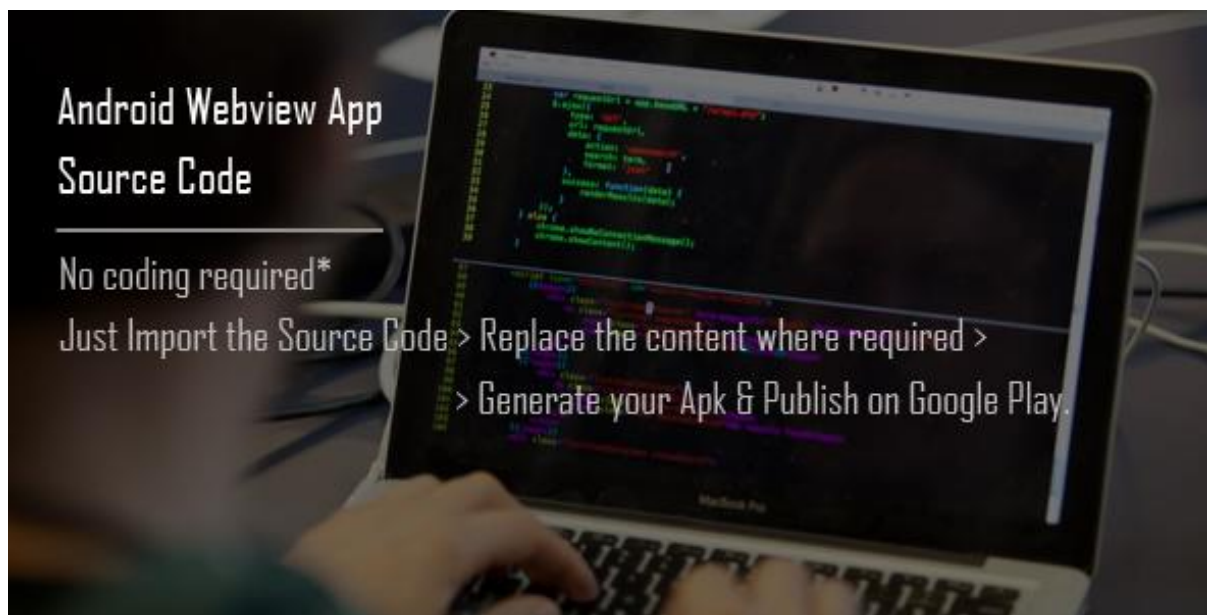


Android Webview App

Documentation



Index:

1. [Introduction](#)
2. [Contents in the Zip file](#)
3. [Import Project and Libraries to your Workspace](#)
4. [Replace the Urls](#)
5. [Splashscreen](#)
6. [Menu](#)
7. [Actionbar](#)
8. [Push Notifications](#)
 - 8.1 [Pushbots Signup](#)
 - 8.2 [Create GCM Project](#)
 - 8.3 [Create GCM API KEY](#)
9. [Sending Notifications](#)
 - 9.1 [Types](#)
 - 9.2 [BigPictureStyle](#)
 - 9.3 [BigTextStyle](#)
10. [Export and Publishing on Google Play](#)

Introduction:

A webview android app is kinda app where we won't parse the data from server. We simply place a url in the code and the app launch that url or run that activity whenever we open the app. The url which we going to place is a permanent url i.e., Url of your blog or Website or Service. So you should be ready with your mobile friendly website. Even Google is recommending all websites owners to make their website a mobile ready.

Already have a mobile ready website/blog then go ahead and place your url and generate the apk and host it on Google Play.

Don't have a website/blog? Are you a startup?

Then get a hosting space from [here](#), domain from [here](#) and host it on wordpress. For any assistance or support do contact us at nkssrikanth@outlook.com

Contents in the Zip file:

1. Source code: Again its a zip file you have to extrat it.
Inside Zip you find OK Vizag, Pull Down to Refresh, DiscreteAppRate, Library, Main and Android Support v7 appcompact folders.
2. Documentation

As stated you should have Eclipse & Android preinstalled on your PC and some basic knowledge on how to configure and fix built properities on first import of the project.

