



IBM Mobile Foundation On Google Home Apps



© Copyright IBM Corporation 2017

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

This document is current as of the initial date of publication and may be changed by IBM at any time.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

Table of Contents

Pre-requisites	4
Introduction	4
Pre-requisites	4
Architecture Patterns	5
Github Download	6
Setup Dialogflow	7
Push Notification	11
Android App	15
Setting up of APP_GUID and CLIENT_SECRET	16
Nodejs Upload	19
Cloudant Addition	20
The Flow	21
References	22

Pre-requisites

You need to have:

IBM Bluemix account: <http://bit.ly/bmregistration>

Google account: <https://accounts.google.com/SignUp?hl=en>

Github: <https://github.ibm.com/akhanand/node-push-notification>

Introduction

We designed a Google Home App that lets you book Movie Ticket from your google home. User will login with “User ID” from app install in his/her device. Google Home will ask all required information then it will send OTP and after verification it will send book movie ticket.

Architecture patterns

1
Talk to ivyBoxOffice



OTP
2

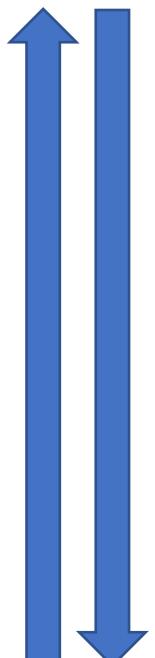


1
Req. Movie, Tickets, Date



Google Services

2
Details Send



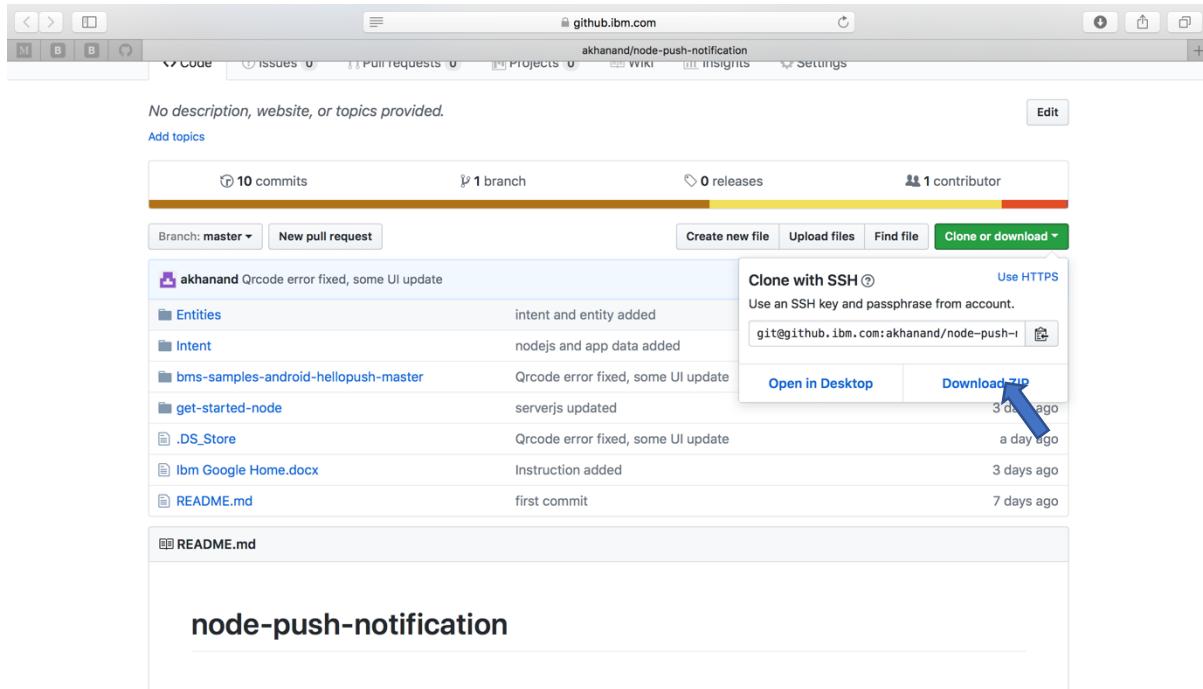
2
Details send
1
OTP send



IBM Bluemix

Github Download

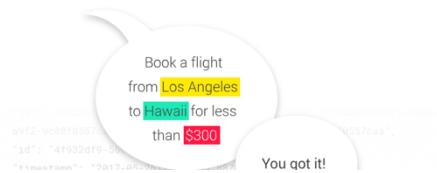
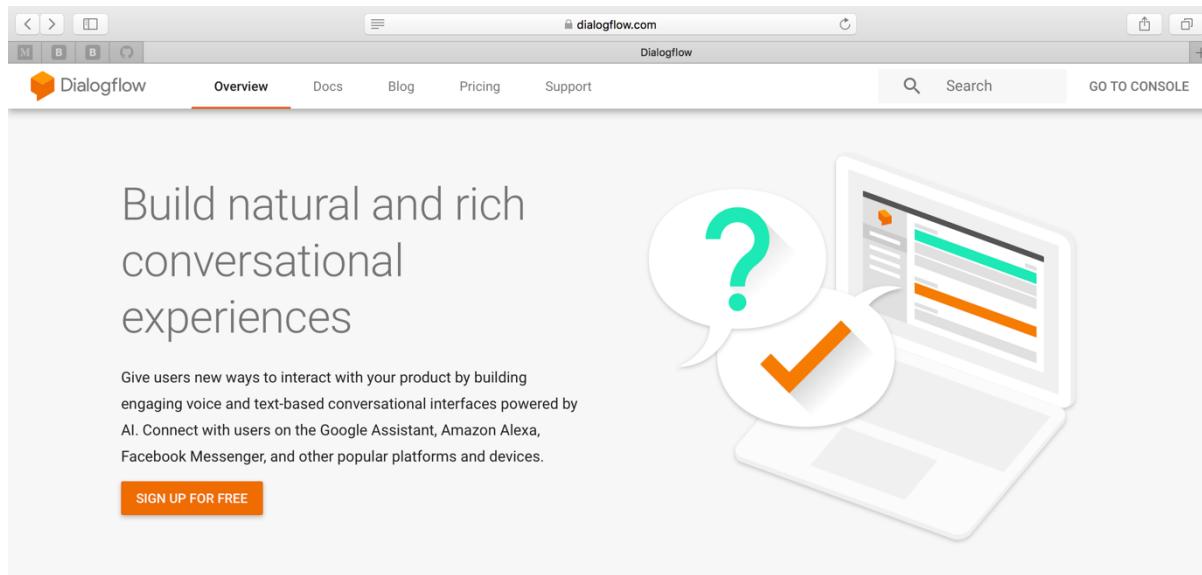
Go to Github link <https://github.ibm.com/akhanand/node-push-notification> and download.



