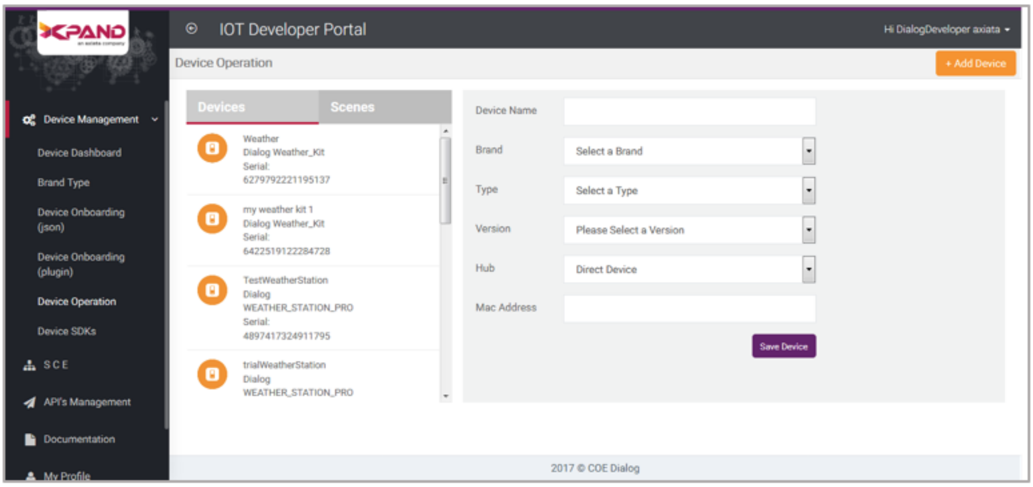


Figure 30 - Device operation page

3.1.4.1 Adding Devices

The first step of the process is to add the device. To add a device the user should first click on the yellow “**+Add Device**” button shown on top right corner of the above shown webpage. After clicking it the user navigates to the below screen with a set of information fields to be filled.



Device Operation

Weather
Dialog Weather_Kit
Serial:
6279792221195137

my weather kit 1
Dialog Weather_Kit
Serial:
6422519122284728

TestWeatherStation
Dialog
WEATHER_STATION_PRO
Serial:
4897417324911795

trialWeatherStation
Dialog
WEATHER_STATION_PRO

Device Name
Brand: Select a Brand
Type: Select a Type
Version: Please Select a Version
Hub: Direct Device
Mac Address

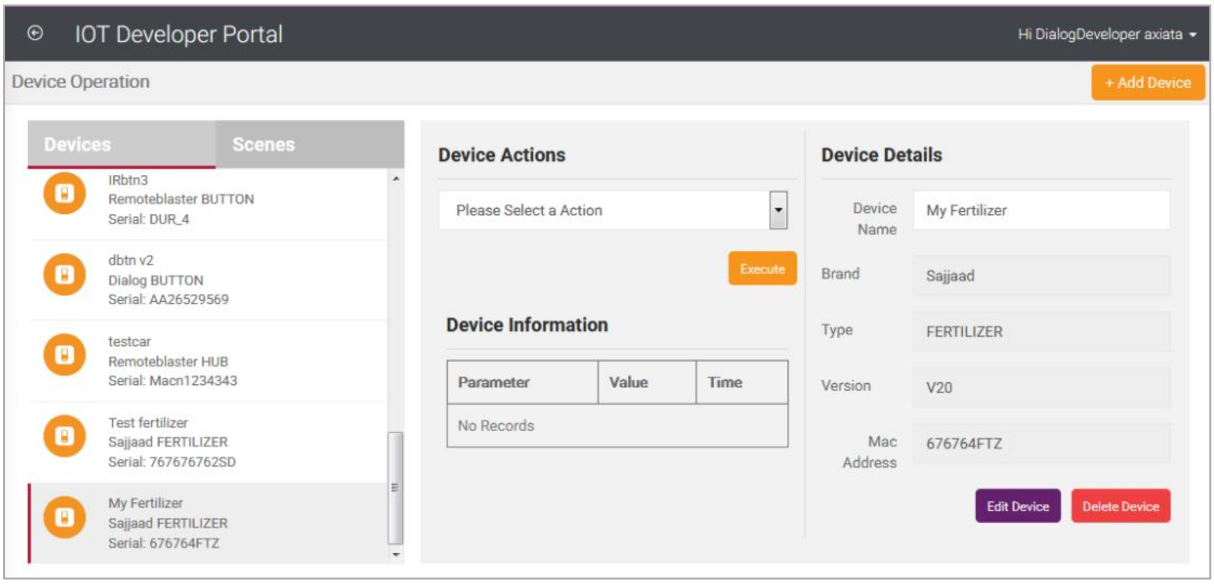
Figure 31 - Device operation (Add devices)

Already added devices are shown under the "Devices" list towards the left of the above screen. Moreover the added device definitions at the onboarding process are shown in the dropdown lists for the ease of the developer. The user can select the necessary attributes for the device he is testing from the already added devices.