First and the Foremost thing:

Import the porjects and Libraries to your Workspace:

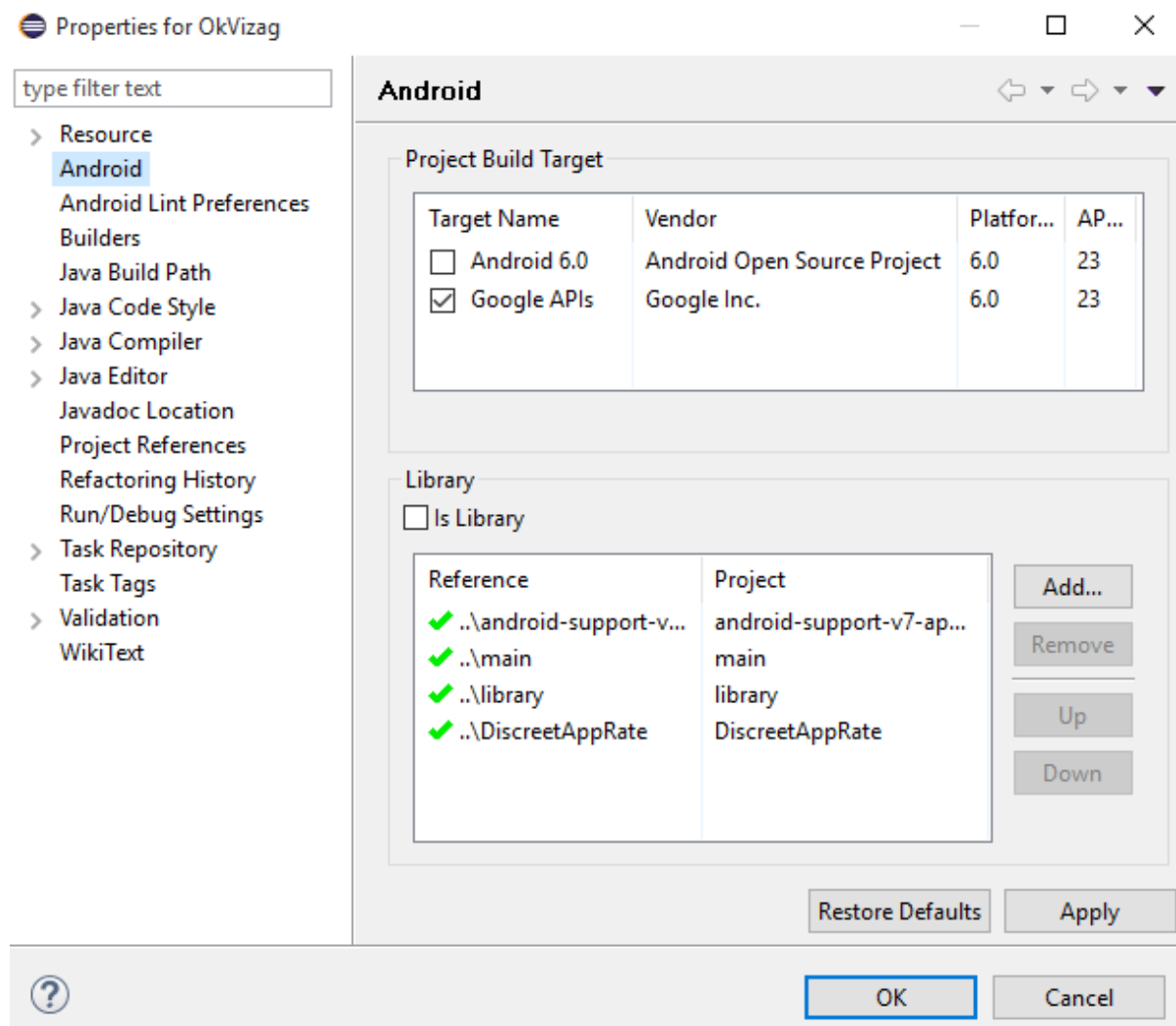
1. Open Eclipse > File > Import > Android project > browse the extracted folder of the source code > select and check the "Copy Contents into Workspace" and click OK.
2. Fix the built properties and dependencies. Make sure there are no errors after importing.
3. Similary import other projects/libraries into your workspace.