Setup the Dialogflow

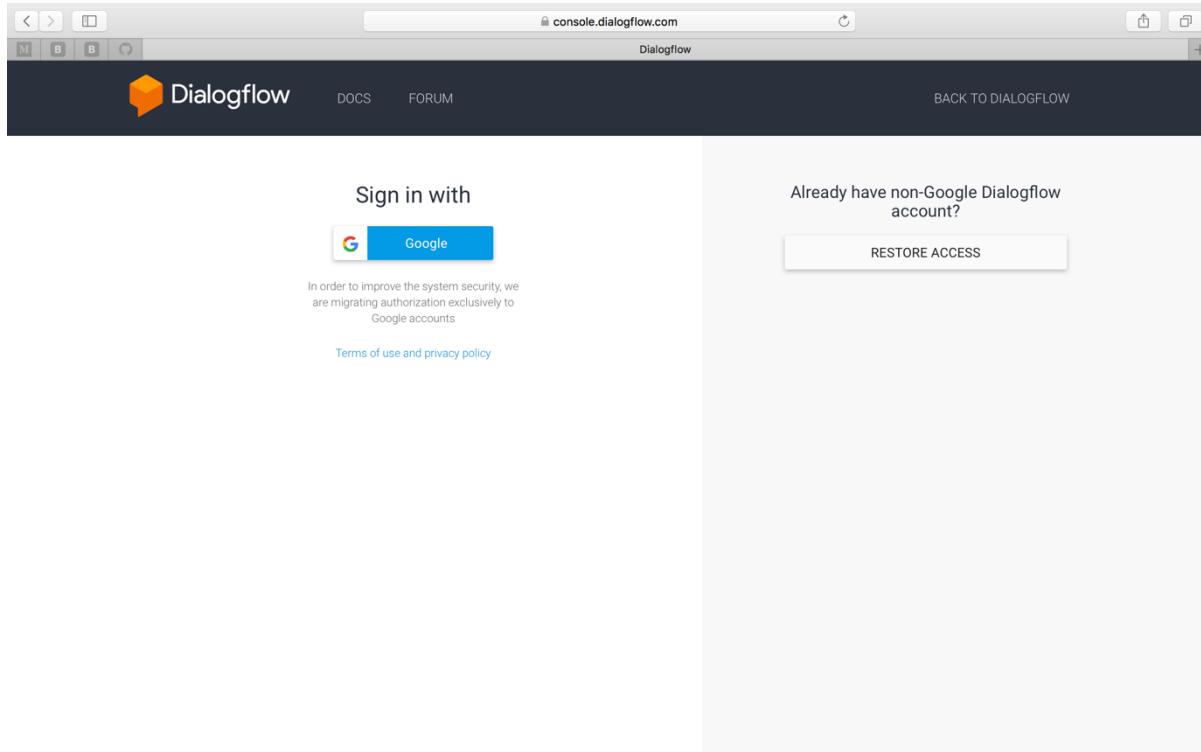
Go to Dialogflow, link: <https://dialogflow.com>

Now click on “GO TO CONSOLE”

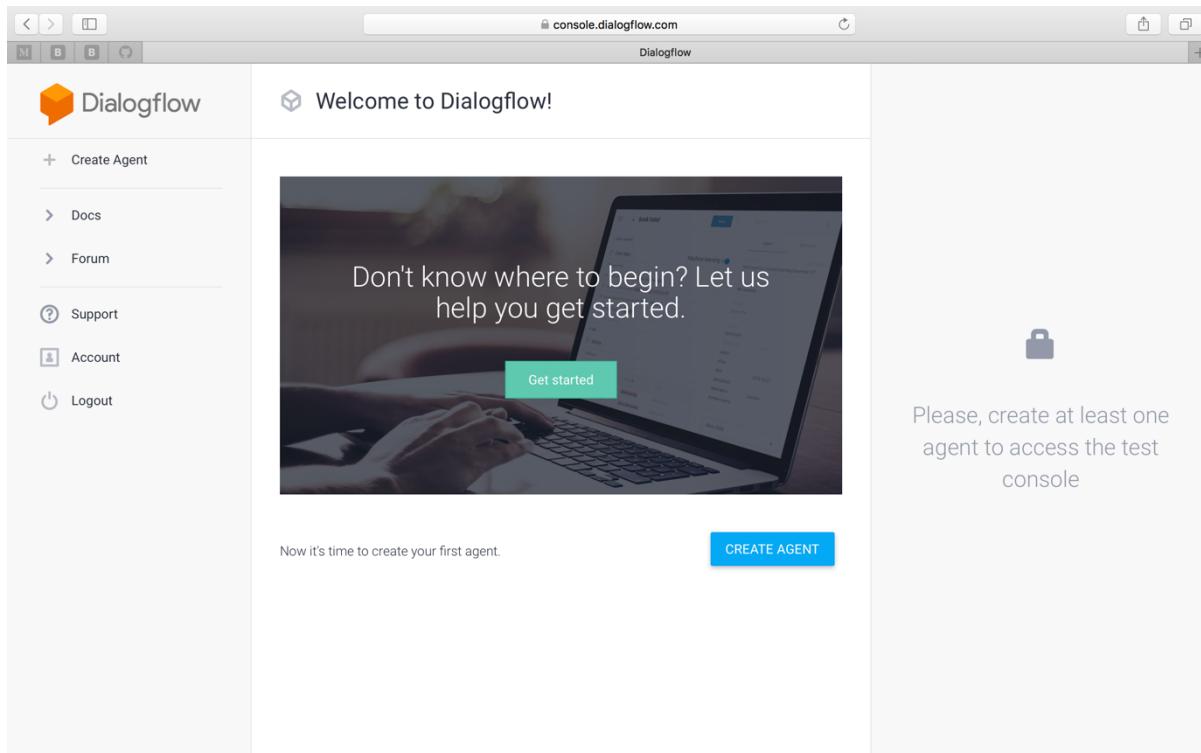


Use machine learning to understand what users are saying

Then click on Google to sign in.



Click on CREATE AGENT to create agent.



Give AGENT NAME and then click CREATE.

Now click on “Intent” to add Intent and then to overflow icon



The screenshot shows the Dialogflow console interface. On the left, a sidebar menu includes 'Sample' (with 'en' selected), 'Intents' (selected and highlighted in blue), 'Entities', 'Training [beta]', 'Integrations', 'Analytics [new]', 'Fulfillment', 'Prebuilt Agents', 'Small Talk', 'Docs', 'Forum', 'Support', and 'Account'. The main area is titled 'Intents' and contains a 'CREATE INTENT' button. Below it is a search bar labeled 'Search intents' with a magnifying glass icon. A list of intents shows 'Default Fallback Intent' and 'Default Welcome Intent'. A note below the list says 'No regular intents yet. [Create the first one.](#)'. To the right, there's a 'Try it now' section with a text input field, a note about using the test console, and a link to see how it works in Google Assistant.