The process of adding the device information is explained below.

- **Device Name**: Specify the name of the device.
- **Brand**: Select the device's brand from the dropdown list.
- **Type**: Select the device type from the dropdown list.
- **Version**: Select the device version from the dropdown list.
- **Hub** :
  - Parent device – No need of adding a hub.
  - Smart device – Select the particular hub the device is connected to.
- **MAC Address**: Specify the MAC address of the particular device.
- Click on "**Save Device**" to save the added device.

After filling out the device information and saving it, the user can navigate to the below screen.



IRbtn3
Remoteblaster BUTTON
Serial: DUR_4

dbtn v2
Dialog BUTTON
Serial: AA26529569

testcar
Remoteblaster HUB
Serial: Macn1234343

Test fertilizer
Sajjaad FERTILIZER
Serial: 767676762SD

My Fertilizer
Sajjaad FERTILIZER
Serial: 676764FTZ

Device Actions

Please Select a Action

Device Information

Parameter	Value	Time
No Records		

Device Details

Device Name	My Fertilizer
Brand	Sajjaad
Type	FERTILIZER
Version	V20
Mac Address	676764FTZ

Figure 32 - Device operation (device details)

The added device details are shown in the rightmost section of the above screen.

– To edit the added device information, click “**Edit Device**” and do the necessary changes.

– If the user needs to delete the added device, click “**Delete Device**”.

Under the “Device Actions” section in the middle, the user can select an action from the dropdown and execute it by clicking the “**Execute**” button. This way an action can be checked in the device.

On successfully adding the device, an instance of the added device will be displayed in the “Devices” list.

3.1.4.2 Adding Scenes

The next process of device operations is to add scenes.

Implementing an Action based on an Event is referred to as a **Scene**. Rather a scene can be interpreted as, the process of setting a rule based on an event. At this, an event can be triggered by various means;

- Due to a natural occurrence.
- Due to a manual occurrence.
- Can be time based, to trigger at a pre-set time.

E.g.: A natural occurrence event;

Event: Temperature in the meeting room has risen above 30 C

Action: Turn the Air-conditioner ON.

Scene: Turn the Air-conditioner ON if the temperature rises above 30 C

Therefore to test how the device reacts to different events, the complete scene needs to be created. When the scene is created, the user gets the ability to check how the device works at the particular situation. The process of scene creation is explained below.

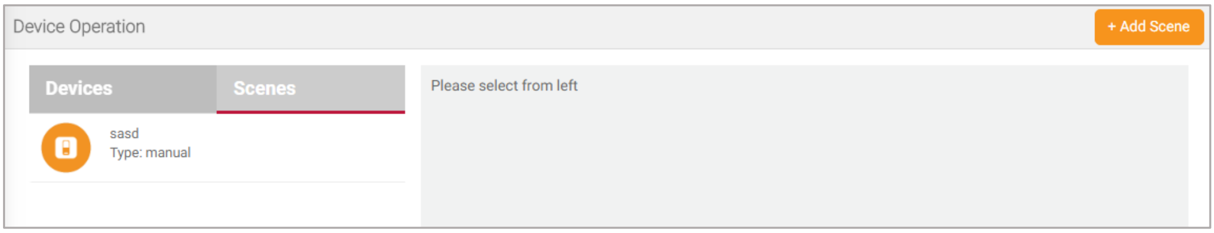


Figure 33 - Add scenes 1

- Select the “Scenes” tab and click “**+Add Scene**” to start scene creation. Upon clicking it the user navigates to the below screen.