Now create a New Android Project with your Own Package Name (eg: "com.okvizag.ctrlplusu" is the package name of the source code. Similarly, create a new project with New Package Name). After creating a new project, copy the Activities(Java files), Layouts (XML files), Values (XML files), Menus (XML files), Dwarawables (png images), libs etc from source code to your project.

Or simply **refactor** the **source project** with your **package name** if you don't need a backup.

➔ Add the libraries by going through the project properties > Android > Library > ADD.

Go through the screenshot below.



Replace the urls:

The Source code contains total 4 activities of which two belong to webviews. One is Main Activity (loads the main website) and other is WebView Activity(loads the url from notification).

Main Activity and WebViewActivites:

1. Find the Main Activity, WebViewActivity and MyAppWebViewActivities for ["http://okvizag.com/stream"](http://okvizag.com/stream) and "okvizag.com" and replace them with your urls ["http://abcd.com/xxxx"](http://abcd.com/xxxx) , "abcd.com"
2. **3 Dot Menu:** This is the menu you find at right side top corner on the action bar. Replace the urls for those options "Rate this app", "Share this app", "How to Post" etc with yours. Adding options under menu is the basic knowledge you should acquire if you don't.

SplashScreen:

1. Replace the the SplashScreen in the drawable folder with yours and replace the name of the image in the SplashScreen layout with yours.[background_9.png and oklogo.png are the splashscreen background and logo images you find in the drawable folders replace them with yours and the names marked in yellow in the below image in “Layouts > Splashscreen.xml”]



2. You can set the logo of the app at the begining or by browsing the png image in the **Android Manifest**.
3. Similary replace other images for Menu options and others.

For generating Splash Screens, Icons that best fits in all screen sizes go [here](#). Simply, upload a high resolution Splash Screen you designed and click “Generate” within a minute a .zip file gets downloaded. Extract it and copy the images in the respective folders to their respective folders in the Eclipse > Res > Drawables.

Menus:

You find two menus under **Res > Menus > main.xml** (for MainActivity) & **web.xml** (for WebViewActivity). So make sure whatever chages you do in the main.xml the same changes should be done in the web.xml. Its not necessary as they are independent but is recommended.

ActionBar:

In MainActivity.java you find some codes amrked as comments at the beginning. You can make use of them to

1. Hide or UnHide the ActionBar
2. Change the color of the ActionBar by replacing the color code “#000000”(black)

Push Notifications:

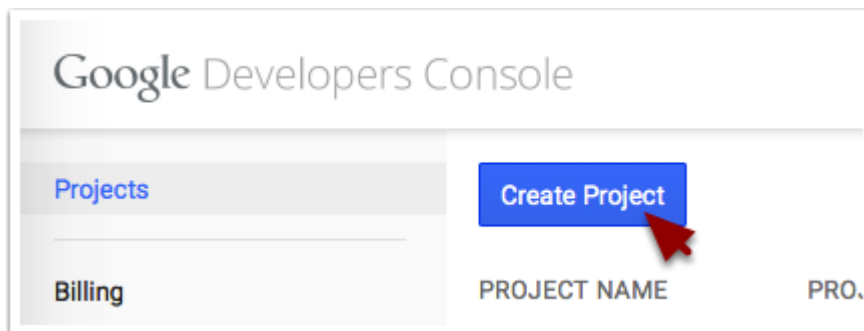
For push notifications the service we are using here is [PUSHBOTS](#). It is free for starters, below is the guide on how to integrate this functionality.