Go to node-push-notification > Intent from the folder you downloaded.
Upload all Intent from the Intent folder.

Now go to “Default Welcome Intent” and in response section delete all the earlier text response.

Click on ADD RESPONSES and add response as “Hi! Welcome to MyBoxOffice”. Then SAVE it.

Now click on “Entities”, go to overflow and upload entity from Entities folder.

The screenshot shows the Dialogflow interface for configuring an intent. On the left, a sidebar lists various sections: Sample, Intents (selected), Entities, Training [beta], Integrations, Analytics [new], Fulfilment, Prebuilt Agents, Small Talk, Docs, Forum, Support, and Account. The main panel displays the 'Default Welcome Intent'. At the top, there's a table for parameters: REQUIRED, PARAMETER NAME, ENTITY, VALUE, and IS LIST. Below this is a 'Responses' section under the 'DEFAULT' tab, which contains a 'Text response' block with two entries: '1 Hi ! Welcome to MyBoxOffice' and '2 Enter a text response variant'. A 'ADD RESPONSES' button is visible below the responses. A note at the bottom says 'Set this intent as end of conversation'. On the right side of the main panel, there are 'Try it now' and 'See how it works in Google Assistant' buttons, along with a message about agent training starting. A status bar at the bottom right shows 'Agent training started' and 'OK'.

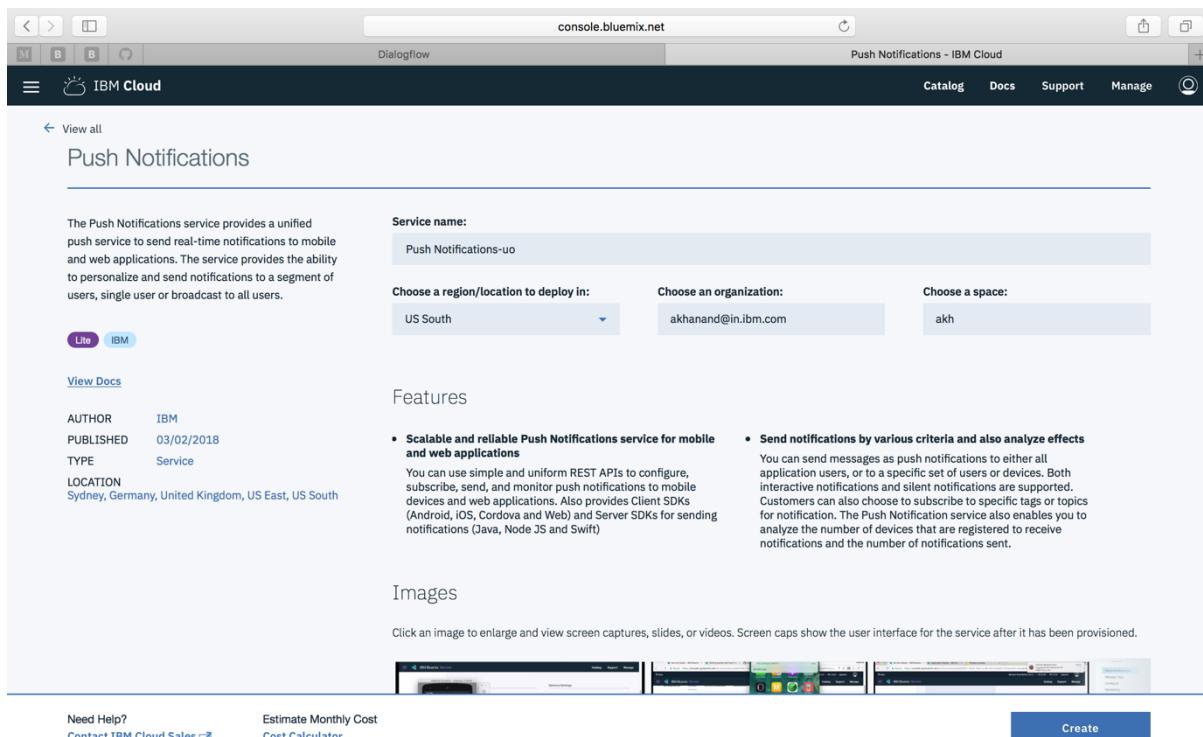
Push Notification Setup

- For better understanding watch this video on Push notification.
https://www.youtube.com/watch?v=dQ1WcY_III4

Instruction:

Now go to IBM Cloud link, <https://console.bluemix.net/developer/mobile/dashboard> and Log in.

Click on Catalog at top, search “Push Notification” and click on it.