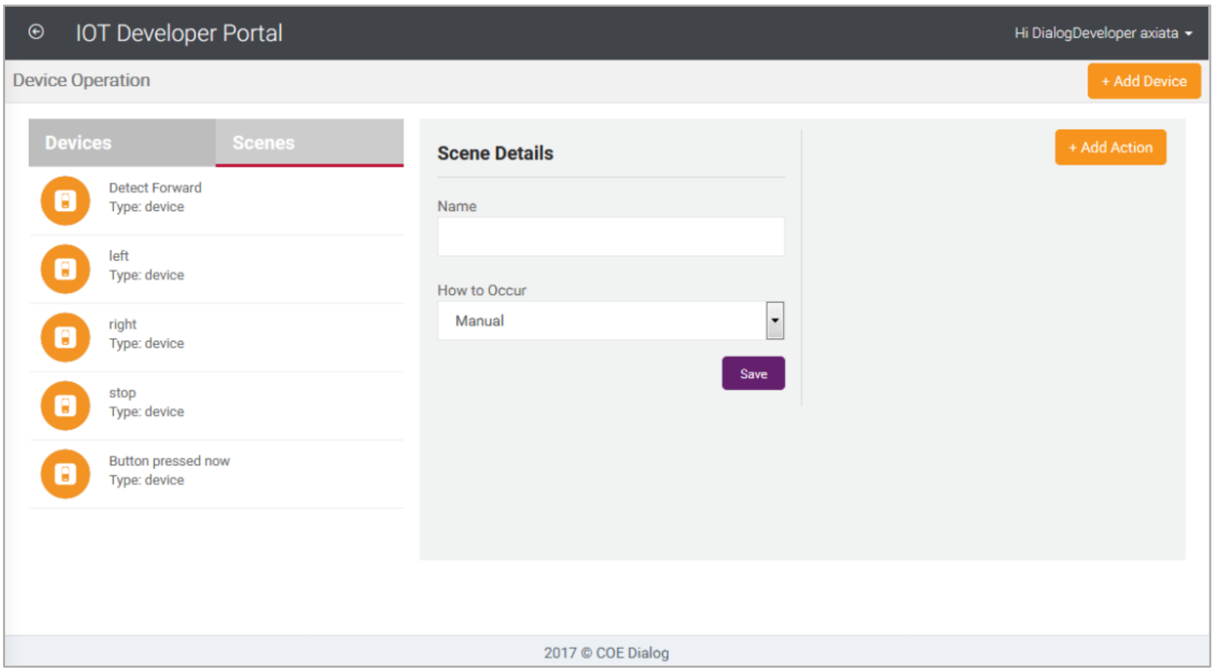
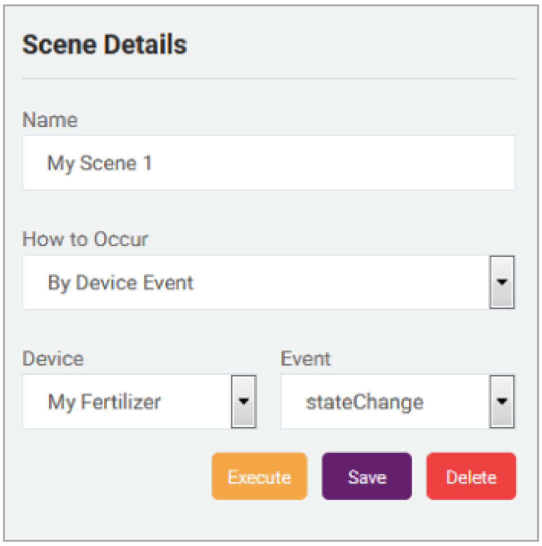


Figure 34 - Add scenes 2

- **Name** – Define a name for the scene.
- **How to Occur** – Select the mode of scene occurrence from the dropdown list.
  - Manual: User manually execute the occurrence of the particular event.
  - By device event: Natural occurrence of the event.
  - By time: The event occurs at a user defined time.
- Click “**Save**” to save the added scene. After saving the scene details the user gets to choose and specify the device and the event for the saved scene. Below screen appears on saving the scene details.

- Choose the Device and the Event from the two respective dropdown lists that appear as shown in the below screen.



