Consume REST API created in API Connect 5.0 using Mobile First

| Miracle API-M TEAM

Pavan Nadipelli
DataPower/API-M Developer
Miracle Software Systems, Inc.

03/15/2016

Goal

In this lab user can get access to Enterprise REST Services (APIs) through IBM API Connect's Developer Portal and then consume those services within a Hybrid Mobile App.

Pre-Requisites

The following installations will need to be completed for this lab to be run successfully,

- JDK
- Eclipse Luna with IBM Mobile First Plugin
- Browser for running mobile application on simulator
- IBM Mobile First Server should turn ON
- Jquery Mobile Library
- IBM API Management Developer Portal
- REST API

Download Links

1. JQuery Library:

https://jquery.com/download/

2. JQuery Mobile:

http://blog.jquerymobile.com/2013/07/19/announcing-jquery-mobile-1-3-2/

3. JDK:

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

4. Eclipse Luna:

http://www.eclipse.org/downloads/packages/release/Luna/SR2

Technology Involved

- IBM Mobile First Platform
- JQuery Mobile
- IBM API Connect beta
- REST APIs

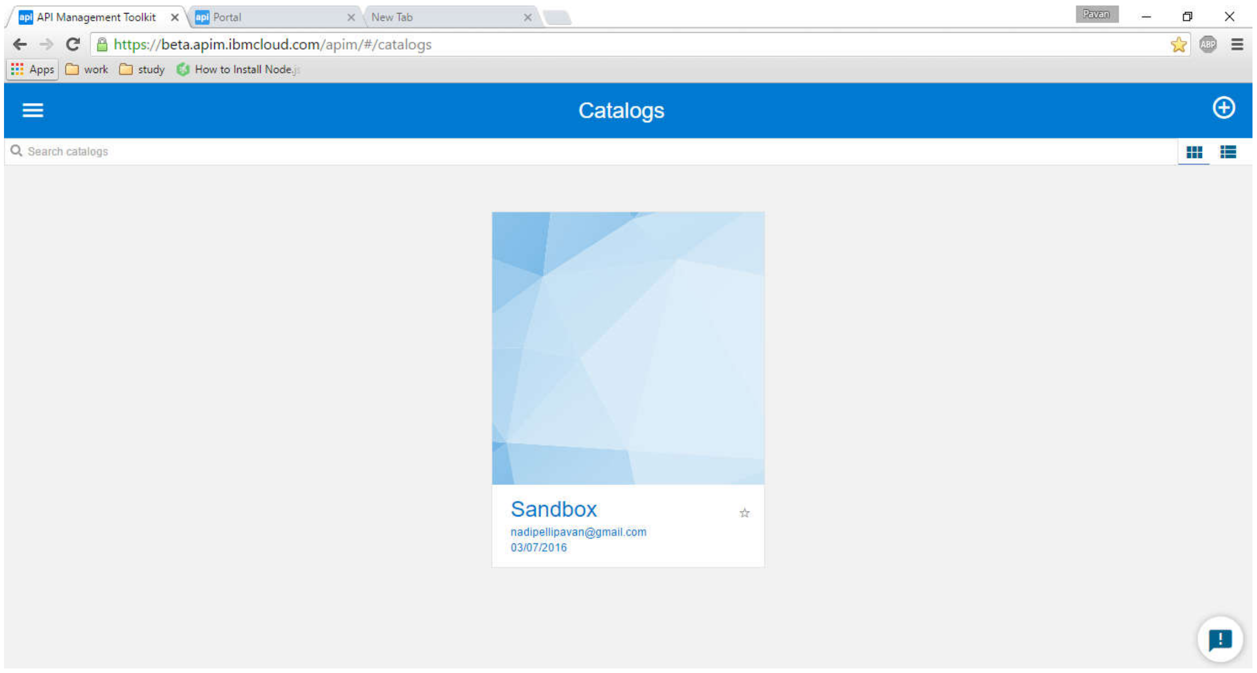
Lab Steps

So, let us get started with the lab!

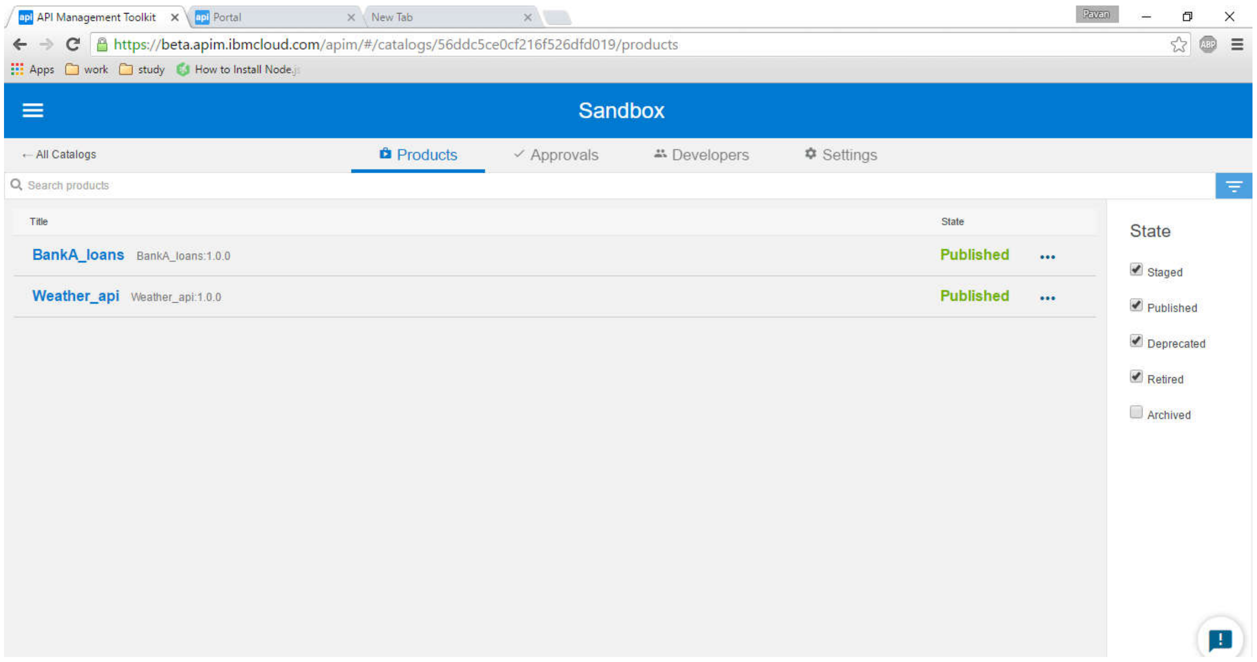
#1 | Getting REST API from APIM Beta Developer Portal

For getting the Client ID,
We need to Login into IBM API connect beta account. Use this link https://beta.apim.ibmcloud.com.

After logging in you can see below page

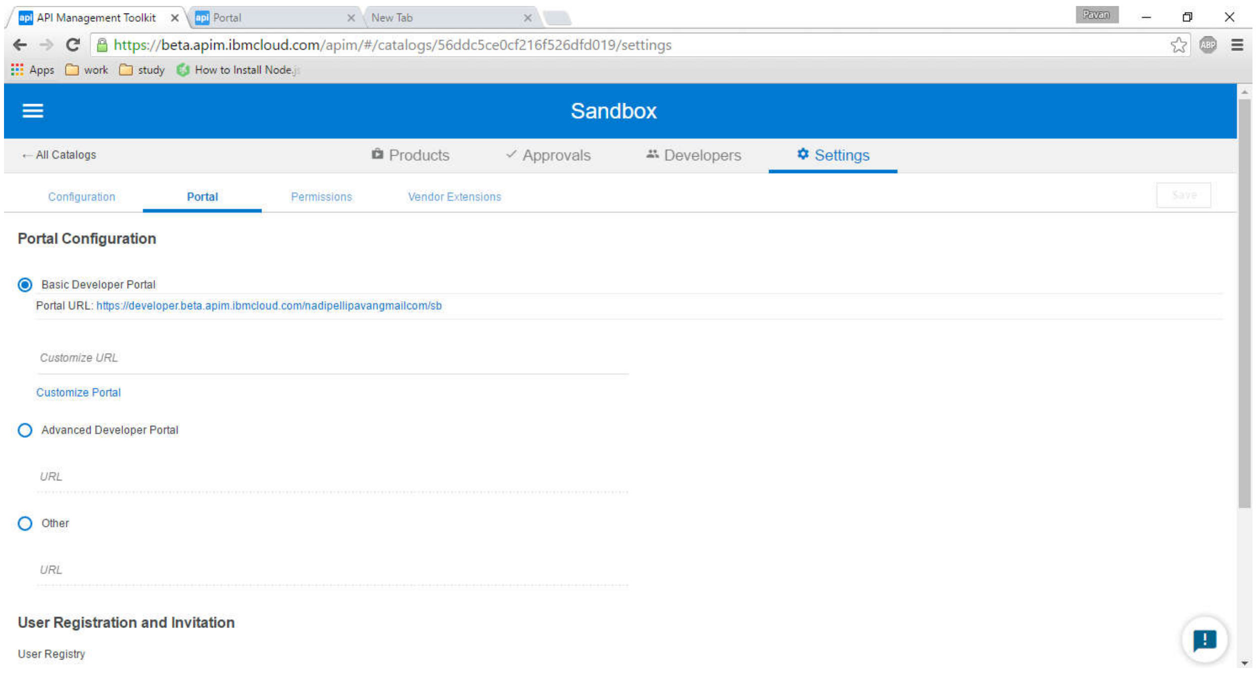


Select Sandbox Catalog (or) any other catalog you have published your API.