1. First SignUp [here](#).
2. Create an 'App' by entering your 'App name' in the given field and hit Enter.
3. The **Application ID** and **Application Secret ID's/ Keys** are generated now. Which are required for API calls. Keep this tab aside we may need in the next step.
4. As I have already integrated the push notifications. What you suppose to do is to replace the keys in the source code with yours. Open Pushbots.xml from **Res > Values > Pushbots.xml**

Now you have to create a project on Google Developers Console page for Google Cloud Messaging GCM.

Steps:

1. Open the [Google Developers Console page](#).
2. Click on **Create Project**:

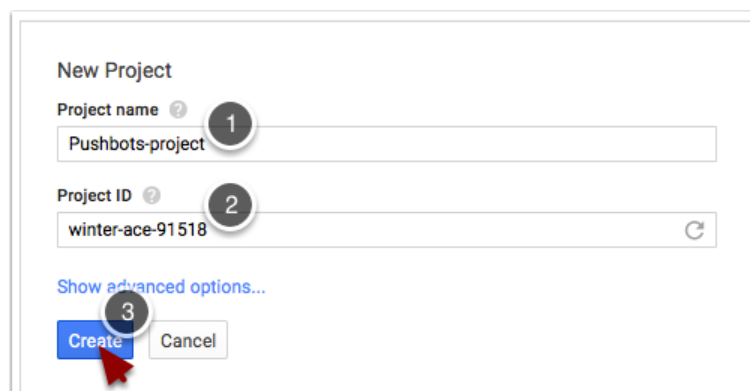


3. Create new project:

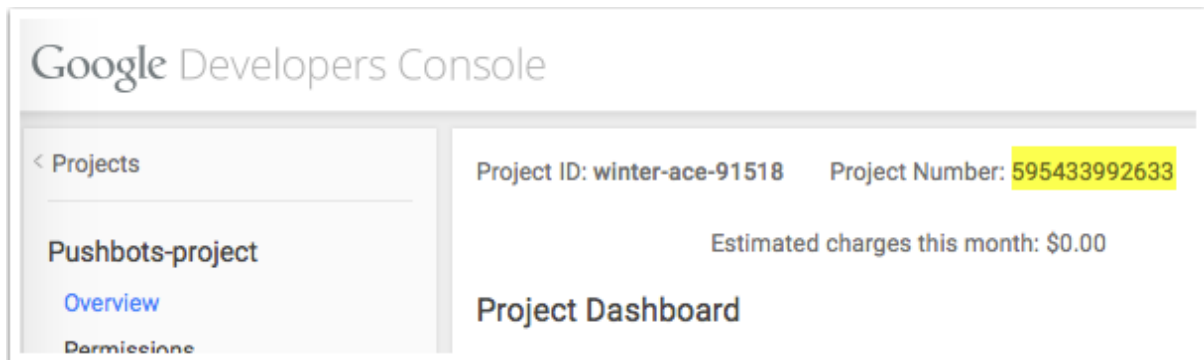
Type project name

The project ID is the globally unique identifier for your project. You cannot change the project ID after the project is created.

click on **Create**.