Scene Details

Name: My Scene 1

How to Occur: By Device Event

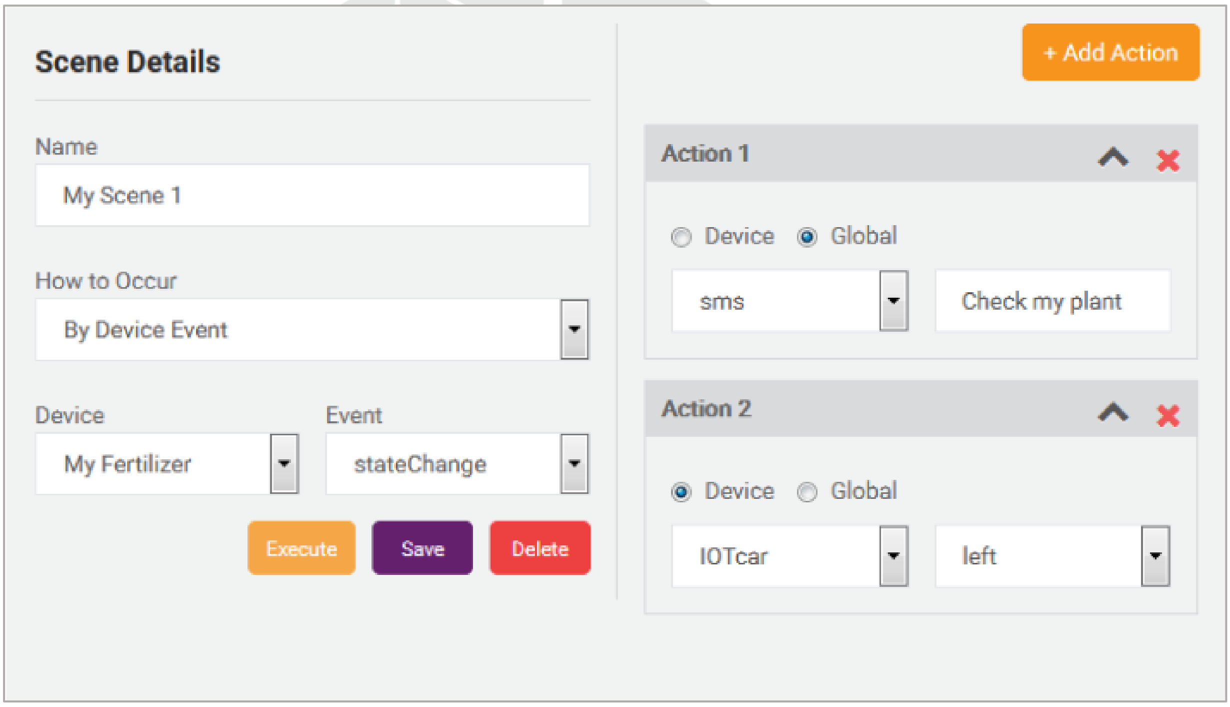
Device: My Fertilizer

Event: stateChange

Figure 35 - Scene details

- Next specify actions for the scene by clicking "**+Add Action**" button. On clicking it the below screen appears.

  A single scene can have multiple actions. So a user can specify many actions for the same scene.



Scene Details

Name: My Scene 1

How to Occur: By Device Event

Device: My Fertilizer

Event: stateChange

Action 1

○ Device ● Global

sms

Check my plant

Action 2

● Device ○ Global

IOTcar

left

Figure 36 - Scene creation (add action)

- Select from the two action modes;
  - **Device** – Select "**Device**" if the action should be a process related to a device

        - Select the particular device from the list of devices.
        - Select the device’s task to perform as the action from the list.
    - **Global** – Select “**Global**” if the action should be a notification.
        - Select the notification method from the list. (SMS or email).
        - Type the notification message in the text box.

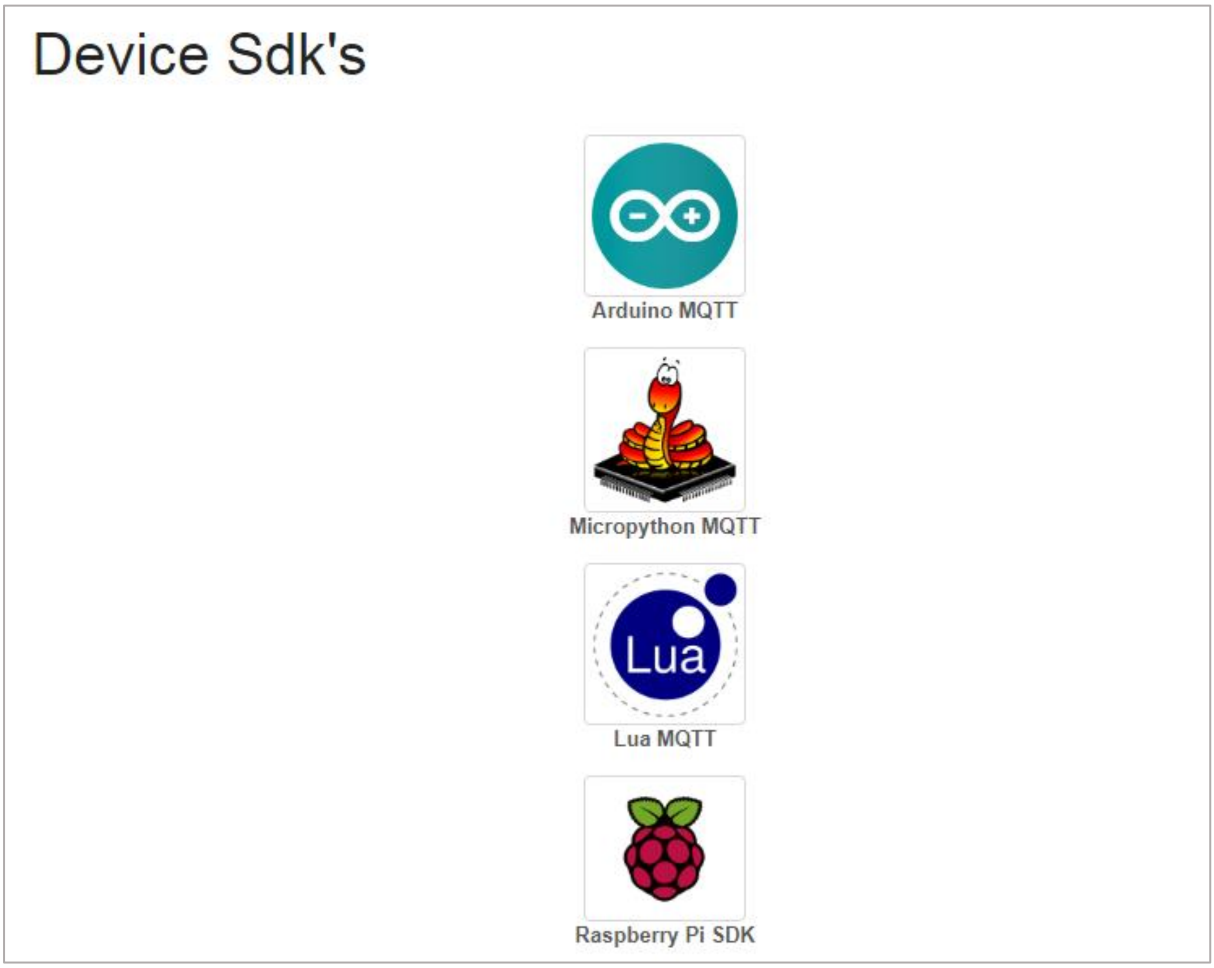
Note: To remove an added action simply click on the “**X**” mark