Title		State
BankA_loans	BankA_loans:1.0.0	Published
Weather_api	Weather_api:1.0.0	Published

Go to settings. And select Portal



Portal Configuration

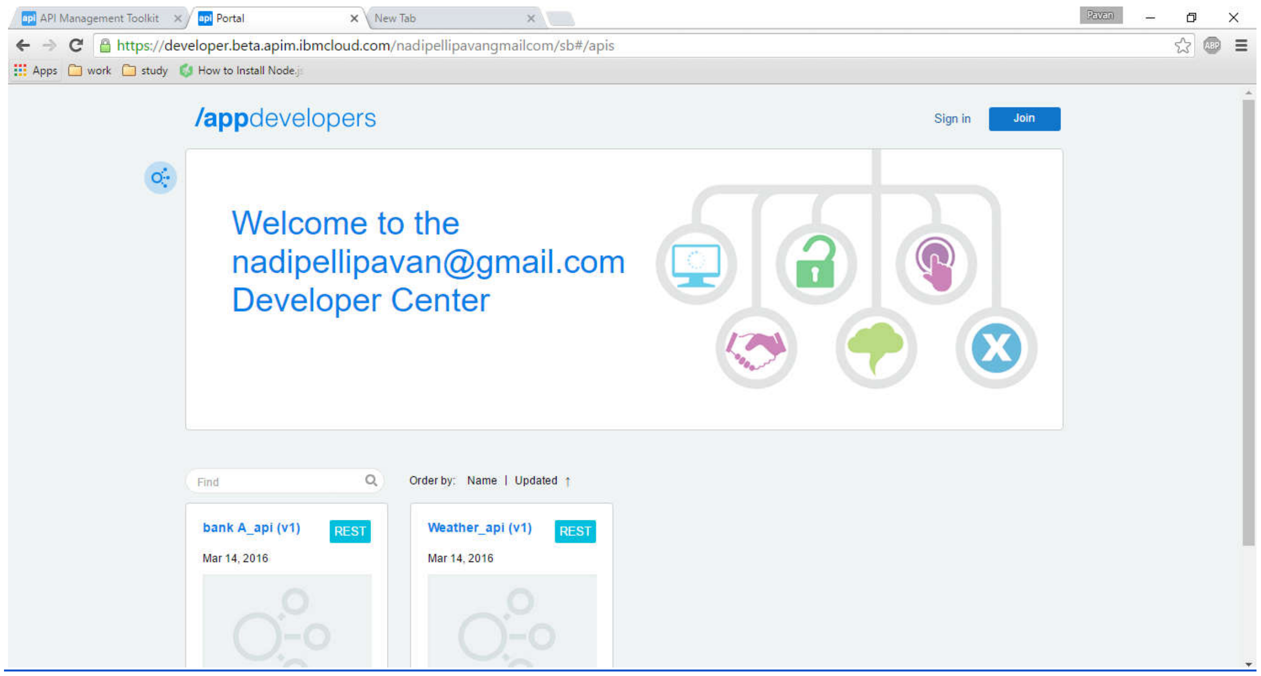
● Basic Developer Portal
Portal URL: https://developer.beta.apim.ibmcloud.com/nadipellipavangmailcom/sb
Customize URL
Customize Portal

○ Advanced Developer Portal
URL

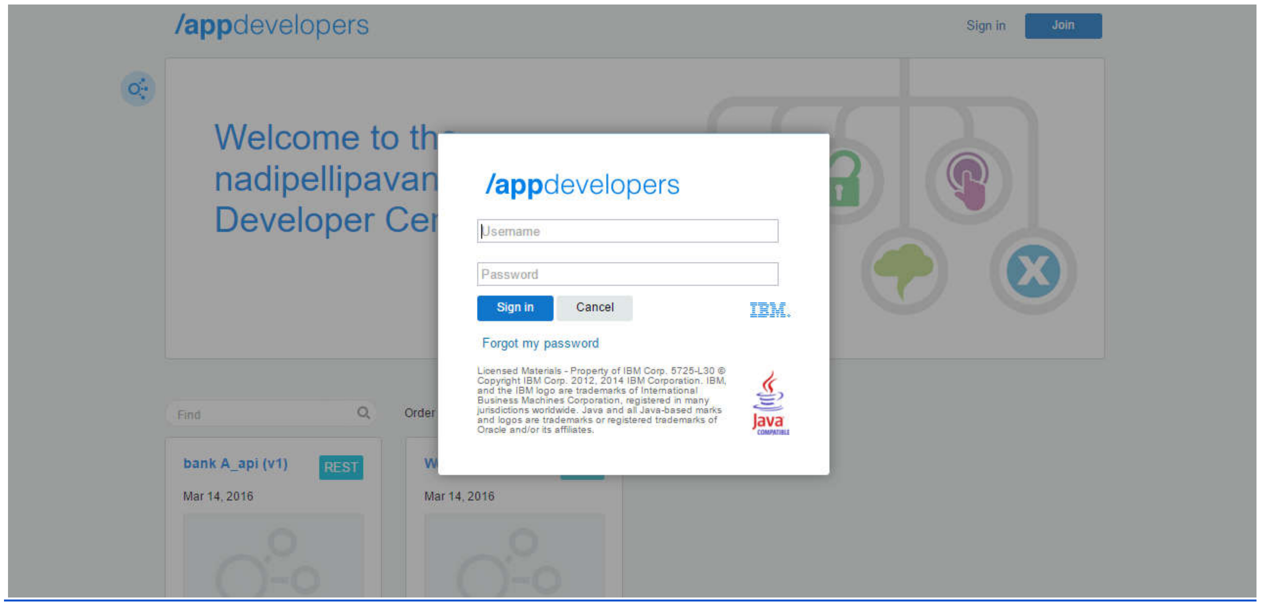
○ Other
URL

User Registration and Invitation
User Registry

Here you can find the developer portal link. After clicking the URL you will be redirecting to API Developer Portal.

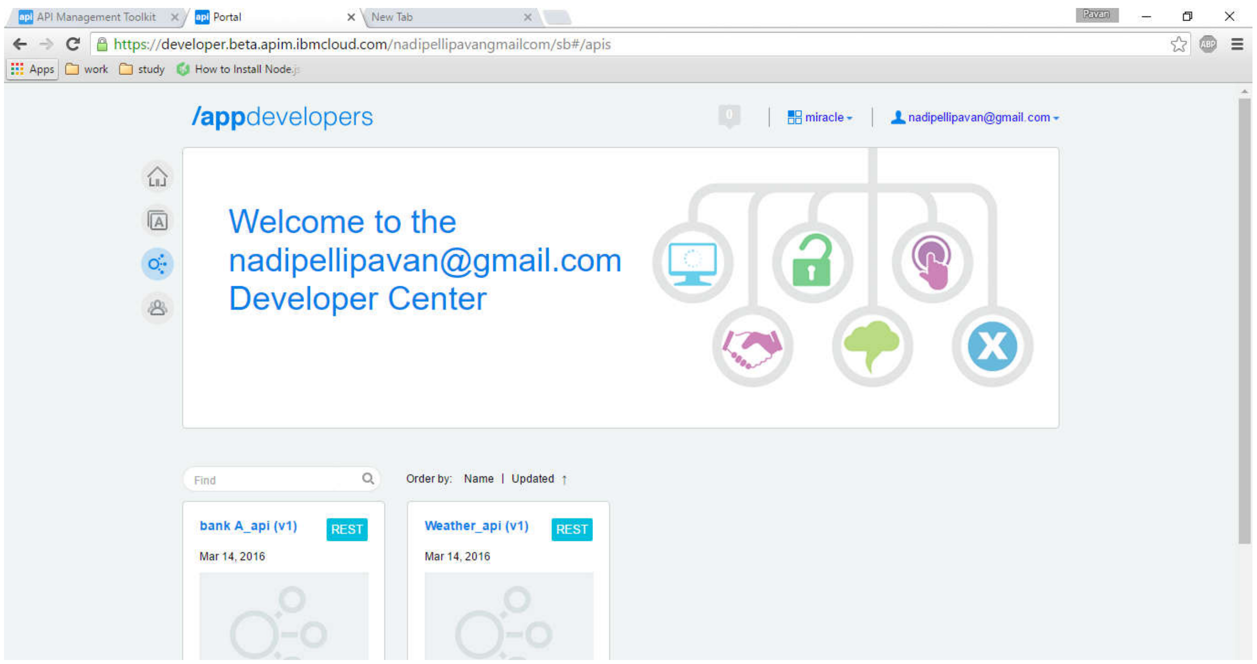


Then click on **Sign in** button to navigate to login page as shown in the figure.

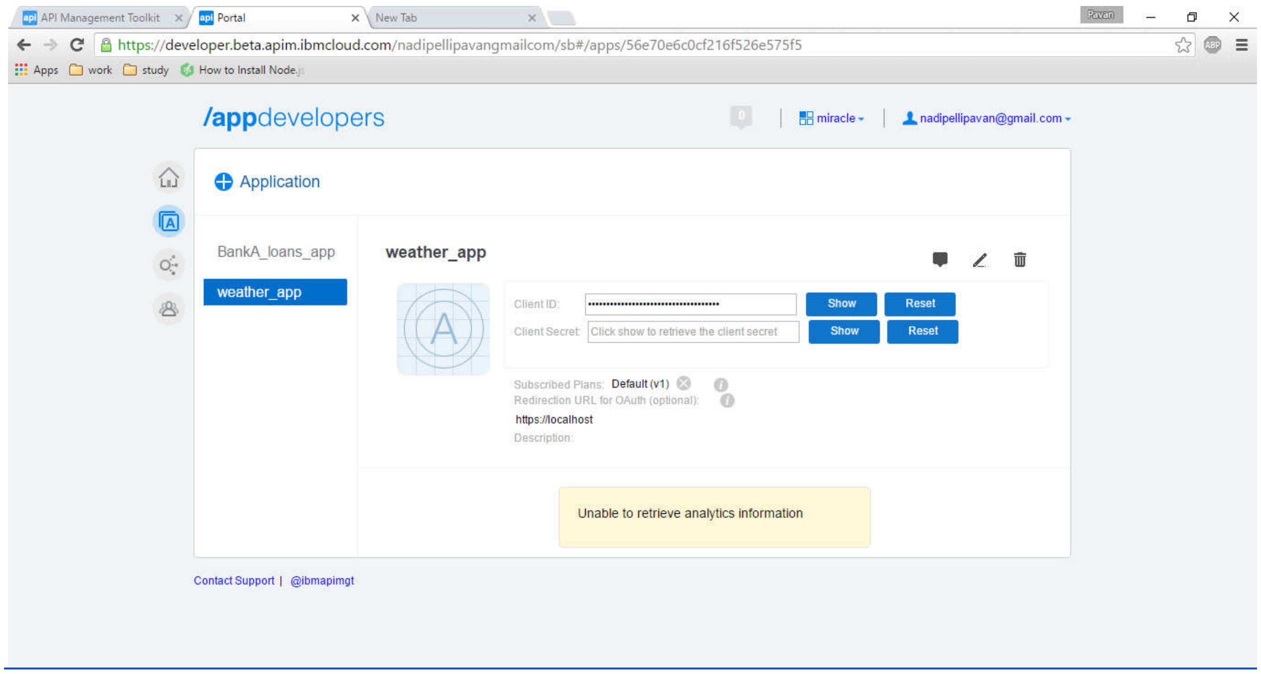

/appdevelopers
Username
Password
Sign in
Cancel
IBM.
Forgot my password
Licensed Materials - Property of IBM Corp. 5725-L30 © Copyright IBM Corp. 2012, 2014 IBM Corporation. IBM, and the IBM logo are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Java and all Java-based marks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Sign in to the IBM API Management by entering your credentials and click on Sign In button.

Now you can see the developer portal home page.



Click on Applications Icon in the Navigation Pane on left side. And select the required application.