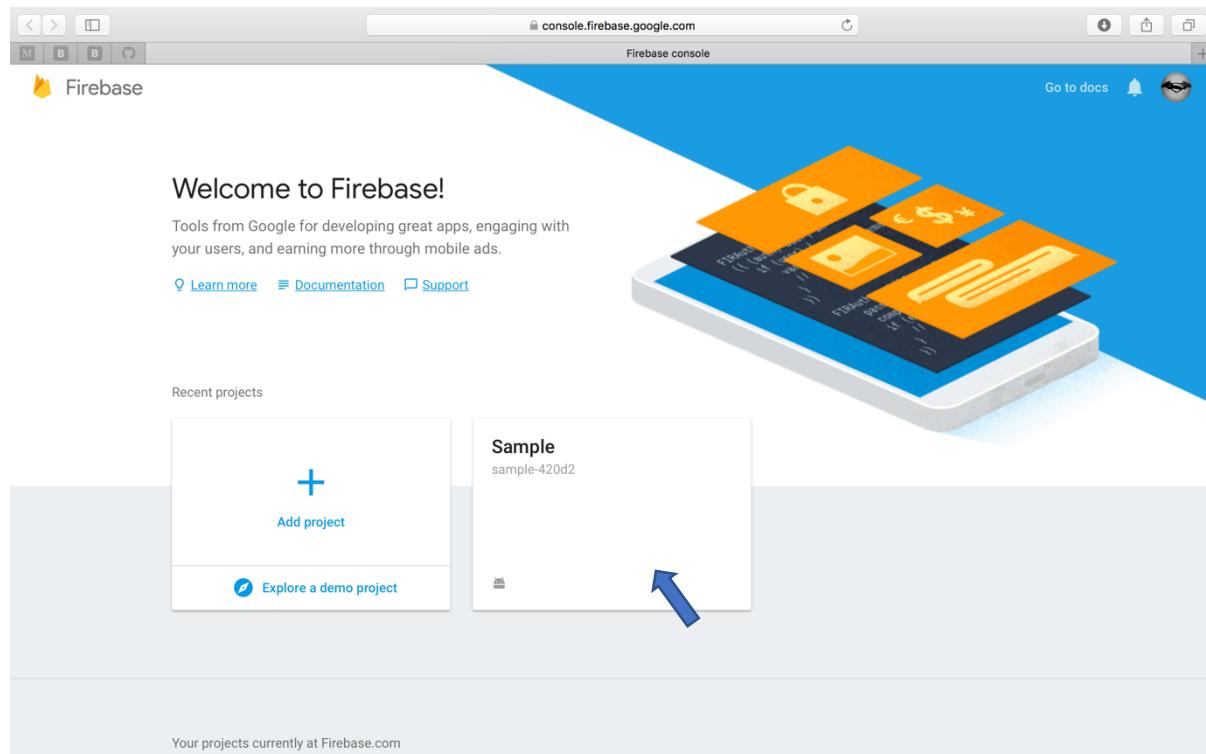


The screenshot shows the IBM Cloud catalog interface. The top navigation bar includes links for Catalog, Docs, Support, and Manage. The main content area is titled "Push Notifications". On the left, there's a sidebar with a "View all" link and a "Push Notifications" section. The main panel has fields for "Service name" (set to "Push Notifications-uo"), "Choose a region/location to deploy in" (set to "US South"), "Choose an organization" (set to "akhanand@in.ibm.com"), and "Choose a space" (set to "akh"). Below these fields, there are sections for "Features" and "Images". The "Features" section lists "Scalable and reliable Push Notifications service for mobile and web applications" and "Send notifications by various criteria and also analyze effects". The "Images" section shows three screenshots of the service interface. At the bottom, there are links for "Need Help?", "Estimate Monthly Cost", and "Contact IBM Cloud Sales", along with a "Create" button.

- Remember the region you going to create the push notification.

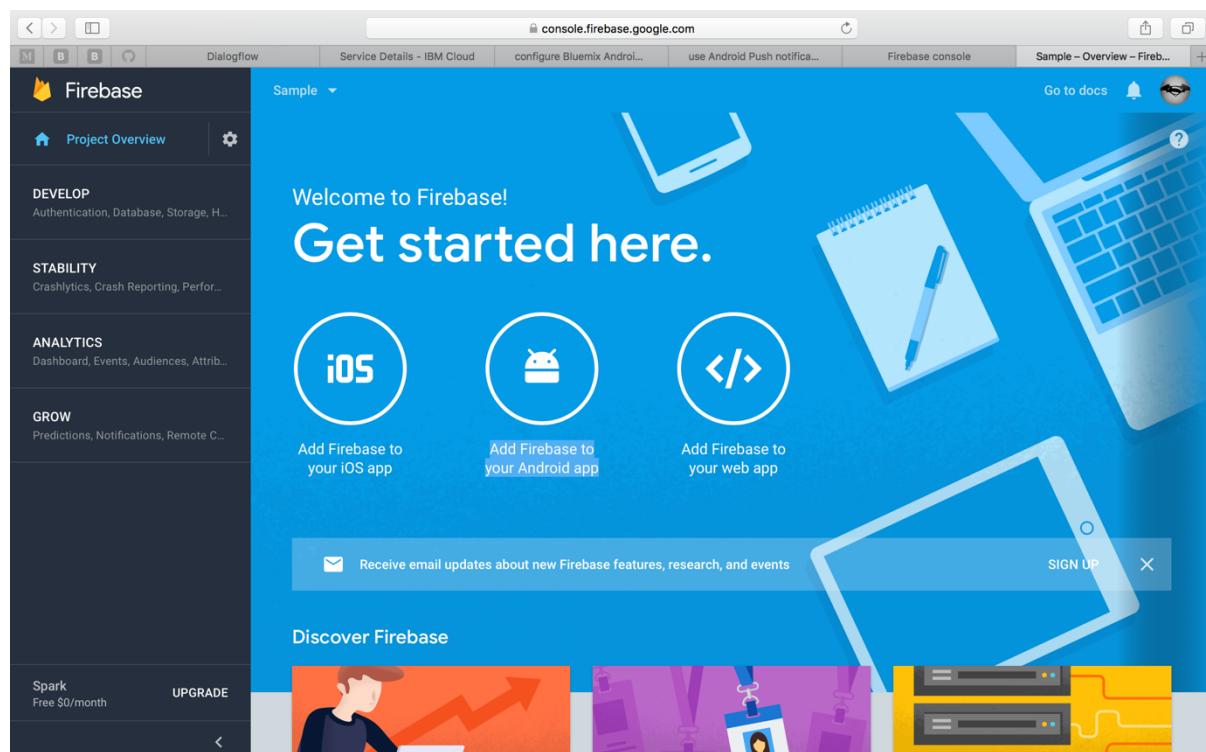
Now click on “Create”.

Now go to Firebase via this link <https://console.firebaseio.google.com/u/2/?pli=1>
Your project will be there just click over that.