- Press “**Save**” to save the entire scene created.
- Press “**Execute**” to manually run the scene and check if the action responses take place as expected.
- Press “**Delete**” to delete the created scene.

3.1.5 Device SDK's

In this option XPAND has included SDK's to be utilized by the developers. The developers can use these SDK's to head-start with the process of onboarding their devices. At this the developers can download sample codes that include the process to connect devices to the system backend. Sample codes for each platform are available under this section. Moreover, the codes include comments for the ease of customizing the code as per developer requirements.

On clicking the "Device SDK's" option of the portal menu the developer navigates to the below screen.



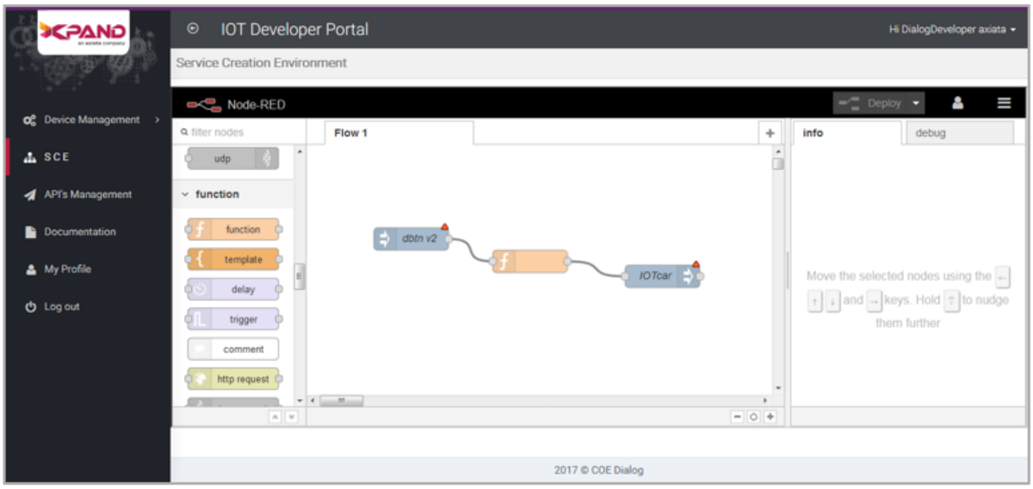
Device Sdk's

Figure 37 - Device SDK's screen

To download a SDK, simply click on the particular platform icon.

3.2 Service Creation Environment (SCE)

The SCE is completely an IoT testing platform. Through the SCE the developer gets the chance to test complex IoT operations. At this the user can create complex scenes using Node-RED functions.



Move the selected nodes using the ← ↑ ↓ and → keys. Hold ⇧ to nudge them further

Figure 38 - SCE page

The developer can create a complex scene by using Node-RED and XPAND nodes available in the left panel and creating custom flows of the IoT scenario within the workspace. Steps to be followed to create scenes with nodes are explained below;

- Simply drag and drop the nodes you need to the workspace.
- Double click on the node to specify the node information.
- Connect the nodes as a flow diagram.

To create scenes, in addition to the Node-RED nodes there are 4 nodes developed by XPAND. The below diagram shows those 4 nodes.