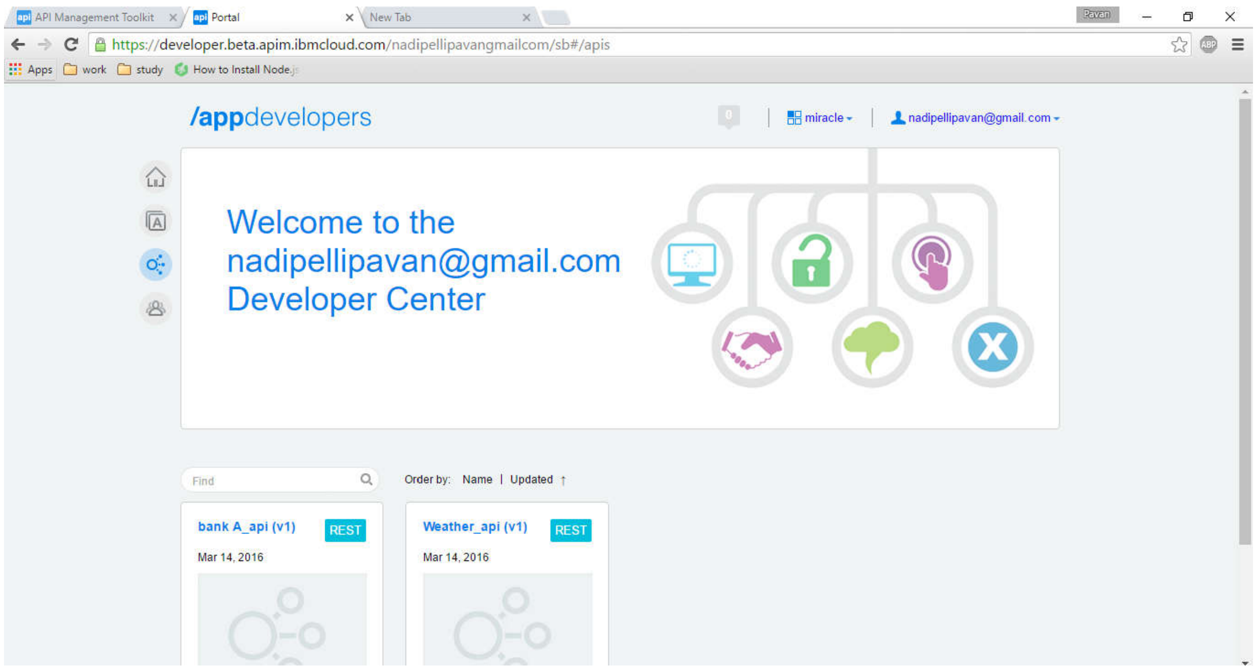

https://developer.beta.apim.ibmcloud.com/nadipellipavangmailcom/sb#/apps/56e70e6c0cf216f526e575f5

weather_app

Client ID: •••••••••••••••••••••••••••••••••••
Client Secret: Click show to retrieve the client secret
Subscribed Plans: Default (v1)
Redirection URL for OAuth (optional):
https://localhost
Description:

Unable to retrieve analytics information

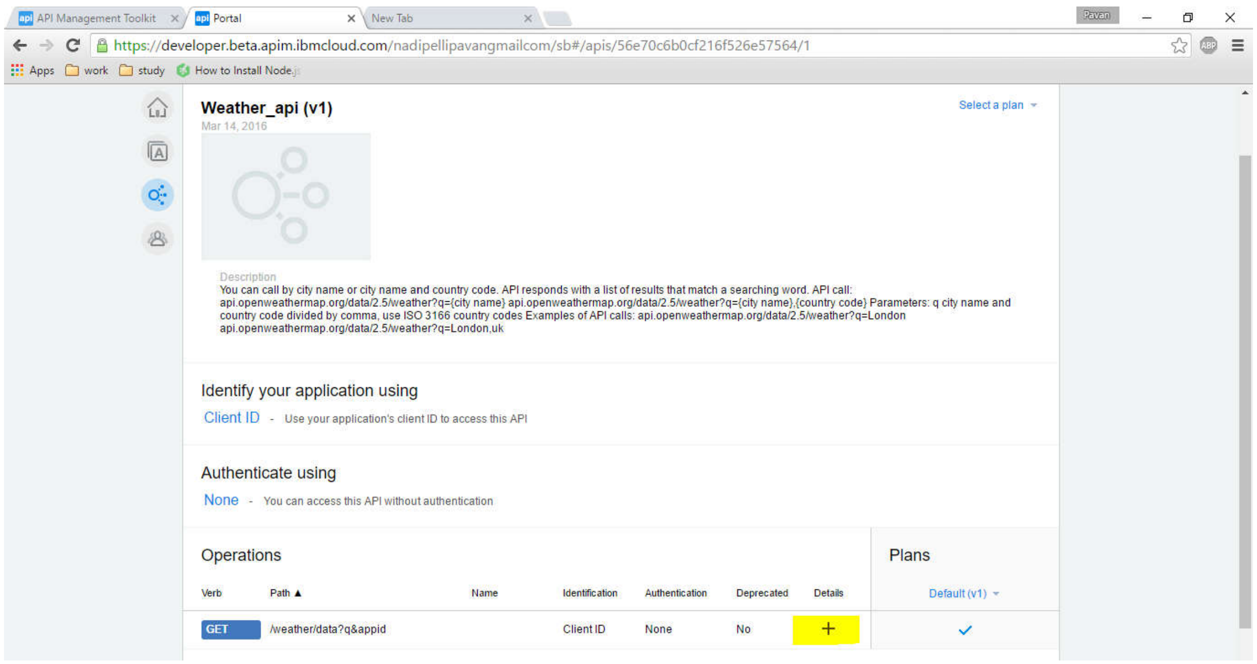
Click on show button and copy the client ID.
Go back to APIs page


https://developer.beta.apim.ibmcloud.com/nadipellipavangmailcom/sb#/apis

Welcome to the nadipellipavan@gmail.com Developer Center

bank A_api (v1) REST Mar 14, 2016
Weather_api (v1) REST Mar 14, 2016

Click on the **Weather_api (v1)**


API Management Toolkit
Portal
New Tab
Pavan
https://developer.beta.apim.ibmcloud.com/nadipellipavangmailcom/sb#/apis/56e70c6b0cf216f526e57564/1
Apps
work
study
How to Install Node.js
Select a plan

Weather_api (v1)
Mar 14, 2016

Description
You can call by city name or city name and country code. API responds with a list of results that match a searching word. API call: api.openweathermap.org/data/2.5/weather?q={city name} api.openweathermap.org/data/2.5/weather?q={city name},{country code} Parameters: q city name and country code divided by comma, use ISO 3166 country codes Examples of API calls: api.openweathermap.org/data/2.5/weather?q=London api.openweathermap.org/data/2.5/weather?q=London,uk

Identify your application using
Client ID - Use your application's client ID to access this API

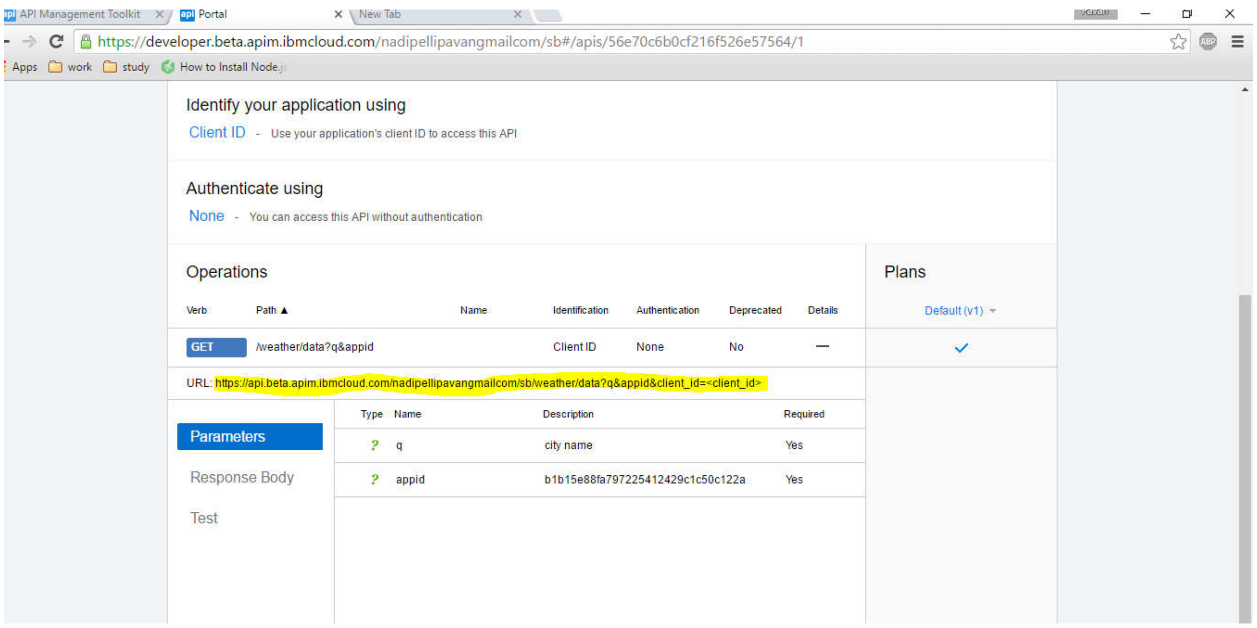
Authenticate using
None - You can access this API without authentication

Operations

Verb	Path ▲	Name	Identification	Authentication	Deprecated	Details
GET	/weather/data?q&appid		Client ID	None	No	+

Plans
Default (v1) ✓

Click on "+", now you can see the API Proxy URL. (Highlighted in yellow)


API Management Toolkit
Portal
New Tab
https://developer.beta.apim.ibmcloud.com/nadipellipavangmailcom/sb#/apis/56e70c6b0cf216f526e57564/1
Apps
work
study
How to Install Node.js
Parameters
Response Body
Test

Identify your application using
Client ID - Use your application's client ID to access this API

Authenticate using
None - You can access this API without authentication

Operations

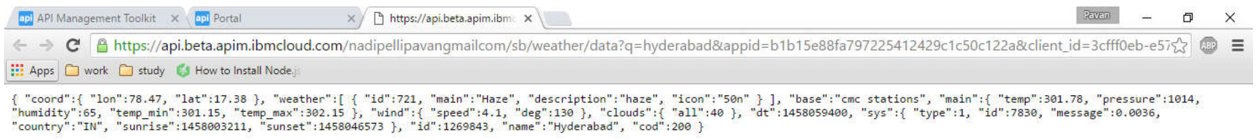
Verb	Path ▲	Name	Identification	Authentication	Deprecated	Details
GET	/weather/data?q&appid		Client ID	None	No	—

Plans
Default (v1) ✓

URL: https://api.beta.apim.ibmcloud.com/nadipellipavangmailcom/sb/weather/data?q&appid&client_id=<client_id>

Type	Name	Description	Required
?	q	city name	Yes
?	appid	b1b15e88fa797225412429c1c50c122a	Yes

Copy the link for the future purpose. Replace the <client_Id> of the URL with the client id generated in the previous step. Now you can test this entire URL in the browser as shown in the figure.