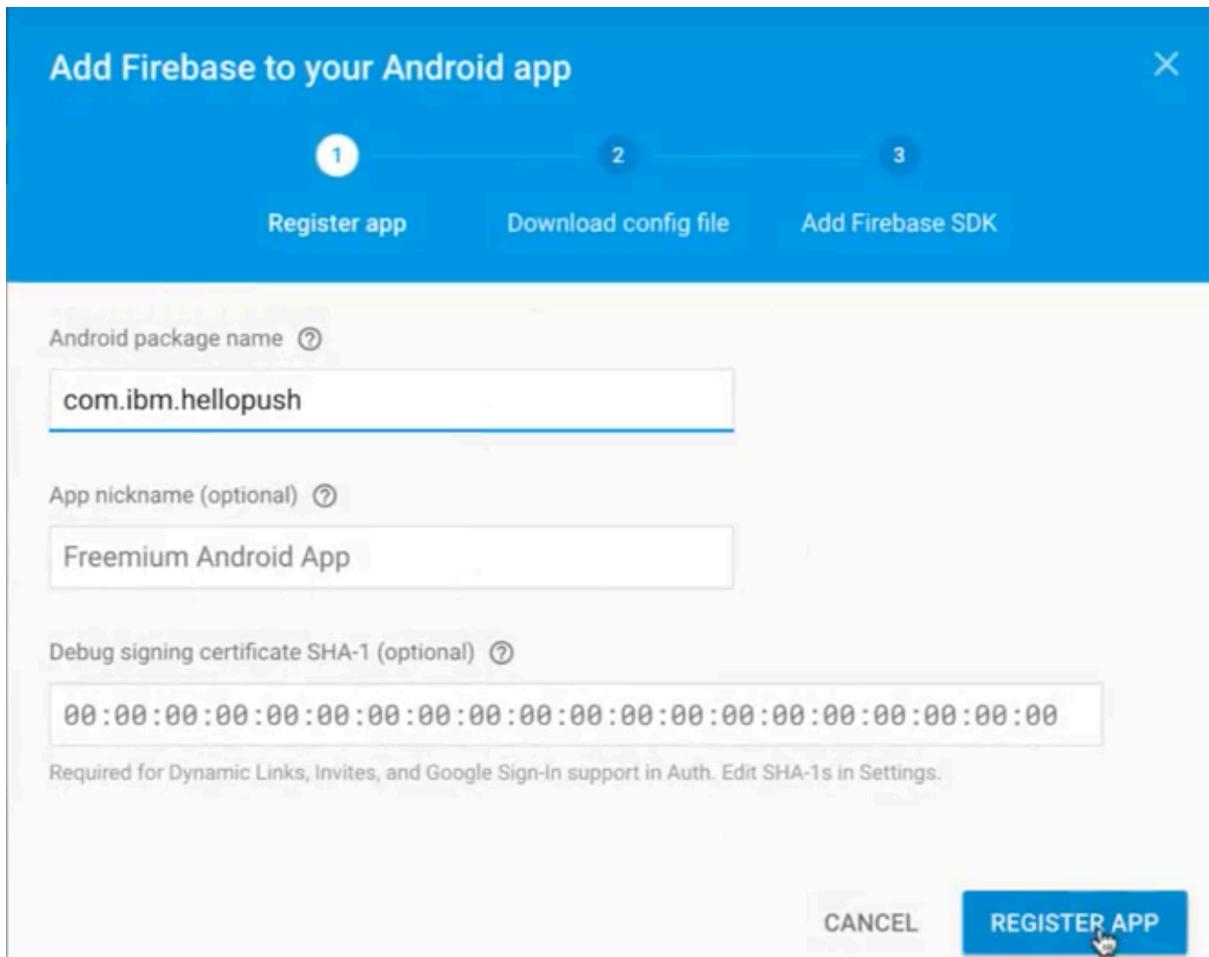


Here project name is “Sample”.

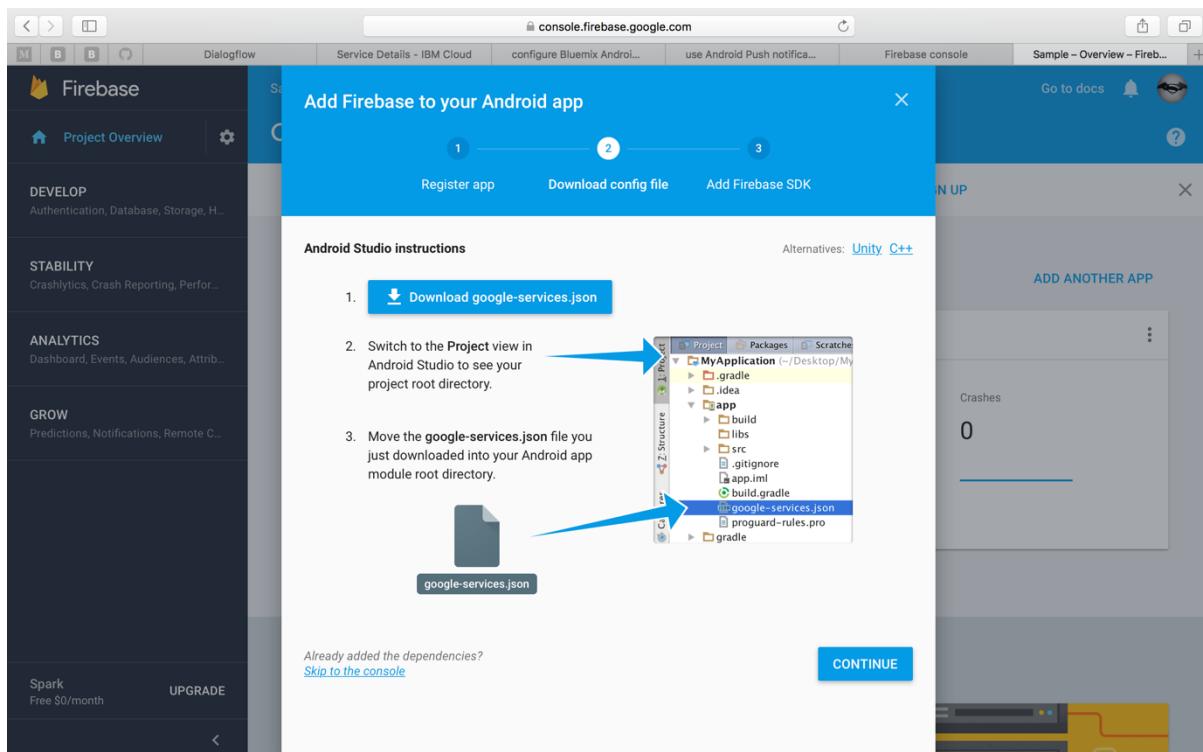
Now click on “Add Firebase to your Android app”



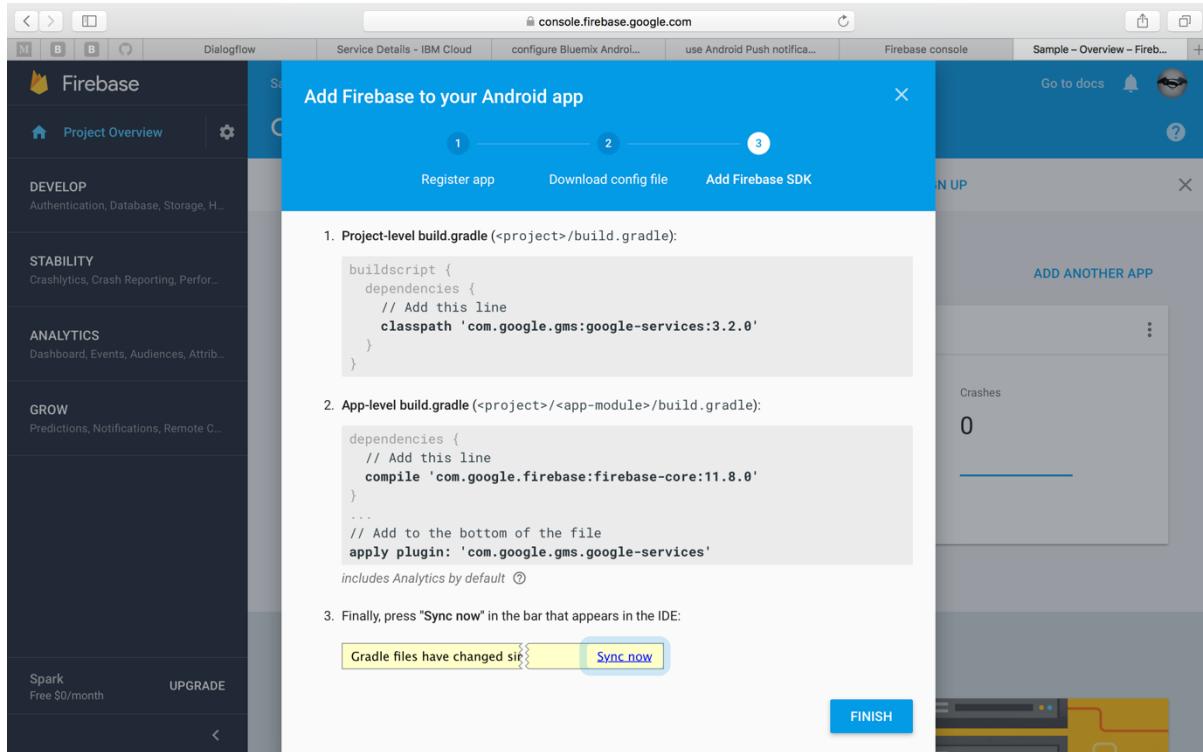
Add “com.ibm.hellopush” in Adroid package name.