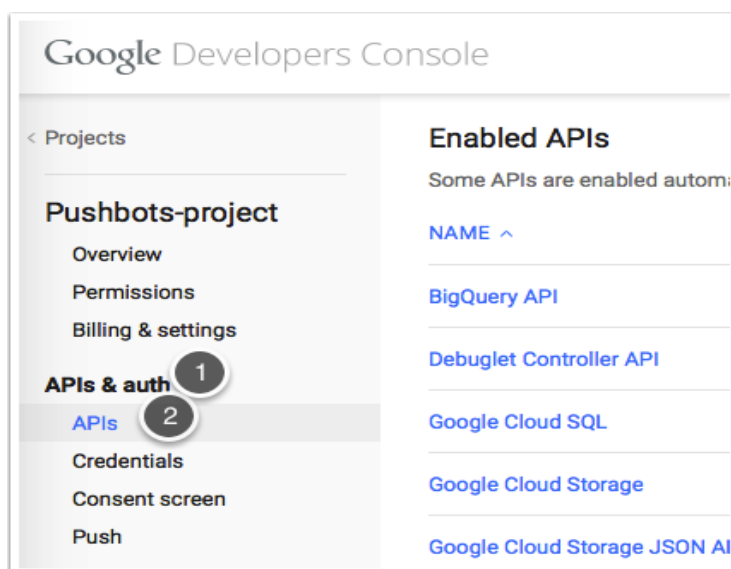


4. Take note of the value of #Project Number: (59093169861 in this example). This is your project number, and it will be used later on as the **GCM sender ID**.