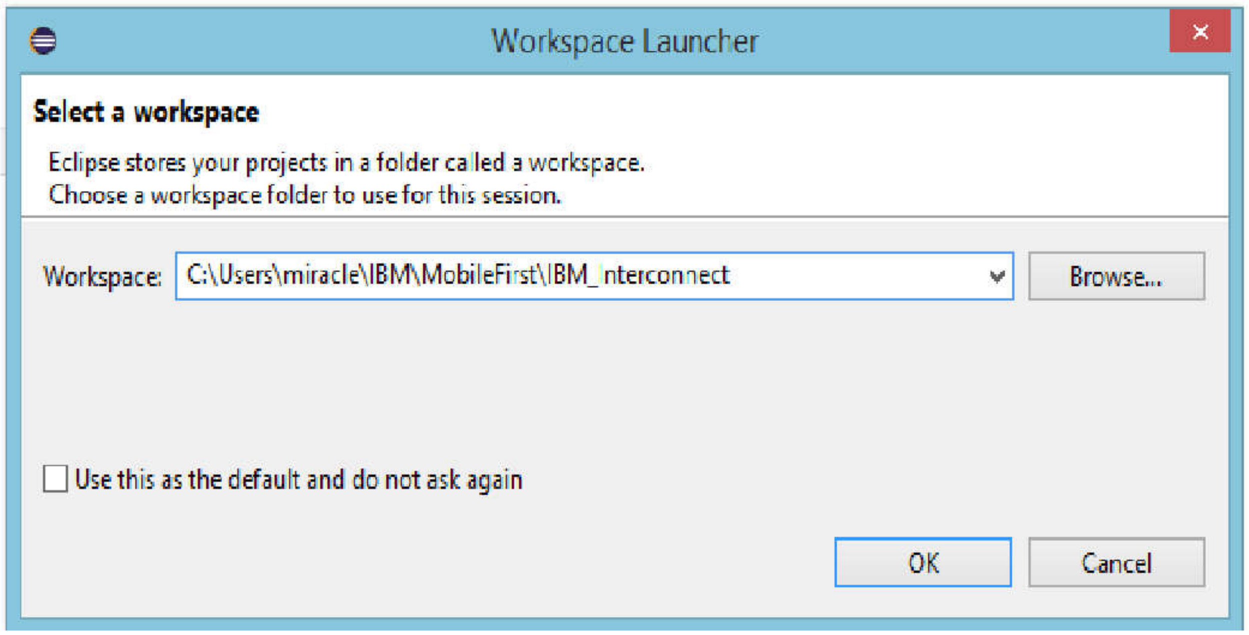


https://api.beta.apim.ibmcloud.com/nadipellipavangmailcom/sb/weather/data?q=hyderabad&appid=b1b15e88fa797225412429c1c50c122a&client_id=3cfff0eb-e57

```
{ "coord":{ "lon":78.47, "lat":17.38 }, "weather":[ { "id":721, "main":"Haze", "description":"haze", "icon":"50n" } ], "base":"cmc stations", "main":{ "temp":301.78, "pressure":1014,
"humidity":65, "temp_min":301.15, "temp_max":302.15 }, "wind":{ "speed":4.1, "deg":130 }, "clouds":{ "all":40 }, "dt":1458059400, "sys":{ "type":1, "id":7830, "message":0.0036,
"country":"IN", "sunrise":1458003211, "sunset":1458046573 }, "id":1269843, "name":"Hyderabad", "cod":200 }
```

#2 | Create project in IBM Mobile First

Open Eclipse IDE and select appropriate workspace as shown in the figure. Here give your workspace path and then click on OK**.**



Select a workspace

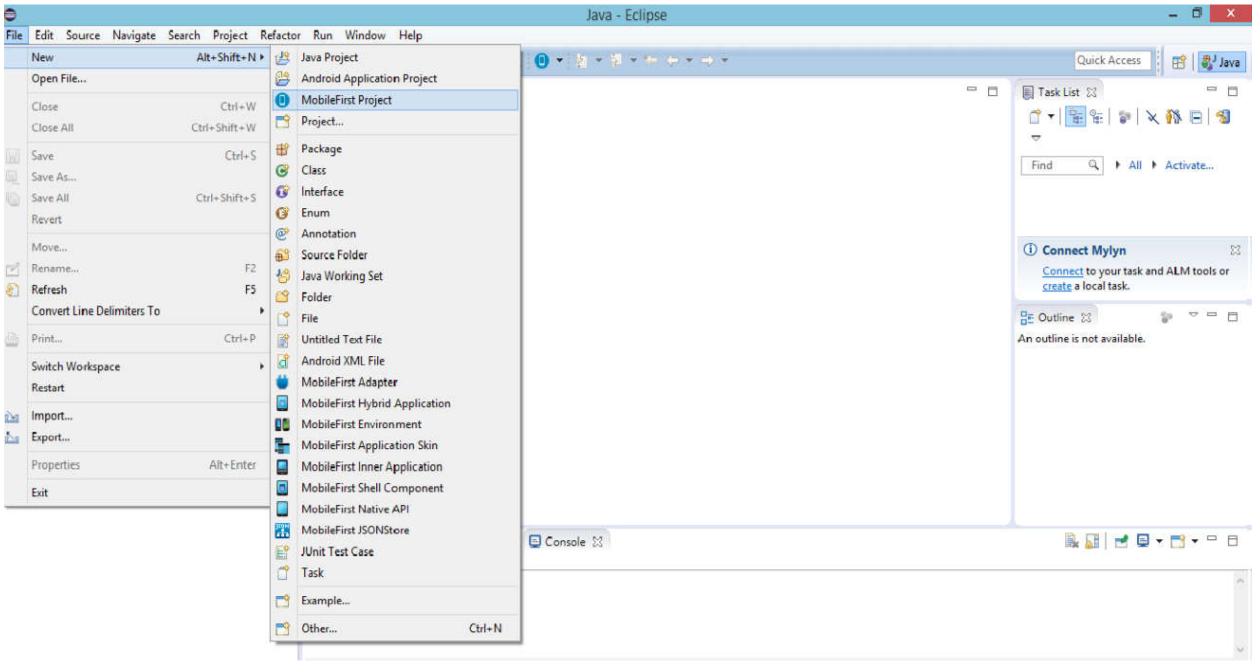
Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Users\miracle\IBM\MobileFirst\IBM_Interconnect

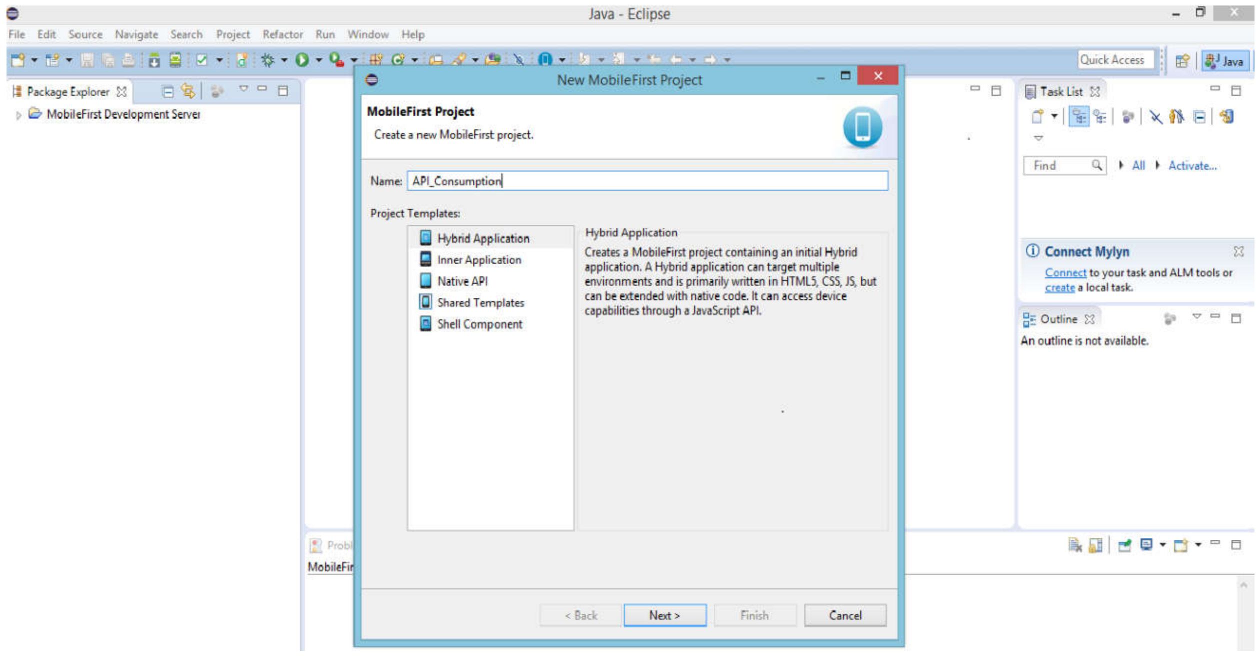
☐ Use this as the default and do not ask again

After clicking on OK, the welcome page gets displayed as shown in the figure.

Just close the welcome page and create a new MobileFirst Project by going to **File-> New->MobileFirst Project** as shown in the figure.



Give the appropriate project name in the Name field. Here I'm giving the project name as API_Consumption and click on next.



New MobileFirst Project

MobileFirst Project

Create a new MobileFirst project.

Name: API_Consumption

Project Templates:

- Hybrid Application
- Inner Application
- Native API
- Shared Templates
- Shell Component

Hybrid Application

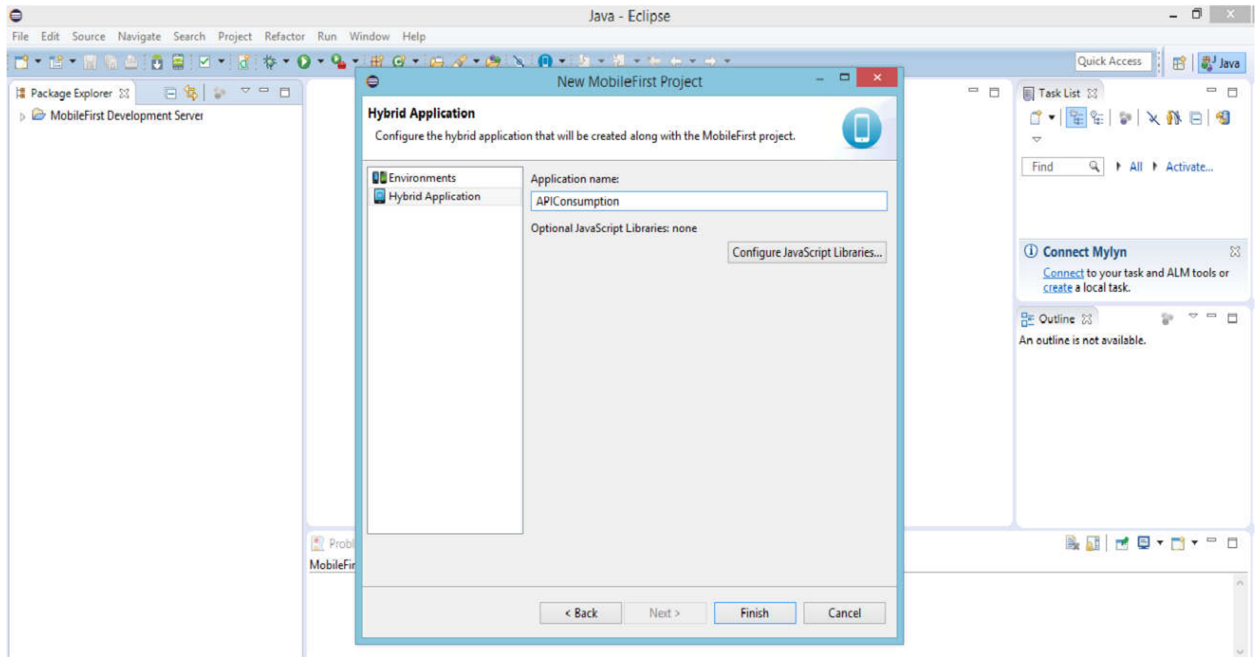
Creates a MobileFirst project containing an initial Hybrid application. A Hybrid application can target multiple environments and is primarily written in HTML5, CSS, JS, but can be extended with native code. It can access device capabilities through a JavaScript API.