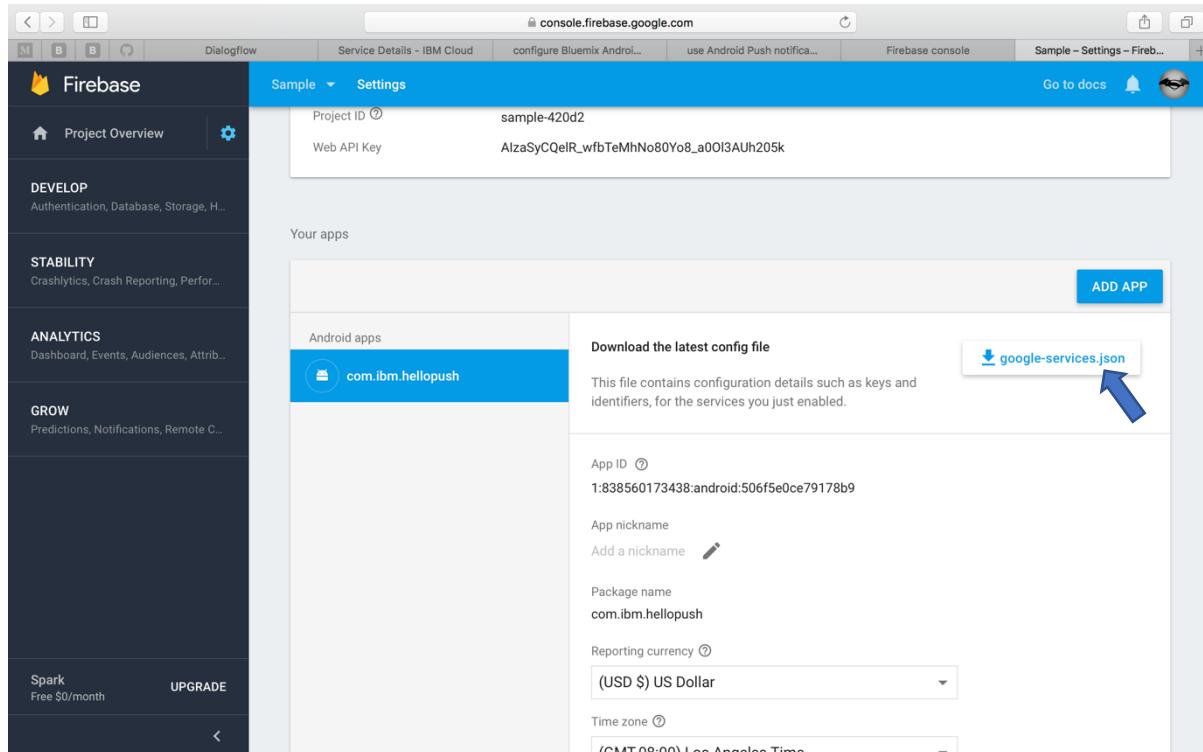
Click on CONTINUE.



Then on FINISH.



Now click on project setting  and Download “google-services.json”.



ANDROID APP:

Go to “bms-samples-android-hellopush-master” from “node-push-notification” folder which you download from Github and paste “google-services.json” file inside > hellopush > app

Now open the Android Studio, open project



Go to “bms-samples-android-hellopush-master” folder and open “hellopush”

After opening app go to “MainActivity” and in that you need to give “APP_GUID” and “CLIENT_SECRET” from cloudant and firebase.

Setting up of APP_GUID and CLIENT_SECRET

Go to firebase, Project setting > Cloud Messaging
Copy “Sender ID” and “Legacy Server Key”

The screenshot shows the Firebase Cloud Messaging settings page. The URL in the browser is console.firebaseio.google.com. The page title is "Settings". The "CLOUD MESSAGING" tab is selected. Under "Project credentials", there is a table with two rows:

Key	Token
Server key	AAAAwz4UzX4:APA91bHZCJP5cKs9RKihcLf7NCBNESDLico9DFeUpffX-MFLhTyHoqXOTUDceEYT91PK4JThNBdnJLSDGE_fS4lv_-1C7whajpUxt4fLH1gbuft3_dRBhzHzEc-JnUGDsON3EuVdw

Below the table, there are fields for "Legacy server key" and "Sender ID", both of which are redacted with blue ink.

Under "iOS app configuration", it says "You don't have an iOS app". Under "Web configuration", there is a section for "Push notification" which is also redacted.

Now to go to IBM cloud Dashboard <https://console.bluemix.net/dashboard/apps>
Push Notification > Manage > Configure

Paste “Sender ID” and “Legacy Server Key” in “Sender ID” and “API Key” field respectively and Save it.

The screenshot shows the IBM Cloud Push Notifications configuration interface. On the left sidebar, under the 'Mobile' section, 'Manage' is selected. In the main area, the service name is 'Push Notifications-uo'. The location is 'US South', org is 'akhanand@in.ibm.com', and space is 'akh'. The 'Configure' tab is active, showing the 'APNs Push Credentials' and 'GCM/Firebase Push Credentials' sections. Under APNs, the environment is set to 'Sandbox/Development APNs Server'. Under GCM/Firebase, the 'Sender ID/Project Number' and 'API Key' fields are empty. A 'Choose file' button is available for each. A 'Save' button is at the bottom. A 'Mobile options' button is located in the top right corner of the configuration panel.