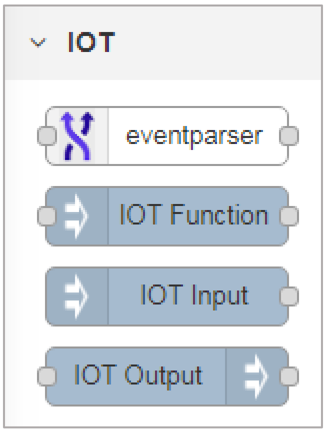


Figure 39 - XPAND SCE Nodes

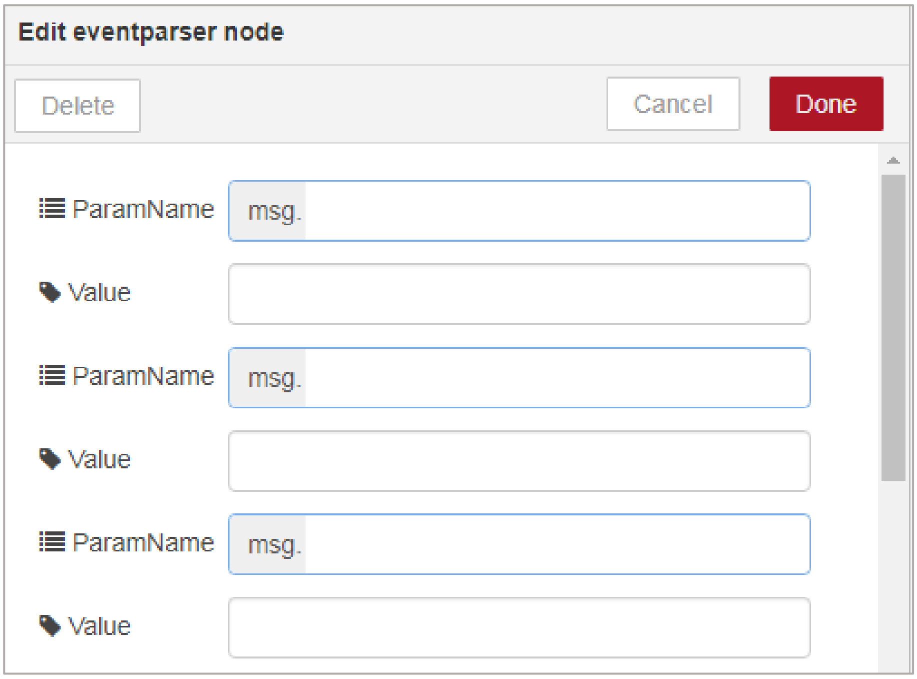
The above four nodes and their implementation are explained below,

1. **Event Parser –** Provides a node for parsing IoT event parameters. Event parser reads the parameters out from the IoT core message so the parameters can be evaluated later in other stages.

   IOT event parameter Masks

   - %iot.deviceId% - used to retrive iot device id of the event
   - %iot.eventName% - used to retrive event name. i.e. switch status change
   - %iot.stateName% - used to retrive event state. i.e. switch on/off
   - %iot.[event_parameter_name]% - Other parameter values if any. i.e. tempurature value of temperature change event

   On double clicking the "Event Parser" node the below screen appears



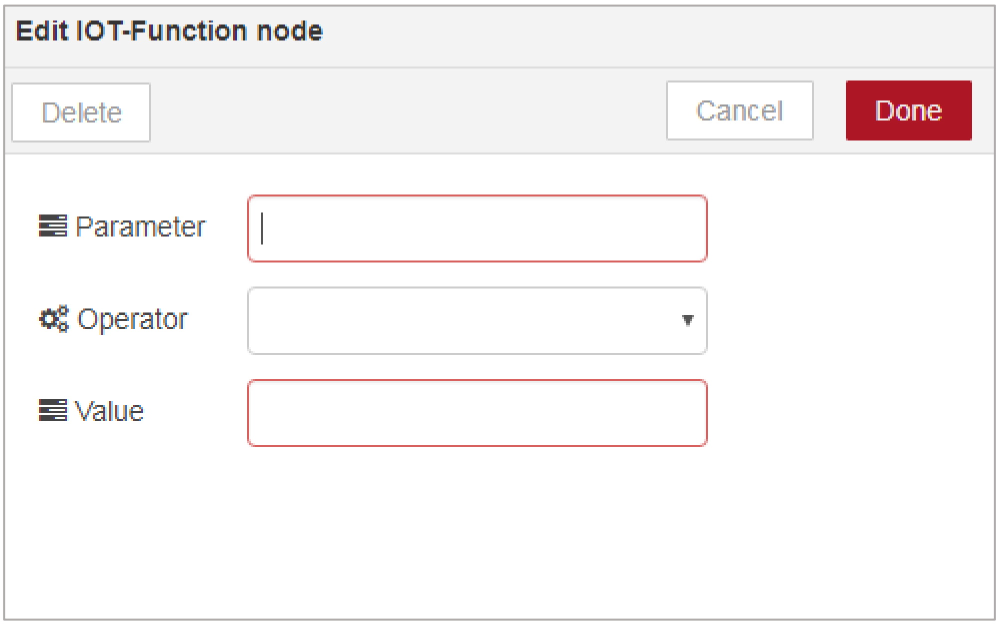
Edit eventparser node

Figure 40 - Event Parser node

   - **ParamName –** Specify the parameter name.
   - **Value –** Specify the parameter value.
   - Click "**Done"** button after adding the information.
   - Click "**Delete"** button to delete the node.

2. **IOT Function** – This is the node which basically controls the event parameter values which are being pushed from the input node, where the user will be able to change the condition according to the appropriate behavior the user expects from the output node.