Connect Mylyn

Connect to your task and ALM tools or create a local task.

An outline is not available.

Give the application name in the **Name** field, here I'm giving API Consumption. And then click on Configure JavaScript Libraries as shown in the figure.



New MobileFirst Project

Hybrid Application

Configure the hybrid application that will be created along with the MobileFirst project.

Application name: APIConsumption

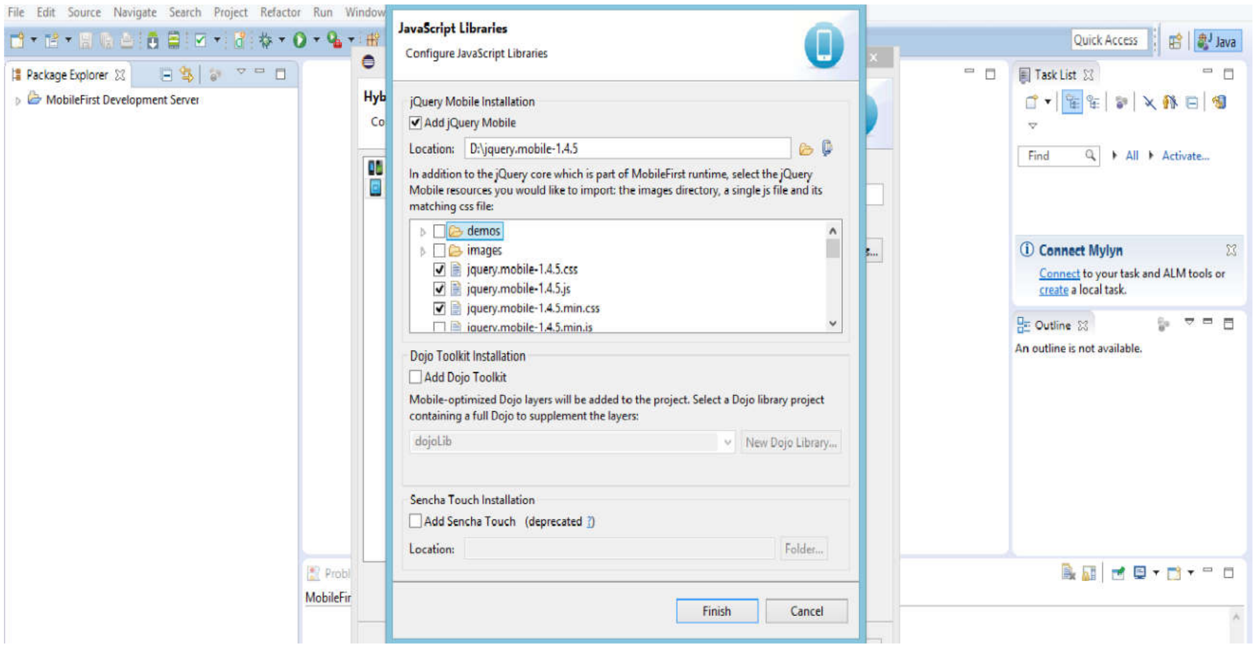
Optional JavaScript Libraries: none

Connect Mylyn

Connect to your task and ALM tools or create a local task.

An outline is not available.

Then the following window will open, as shown in the figure.



JavaScript Libraries

Configure JavaScript Libraries

jQuery Mobile Installation

[✓] Add jQuery Mobile

Location: D:\jquery.mobile-1.4.5

In addition to the jQuery core which is part of MobileFirst runtime, select the jQuery Mobile resources you would like to import: the images directory, a single js file and its matching css file:

- [ ] demos
- [ ] images
- [✓] jquery.mobile-1.4.5.css
- [✓] jquery.mobile-1.4.5.js
- [✓] jquery.mobile-1.4.5.min.css
- [ ] jquery.mobile-1.4.5.min.js

Dojo Toolkit Installation

[ ] Add Dojo Toolkit

Mobile-optimized Dojo layers will be added to the project. Select a Dojo library project containing a full Dojo to supplement the layers:

dojoLib — New Dojo Library...

Sencha Touch Installation

[ ] Add Sencha Touch (deprecated)

Location: — Folder...

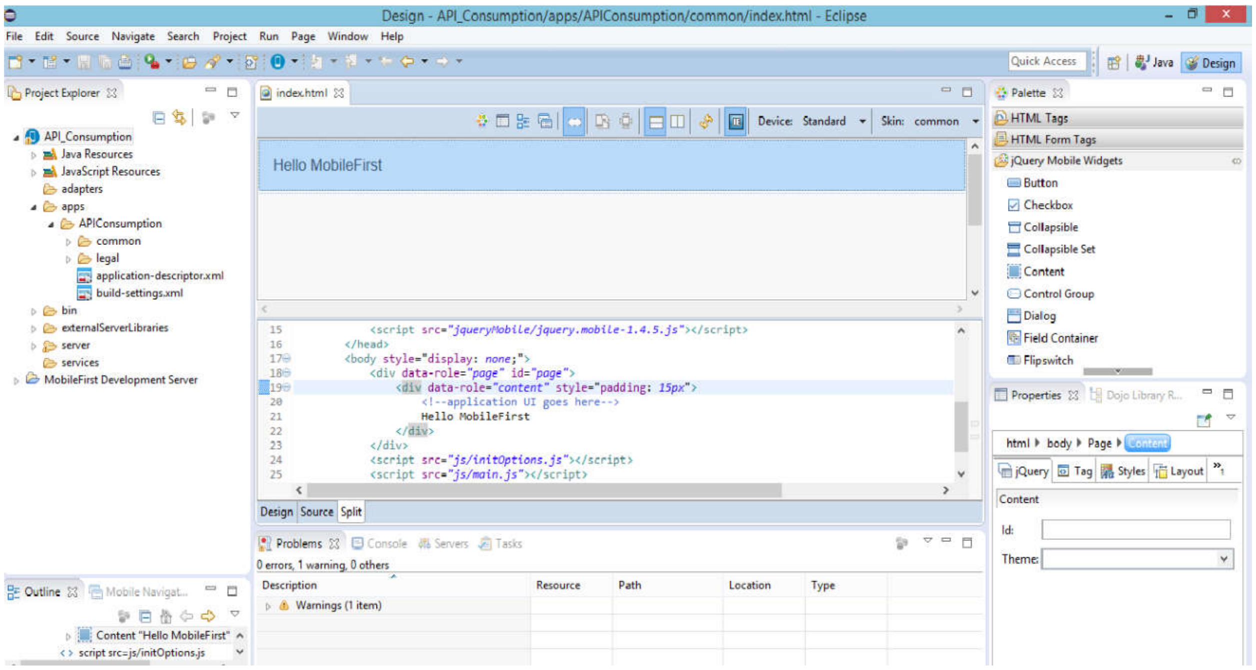
Connect Mylyn

Connect to your task and ALM tools or create a local task.

An outline is not available.

Now select any JavaScript library, here I'm using the JQuery Mobile library. Select the library from your local machine. And then select Jquery.mobile-1.4.5.js, Jquery.mobile-1.4.5.css and Jquery.mobile-1.4.5.min.css. Click on Finish.

After clicking on the Finish button, the project creation will get completed and the screen will appear as follows.



```
        <script src="jqueryMobile/jquery.mobile-1.4.5.js"></script>
    </head>
    <body style="display: none;">
        <div data-role="page" id="page">
            <div data-role="content" style="padding: 15px">
                <!--application UI goes here-->
                Hello MobileFirst
            </div>
        </div>
        <script src="js/initOptions.js"></script>
        <script src="js/main.js"></script>
```

Problems | Console | Servers | Tasks

0 errors, 1 warning, 0 others

Description	Resource	Path	Location	Type
Warnings (1 item)				

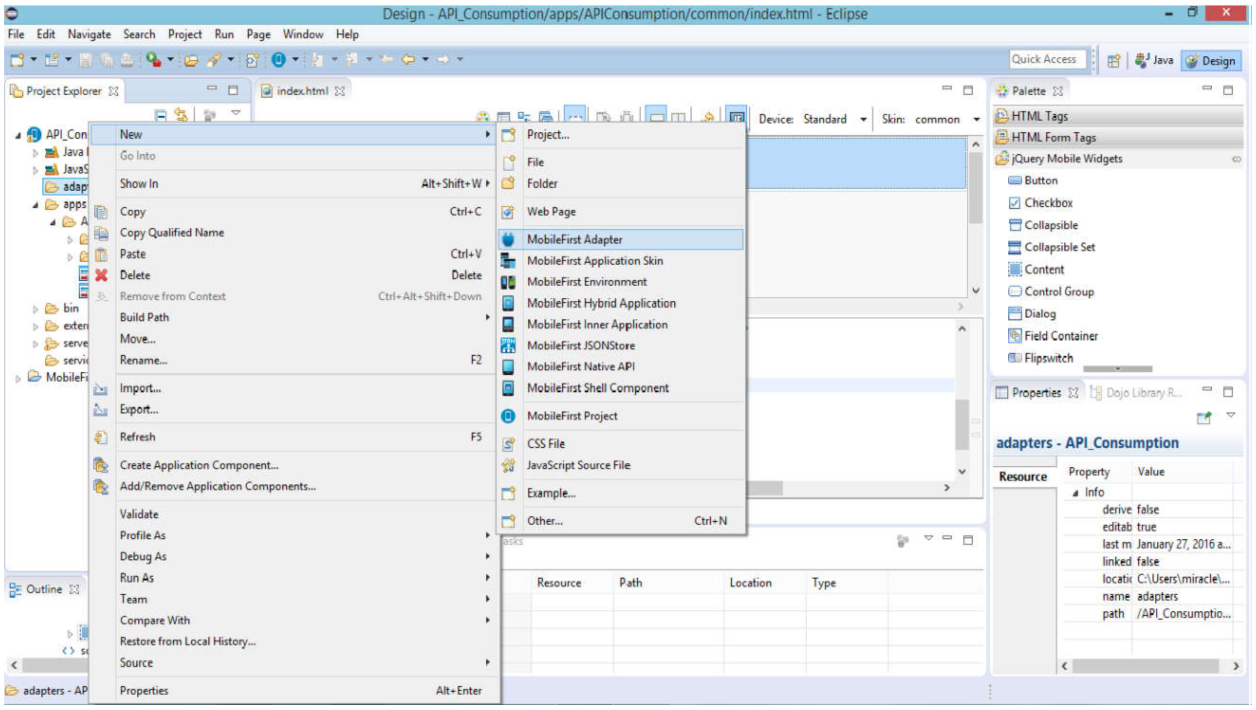
Outline | Mobile Navigat...