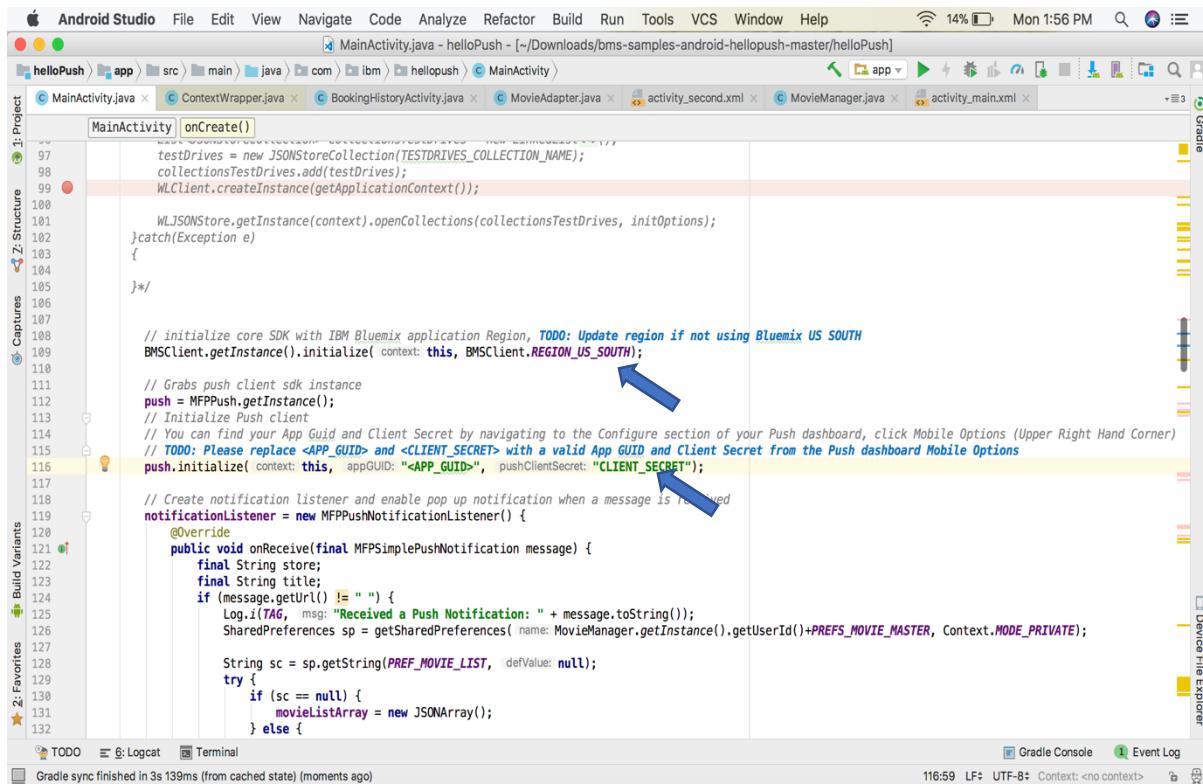
Go to “Mobile options” and copy “AppGuid” and “Client secret”

The screenshot shows the same configuration interface as above, but with a modal window titled 'Mobile options' overlaid. The modal contains a warning message: 'The use of App Route parameter is deprecated. Please use the Service Credentials tab to obtain credentials required to access the Push Notifications service.' It also displays three fields: 'AppGuid' (redacted), 'App secret' (redacted), and 'Client secret' (redacted). Each field has a copy icon (a blue square with a white 'C') to its right. The background configuration panel is partially visible behind the modal.

Now to android studio and paste “AppGuid” and “Client secret” in “APP_GUID” and “CLIENT_SECRET” fields.

Also change the Region in which Push Notification is created.

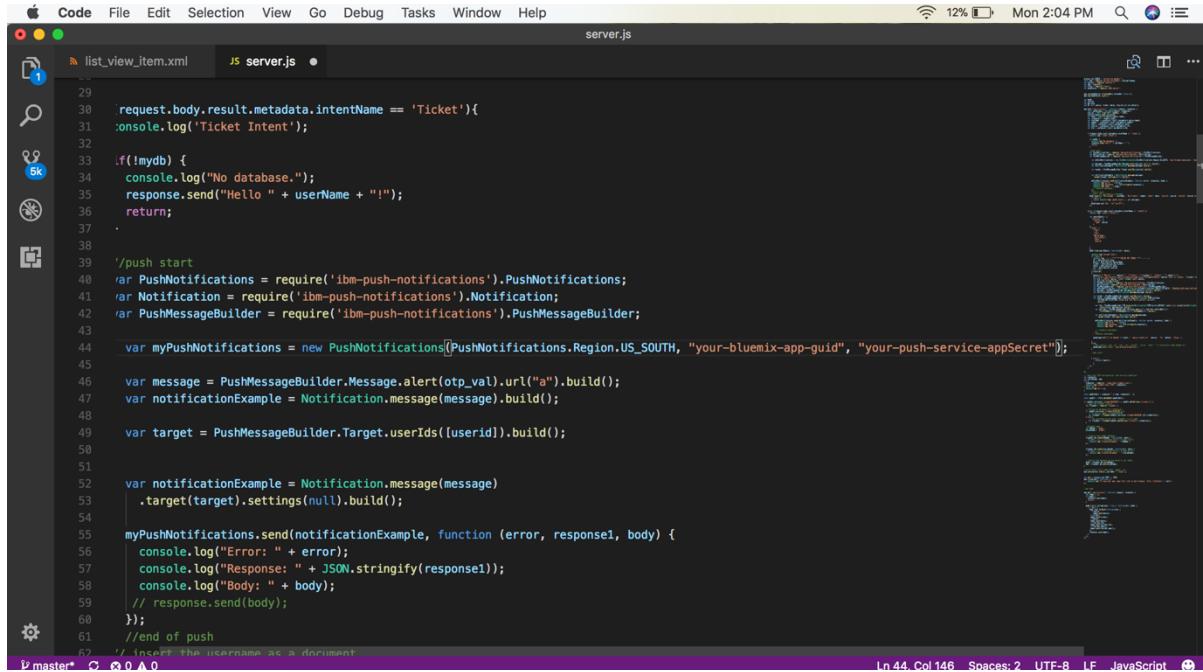
- Use References (1) at last page for giving the region name.



```
97     testDrives = new JSONStoreCollection(TESTDRIVES_COLLECTION_NAME);
98     collectionsTestDrives.add(testDrives);
99     WLClient.createInstance(getApplicationContext());
100
101    WLJSONstore.getInstance(context).openCollections(collectionsTestDrives, initOptions);
102    } catch(Exception e)
103    {
104    }
105
106
107
108    // initialize core SDK with IBM Bluemix application Region, TODO: Update region if not using Bluemix US SOUTH
109    BMSSClient.getInstance().initialize(context: this, BMSSClient.REGION_US_SOUTH);
110
111    // Grabs push client sdk instance
112    push = MFPPush.getInstance();
113    // Initialize Push client
114    // You can find your App Guid and Client Secret by navigating to the Configure section of your Push dashboard, click Mobile Options (Upper Right Hand Corner)
115    // TODO: Please replace <APP_GUID> and <CLIENT_SECRET> with a valid App GUID and Client Secret from the Push dashboard Mobile Options
116    push.initialize( context: this, appGUID: <APP_GUID>, pushClientSecret: "CLIENT_SECRET");
117
118    // Create notification listener and enable pop up notification when a message is received
119    notificationListener = new MFPPushNotificationListener()
120    {
121        @Override
122        public void onReceive(final MFPSimplePushNotification message) {
123            final String store;
124            final String title;
125            if (message.getUrl() != " ") {
126                Log.i(TAG, msg: "Received a Push Notification: " + message.toString());
127                Sharedpreferences sp = getSharedpreferences( name: MovieManager.getInstance().getUserId() + PREFS_MOVIE_MASTER, Context.MODE_PRIVATE);
128
129                String sc = sp.getString(PREF_MOVIE_LIST, defaultValue: null);
130                try {
131                    if (sc == null) {
132                        movieListArray = new JSONArray();
133                    } else {
```