On double clicking the IOT function node the below screen appears



Edit IOT-Function node

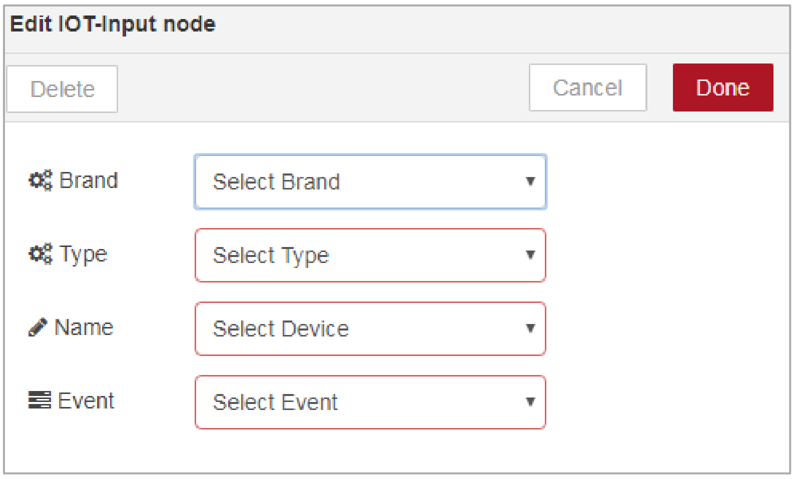
Parameter
Operator
Value

Figure 41 - IoT Function node

- **Parameter –**
- **Operator –**
- **Value –**
- Click "**Done**" button after adding the information.
- Click "**Delete**" button to delete the node.

3. **IOT Input –** This is the node which lists down all the event devices and the events for each event device. So, a user will be able to set the node to a particular device brand and the event for it, and then will be able to manipulate it using the function node or could directly send it to the output node.

   On double clicking the IoT Input node the below screen appears.



Edit IOT-Input node

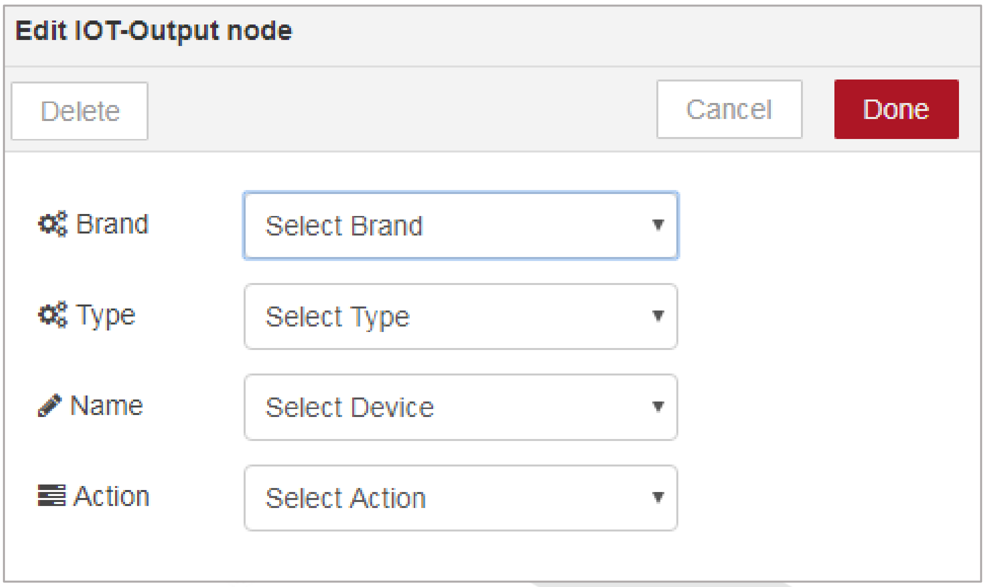
Brand: Select Brand
Type: Select Type
Name: Select Device
Event: Select Event

Figure 42 - IoT Input node

- **Brand –** Select the input device brand from the dropdown.
- **Type –** Select the device type from the dropdown.
- **Name –** Select the input device name from the dropdown.
- **Event –** Select the event from the dropdown.

4. **IOT Output –** This is the node which lists down all the action devices and the actions for each action device. So, the user will be able to set the node to a particular action brand and then an action for it. Hence the user can make sure that the output node gets triggered if the function node condition is being met. For example, if there's a temperature change, an alarm could be raised.

   On double clicking the IoT Output node the below screen appears.