Content "Hello MobileFirst"

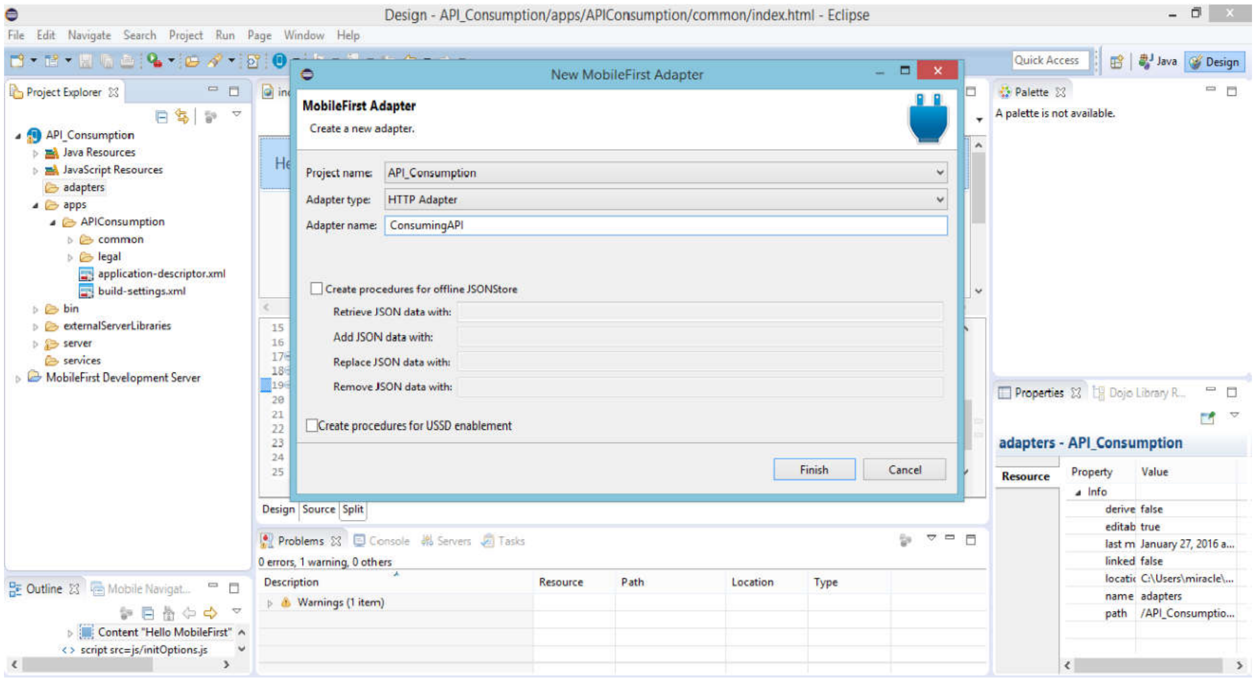
script src=js/initOptions.js

#3 | Create IBM Mobile First Adapter (HTTP)

Create an HTTP Adapter, for this simply right click on the project and select the new Mobile First Adapter as shown in the figure.



After selecting the Mobile First Adapter, the new window will open and select the particular adapter (HTTP Adapter) and give an appropriate name to it as shown in the figure.


New MobileFirst Adapter

MobileFirst Adapter

Create a new adapter.

Project name:	API_Consumption
Adapter type:	HTTP Adapter
Adapter name:	ConsumingAPI

☐ Create procedures for offline JSONStore

☐ Create procedures for USSD enablement

Here the adapter will have two different files,

1. Adapter.xml: Settings and metadata will store on Adapter.xml. To edit the xml file you must set the following things

a. The protocol to HTTP or HTTPS

b. The HTTP domain to the domain part of HTTP URL

c. The TCP Port

d. And, declare the procedure to be implemented.

2.JavaScript Implementation file: Here we will implement the declared procedures in the Adapter.xml file

#4 | Configuring and Implementation of an HTTP Adapter

Divide the REST URL into three parts (protocol, domain and path) and configured in the adapter.xml file. Here protocol and domain are useful in configuring the xml file. The remaining path is useful in the implementation part.

Adapter.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
   Licensed Materials - Property of IBM
   5725-I43 (C) Copyright IBM Corp. 2011, 2013. All Rights Reserved.
   US Government Users Restricted Rights - Use, duplication or
   disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-->
<wl:adapter name="ConsumingAPI"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:wl="http://www.ibm.com/mfp/integration"
          xmlns:http="http://www.ibm.com/mfp/integration/http">

          <displayName>ConsumingAPI</displayName>
          <description>ConsumingAPI</description>
          <connectivity>
                    <connectionPolicy xsi:type="http:HTTPConnectionPolicyType">
```

```
                        <protocol>https</protocol>
                        <domain>api.beta.apim.ibmcloud.com</domain>
                        <port>443</port>

        <connectionTimeoutInMilliseconds>30000</connectionTimeoutInMilliseconds>
                        <socketTimeoutInMilliseconds>30000</socketTimeoutInMilliseconds>

        <maxConcurrentConnectionsPerNode>50</maxConcurrentConnectionsPerNode>
                        <!-- Following properties used by adapter's key manager for choosing
specific certificate from key store
                        <sslCertificateAlias></sslCertificateAlias>
                        <sslCertificatePassword></sslCertificatePassword>
                        -->
                </connectionPolicy>
        </connectivity>

        <procedure name="getWeatherReport" connectAs="endUser"/>
</wl:adapter>
```

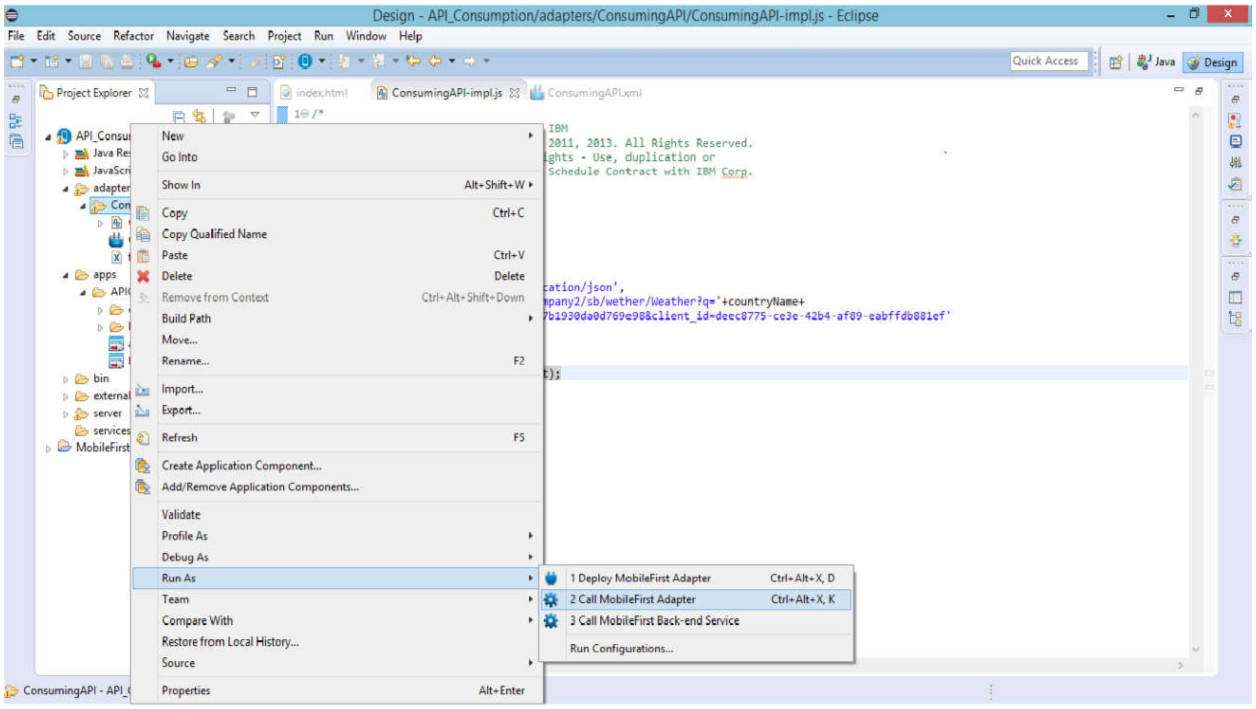
Adapter-impl.js

Here we need to implement the procedures, which are declared in the adapter configuration file. The following is the code snippet for implementing the procedure.

```
function getWeatherReport(country) {
    var countryName = country;

    var input = {
      method : 'get',
      returnedContentType : 'application/json',
      path :
'/nadipellipavangmailcom/sb/weather/data?q='+countryName+'&appid=b1b15e88fa79722541
2429c1c50c122a&client_id=3cfff0eb-e57d-4961-8fe1-519a27c15ff0'
  };

  return WL.Server.invokeHttp(input);

  }
```

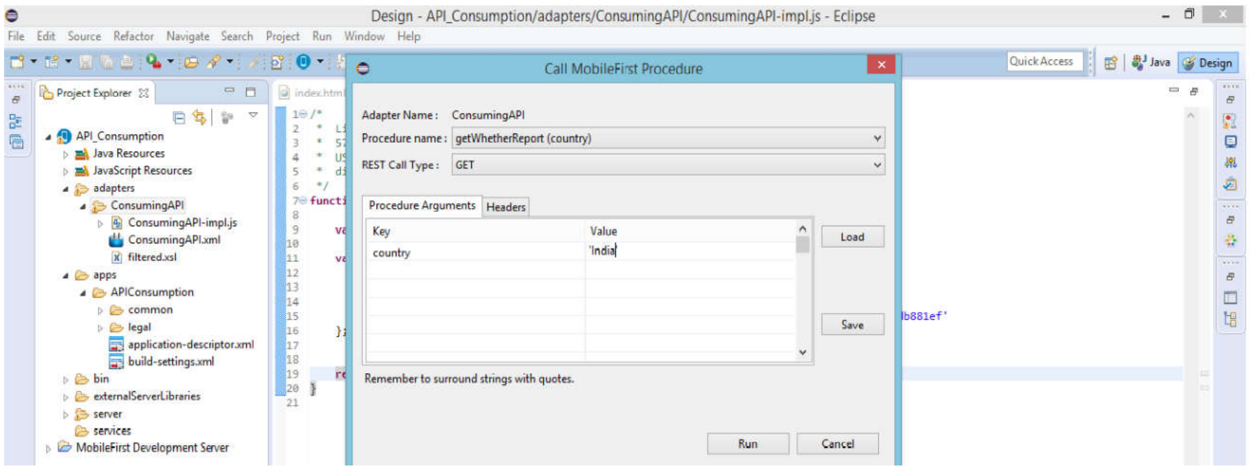
#5 | Deploying and testing the Adapter in IBM Mobile First Server

Before deploying and testing, we should start the Mobile First Server, It will take 2-3 min of time. Here, there is no need of deploying because, Mobile First adapter will automatically deploy after saving the adapter. For testing the adapter just right click on the Adapter->Run as -> Call Mobile First Adapter as shown in the figure.