NODEJS Upload

Go to “node-push-notification” > get-started-node and open “server.js”
Copy and Paste “AppGuid” and “App secret” in “server.js” and save.



The screenshot shows the IBM Cloud Code Editor interface. The top menu bar includes Code, File, Edit, Selection, View, Go, Debug, Tasks, Window, and Help. The title bar shows "server.js". The main editor area contains the following code:

```
list_view_item.xml JS server.js ●
29
30 if(request.body.result.metadata.intentName == 'Ticket'){
31 console.log('Ticket Intent');
32
33 if(!mydb) {
34   console.log("No database.");
35   response.send("Hello " + userName + "!");
36   return;
37 }
38
39 '/push start
40 var PushNotifications = require('ibm-push-notifications').PushNotifications;
41 var Notification = require('ibm-push-notifications').Notification;
42 var PushMessageBuilder = require('ibm-push-notifications').PushMessageBuilder;
43
44 var myPushNotifications = new PushNotifications(PushNotifications.Region.US_SOUTH, "your-bluemix-app-guid", "your-push-service-appSecret");
45
46 var message = PushMessageBuilder.Message.alert(otp_val).url("a").build();
47 var notificationExample = Notification.message(message).build();
48
49 var target = PushMessageBuilder.Target.userIds([userid]).build();
50
51
52 var notificationExample = Notification.message(message)
53   .target(target).settings(null).build();
54
55 myPushNotifications.send(notificationExample, function (error, response1, body) {
56   console.log("Error: " + error);
57   console.log("Response: " + JSON.stringify(response1));
58   console.log("Body: " + body);
59   // response.send(body);
60 });
61
62 // end of push
63 // Insert the username as a document
```

The status bar at the bottom indicates "Ln 44, Col 146, Spaces: 2, UTF-8, LF, JavaScript".

Open terminal, move to directory /node-push-notification/get-started-node

Type “npm install” and then “npm install ibm-push-notifications –save”

Do “bx login –sso” and then “bx api <API-endpoint>”

Also <API-endpoint> is Region name.

📘 Use References (2) for <API-endpoint>

Then do “bx cf push”

CLOUDANT Addition

Go to IBM Cloud link, <https://console.bluemix.net/developer/mobile/dashboard>

Click on Catalog at top, search “Cloudant NoSQL DB”.

After clicking on “Cloudant NoSQL DB”, click on Create.

The screenshot shows the IBM Cloud service catalog interface. A new service named "Cloudant NoSQL DB-9i" is being created. The configuration includes deployment in "US South", organization "akhanand@in.ibm.com", and space "akh". The "Features" section highlights "Fully managed DBaaS" and "Powerful query, analytics, replication, and sync". Below the features are sections for "Images" (with a placeholder image) and "Pricing" (with a link to "Cost Calculator"). A "Create" button is visible at the bottom right.

Now go to DashBoard > getstartednode > Connections

Click “Create connection” and select Cloudant from that and do “Connect”

Go to “Routes” at top get the link, it will be like <https://getstartednode-s.mybluemix.net>

Now go to Dialogflow, then Fulfillment.

Enable the webhook and in URL Field paste the link got above and add “/api/visitors” in url.

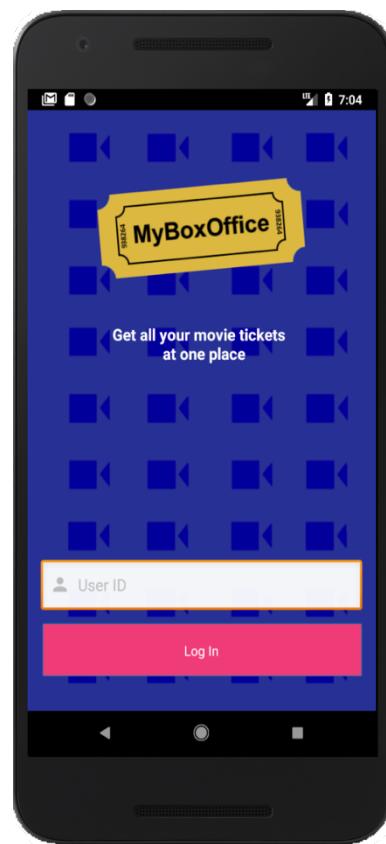
It will look like this

<https://getstartednode-s.mybluemix.net/api/visitors>

and SAVE.

THE FLOW

Start the app from Android Studio.
and give “User ID”



And it will register with your User ID.

Go to your Dialogflow > Google Assistant

Type “talk to my test app”

And then follow the instruction and you will receive a OTP, enter it.

And last booked movie ticket will be send to your app.

References

1. Guideline for specifying region in “server.js” and “Android Studio”

US SOUTH: **.Region.US_SOUTH**

UK: **.Region.UK**

SYDNEY: **.Region.SYDNEY**

FRANKFURT: **.Region.FRANKFURT**

US_EAST: **.Region.US_EAST**

2. Guideline for “api <API-endpoint>”

Region name	Geographic location	API endpoint
US South region	Dallas, US	api.ng.bluemix.net
US East region	Washington, DC, US	api.us-east.bluemix.net
United Kingdom region	London, England	api.eu-gb.bluemix.net
Sydney region	Sydney, Australia	api.au-syd.bluemix.net
Germany region	Frankfurt, Germany	api.eu-de.bluemix.net