Edit IOT-Output node

Brand: Select Brand

Type: Select Type

Name: Select Device

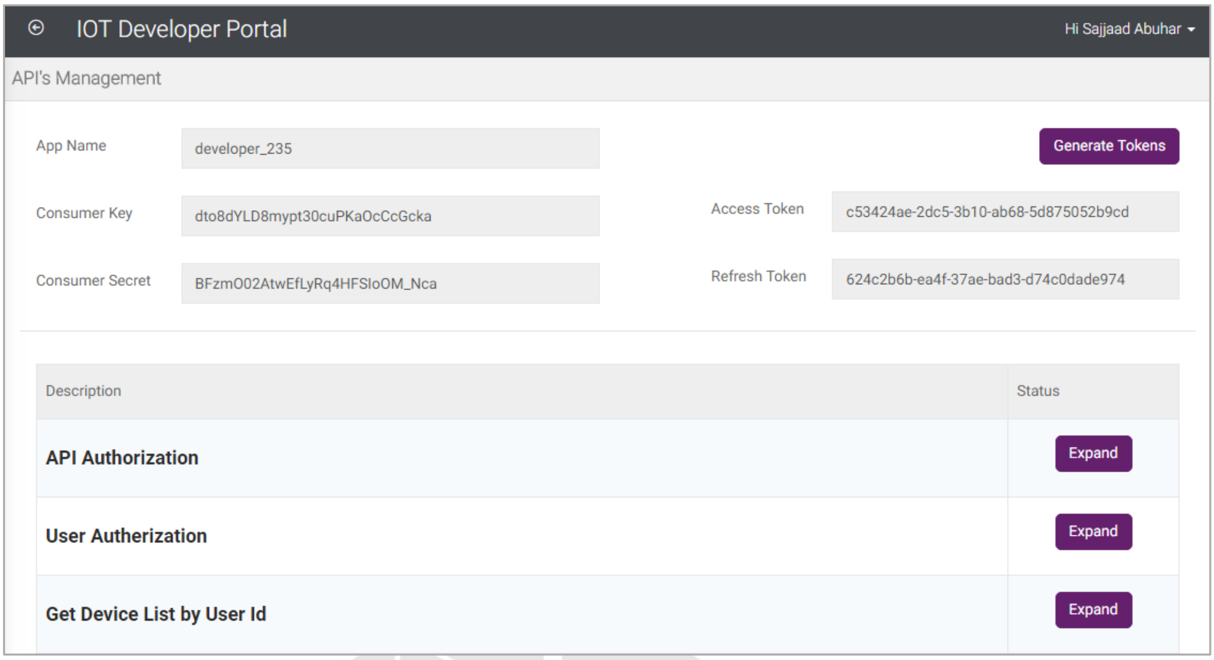
Action: Select Action

Figure 43 - IoT Output node

- **Brand –** Select the input device brand from the dropdown.
- **Type –** Select the device type from the dropdown.
- **Name –** Select the input device name from the dropdown.
- **Action –** Select the action from the dropdown.

3.3 API's Management

This function exposes the user to the Application Programming Interfaces (APIs) provided by XPAND IoT. These available API's can be referred and utilized by the developers to create their own custom front-end applications. On selecting the "API's Management" option in the menu, the user is able to view the API page shown below.



App Name: developer_235

Consumer Key: dto8dYLD8mypt30cuPKaOcCcGcka

Consumer Secret: BFzmO02AtwEfLyRq4HFSIoOM_Nca

Access Token: c53424ae-2dc5-3b10-ab68-5d875052b9cd

Refresh Token: 624c2b6b-ea4f-37ae-bad3-d74c0dade974

Description	Status
API Authorization	Expand
User Autherization	Expand
Get Device List by User Id	Expand

Figure 44 - API page

If a user needs to execute a particular API, the user should first subscribe to it. Afterwards the subscription request is forwarded to the admin for approval. The user gets to execute the API after receiving the approval.

For further information on API management please refer the "XPAND IoT API Document".

3.4 Logout

A developer can logout from the portal in two ways

1. The user can select the “**Logout**” option from the main menu.

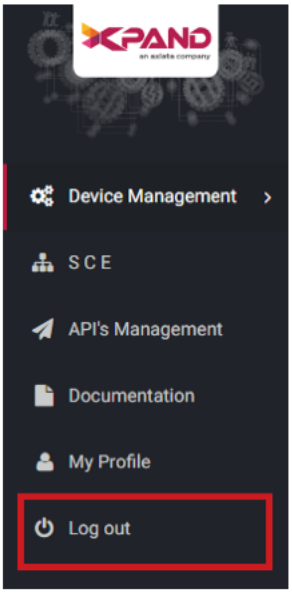


Figure 45 - Logout function 1

2. The other way to logout is;
   - Click the user’s name shown in the top right corner of the portal page. Upon clicking the name a dropdown appears as shown below.

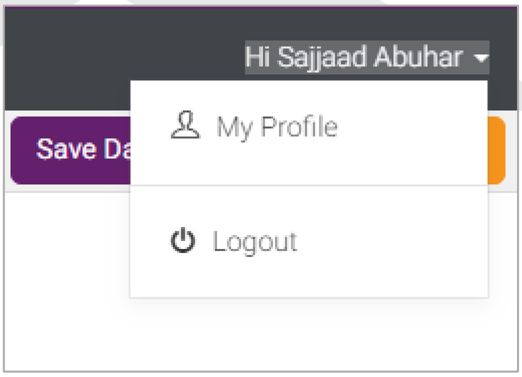


Figure 46 - Logout function 2

   - Click the “**Logout**” option to logout of the portal account.

After logging out successfully a user navigates back to the portal login page.