```
IBM
2011, 2013. All Rights Reserved.
ights - Use, duplication or
Schedule Contract with IBM Corp.

cation/json',
mpany2/sb/wether/Weather?q='+countryName+
7b1930da0d769e98&client_id=deec8775-ce3e-42b4-af89-eabffdb881ef'

t);
```

Then the new window will open and ask for the Input. Here we have to select procedure name and REST Call Type. In my case the procedure name is getWeatherReport and type is GET. And then we need to give the input value and click on the Run button.



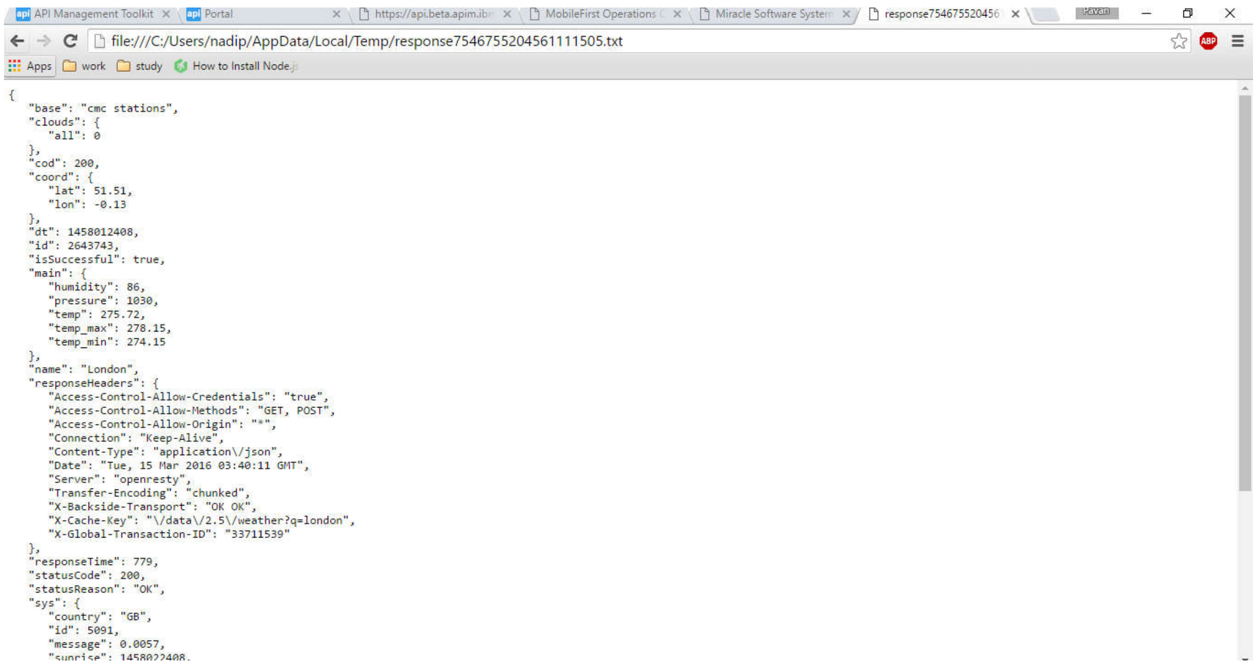
Call MobileFirst Procedure

Adapter Name: ConsumingAPI
Procedure name: getWhetherReport (country)
REST Call Type: GET

Key	Value
country	'India'

Remember to surround strings with quotes.

As my default browser is Google Chrome, the response of an adapter will open in this browser. The response here is in the form of JSON and it looks like follows.



file:///C:/Users/nadip/AppData/Local/Temp/response7546755204561111505.txt

```
{
   "base": "cmc stations",
   "clouds": {
      "all": 0
   },
   "cod": 200,
   "coord": {
      "lat": 51.51,
      "lon": -0.13
   },
   "dt": 1458012408,
   "id": 2643743,
   "isSuccessful": true,
   "main": {
      "humidity": 86,
      "pressure": 1030,
      "temp": 275.72,
      "temp_max": 278.15,
      "temp_min": 274.15
   },
   "name": "London",
   "responseHeaders": {
      "Access-Control-Allow-Credentials": "true",
      "Access-Control-Allow-Methods": "GET, POST",
      "Access-Control-Allow-Origin": "*",
      "Connection": "Keep-Alive",
      "Content-Type": "application\/json",
      "Date": "Tue, 15 Mar 2016 03:40:11 GMT",
      "Server": "openresty",
      "Transfer-Encoding": "chunked",
      "X-Backside-Transport": "OK OK",
      "X-Cache-Key": "\/data\/2.5\/weather?q=london",
      "X-Global-Transaction-ID": "33711539"
   },
   "responseTime": 779,
   "statusCode": 200,
   "statusReason": "OK",
   "sys": {
      "country": "GB",
      "id": 5091,
      "message": 0.0057,
      "sunrise": 1458022408,
```

#6 | Design the UI for mobile screen using UI framework called JQuery Mobile

Here is the sample JQuery Mobile code for the UI. The UI contains header, footer and one text field and a submit button.

index.html

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>APIConsumption</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=0">
<link href="jqueryMobile/jquery.mobile-1.4.5.min.css" rel="stylesheet">
<link href="jqueryMobile/jquery.mobile-1.4.5.css" rel="stylesheet">
<link rel="stylesheet" href="css/main.css">
<script> window.$ = window.jQuery = WLJQ; </script>
<script src="jqueryMobile/jquery.mobile-1.4.5.js"></script>
</head>
<body style="display: none;">
<div data-role="page" id="pageone">
<div data-role="header" style="background-color: skyblue">
<h1 style="color: darkblue">Miracle Software Systems</h1> </div>
<div data-role="main" class="ui-content">
<p>Weather Report</p>
<form method="post" name="form1">
<div class="ui-field-contain">
<label for="fname"></label>
<input type="text" placeholder="Proxy Settings" name="cityName" id="cityName" value="City">
<input type="submit" data-inline="true" value="Submit" onclick="getWeatherReport()"> </div>
</form>
```

```
</div>
<div id="wrapper">
<ul id="itemsList"></ul>
</div>
<div data-role="footer" data-position="fixed" style="background-color: lightblue">
<h1 style="color: darkblue">MSS</h1>
</div>
</div>
<script src="js/initOptions.js"></script>
<script src="js/main.js"></script>
<script src="js/messages.js"></script>
</body>
</html>
```

#7 | Implement the JavaScript function for Invoking the HTTP Adapter

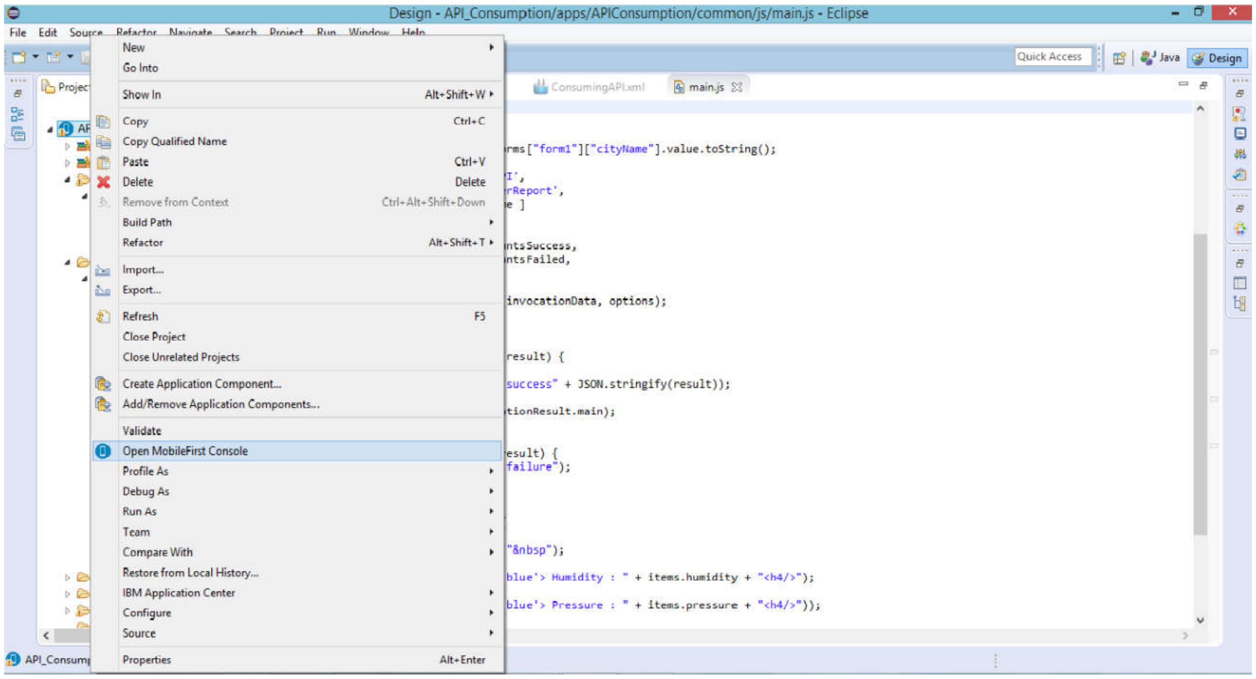
Here is the code snippet for invoking the HTTP adapter and parsing the JSON response which is coming from the server.

main.js

```
function wlCommonInit() {

}
function getWeatherReport() {
        var cityName = document.forms["form1"]["cityName"].value.toString();
        var invocationData = {
                        adapter : 'ConsumingAPI',
                        procedure : 'getWeatherReport',
                        parameters : [ cityName ]
        };
        var options = {
                        onSuccess : onGetAccountsSuccess,
                        onFailure : onGetAccountsFailed,
                timeout : 30000
        };
        WL.Client.invokeProcedure(invocationData, options);

}
function onGetAccountsSuccess(result) {
        //alert(result);
        WL.Logger.debug("Retrieve success" + JSON.stringify(result));
        displayFeeds(result.invocationResult.main);
}

function onGetAccountsFailed(result) {
        WL.Logger.error("Retrieve failure");

}
// Parsing the JSON response.
function displayFeeds(items) {
        var ul = $('#itemsList');
        ul = $('#itemsList').html("&nbsp");
        var li = $('<li/>').html(
                "<h4 style='color:blue'> Humidity : " + items.humidity + "<h4/>");
        li.append($('<li/>').html(
                "<h4 style='color:blue'> Pressure : " + items.pressure + "<h4/>"));
        li.append($('<li/>').html(
                "<h4 style='color:blue'> Tempeature : " + items.temp + "<h4/>"));
        li.append($('<li/>').html(
                "<h4   style='color:blue'>   Temp_Min   :   "   +   items.temp_min   +
"<h4/>"));
        li.append($('<li/>').html(
                "<h4   style='color:blue'>   Temp_Max   :   "   +   items.temp_max   +
"<h4/>"));
        ul.append(li);

}
```

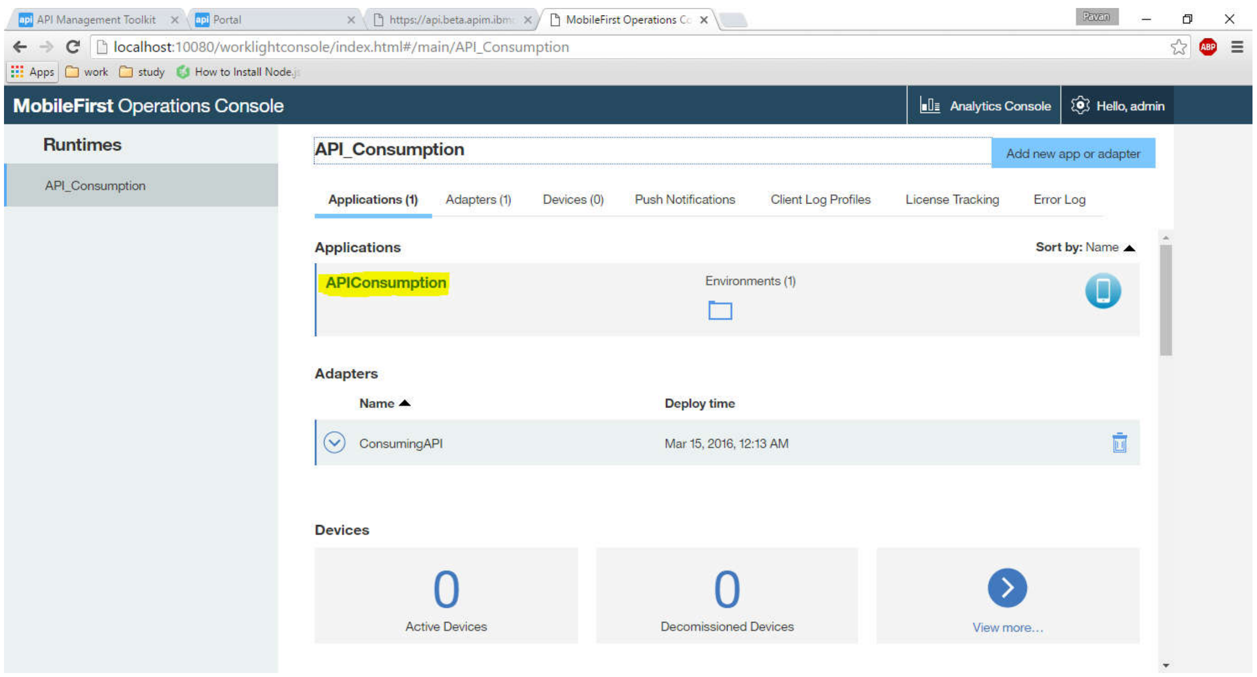
#8 | Running the entire Application

Now save all the files and then run the application by right click on application and click on Open Mobile First Console as shown in the figure.

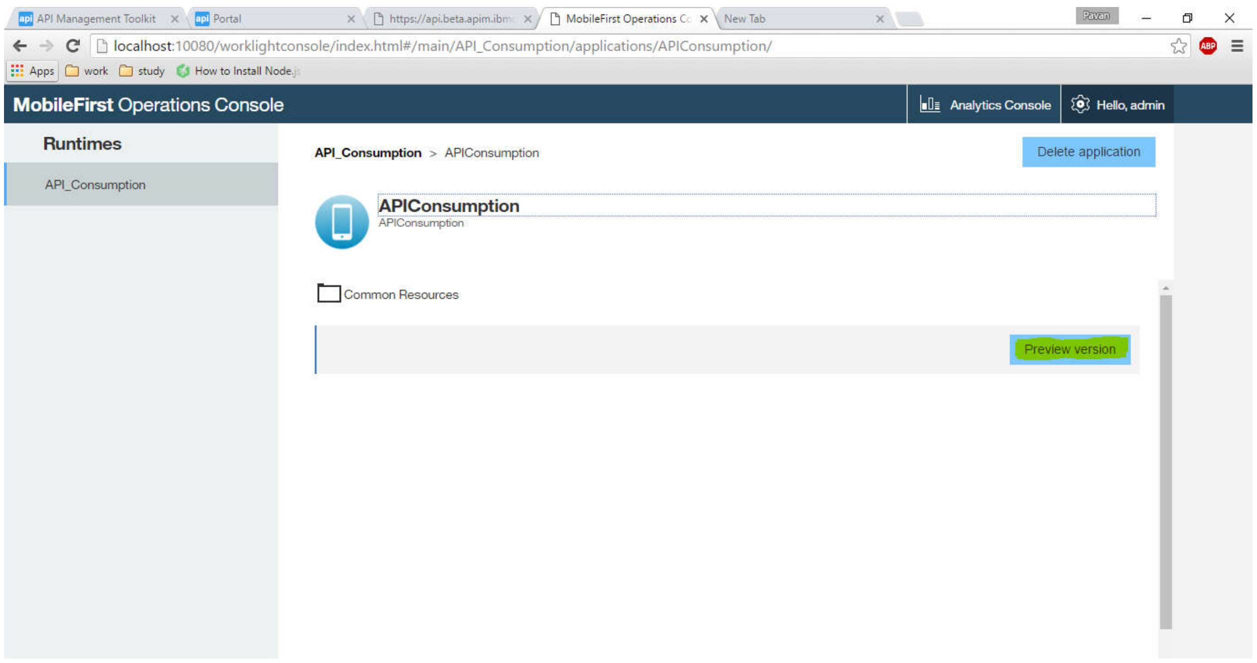


```
rms["form1"]["cityName"].value.toString();
I',
rReport',
e ]

ntsSuccess,
ntsFailed,

invocationData, options);

result) {
success" + JSON.stringify(result));
tionResult.main);

esult) {
failure");

"&nbsp");
blue'> Humidity : " + items.humidity + "<h4/>");
blue'> Pressure : " + items.pressure + "<h4/>"));
```

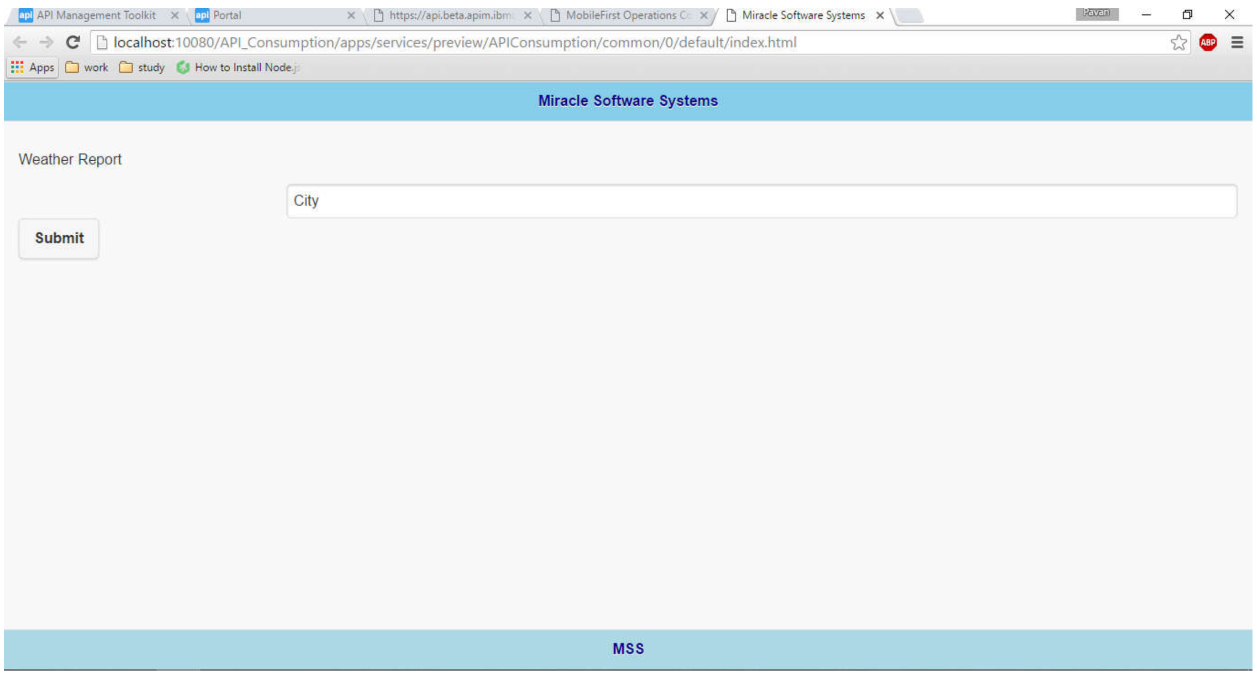
Then the application will run on the IBM Mobile First Platform Operations Console as shown in the figure. Here in the MF Console, one Application and an Adapter are deployed successfully. For opening the application, we need to click on the Applications button.



Now click on **API Consumption**.



Then, click on **PREVIEW** to open the current application as shown in the figure.


Miracle Software Systems

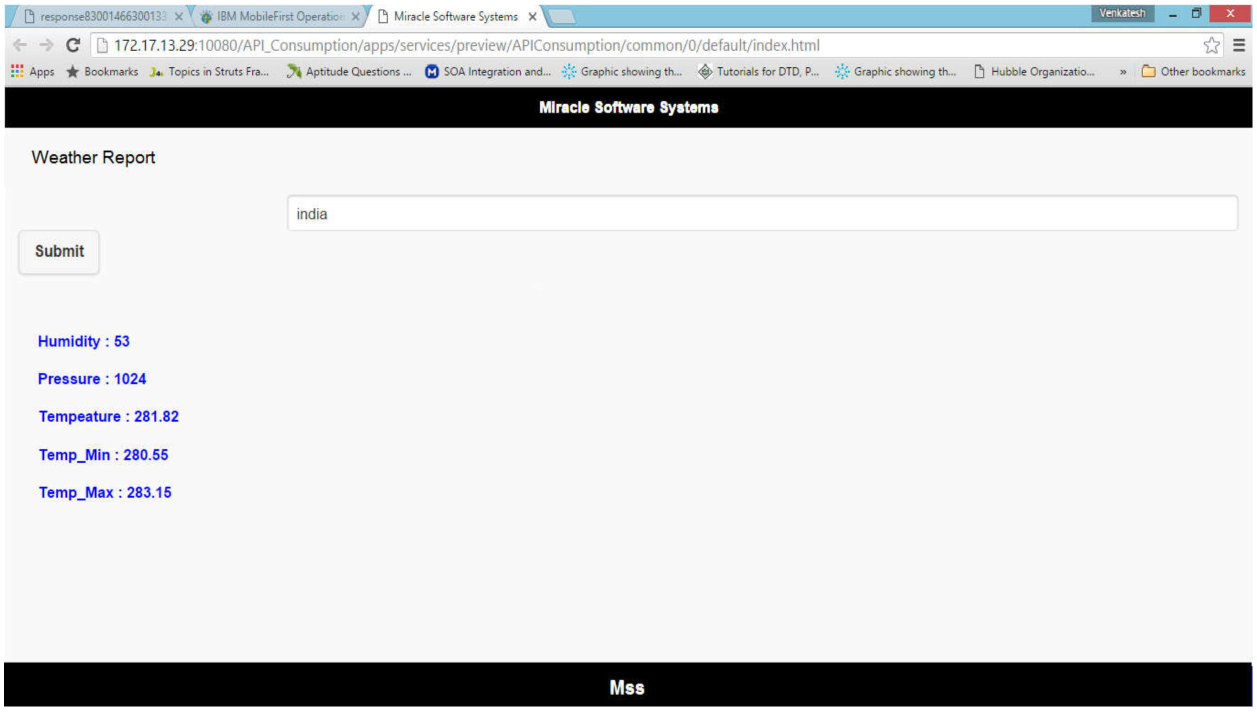
Weather Report

City

Submit

MSS

Enter a country name and click on submit to see the Weather Report of particular city. Here I'm giving India as the country name. Following is the final output of our application.


Miracle Software Systems

Weather Report

india

Submit

Humidity : 53

Pressure : 1024

Tempeature : 281.82

Temp_Min : 280.55

Temp_Max : 283.15

Mss