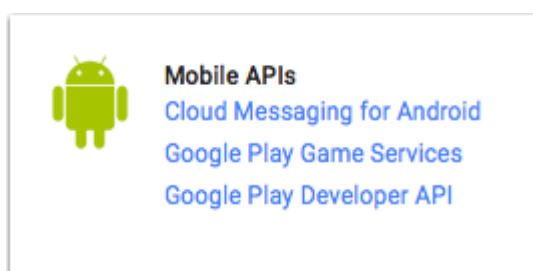


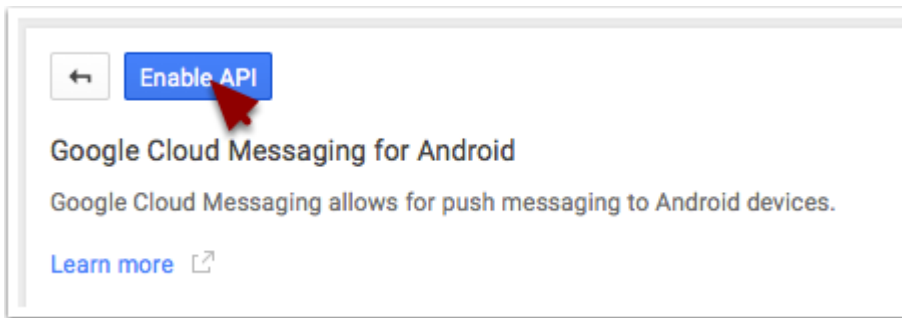
5. Create GCM API Key

1. In the main [Google Developers Console page](#), select **APIs & auth**, then click on **APIs**.



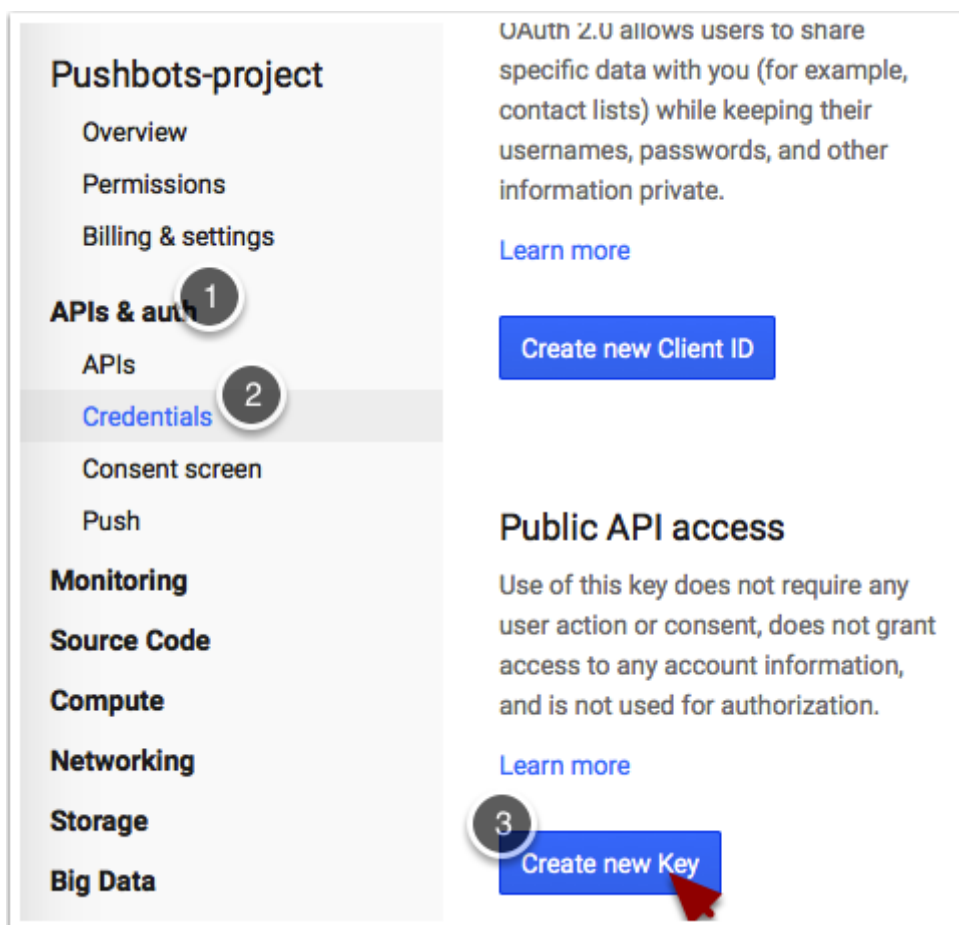
2. Scroll to **Cloud Messaging for Android** and click on it then click on **Enable API**.



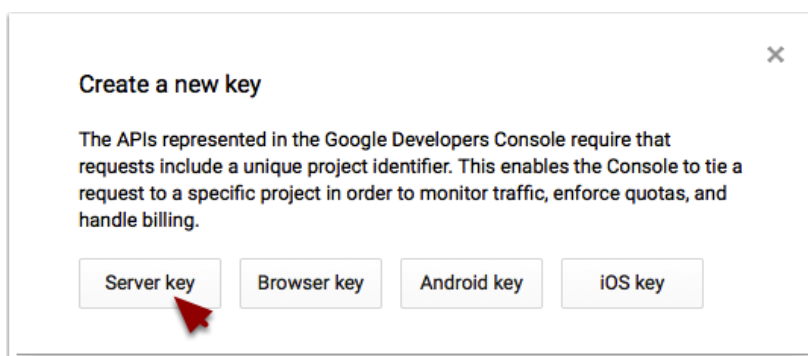


3. In the main **Google APIs Console** page

1. select **APIs & auth** from sidebar.
2. Click on **Credentials**.
3. Click Create **new key**.



4. A new prompt will appear, click on **server key**, then click **Create**



Create a server key and configure allowed IPs

This key should be kept secret on your server.

Every API request is generated by software running on a machine that you control. Per-user limits will be enforced using the address found in each request's `userIp` parameter, (if specified). If the `userIp` parameter is missing, your machine's IP address will be used instead. [Learn more](#)

ACCEPT REQUESTS FROM THESE SERVER IP ADDRESSES

One IP address or subnet per line. Example: 192.168.0.1, 172.16.0.0/16, 2001:db8::1 or 2001:db8::/64

Create

Cancel

- Note down the API Key. This API key will be used when sending requests to GCM server through PushBots.


Public API access

Use of this key does not require any user action or consent, does not grant access to any account information, and is not used for authorization.

[Learn more](#)

Create new Key

Key for server applications

API KEY	AlzaSyCgXaODHWYc2ltleZTgEcFfYe1TgKCEumg
IPS	Any IP allowed
ACTIVATION DATE	Mar 20, 2015, 9:41:00 AM
ACTIVATED BY	 (you)

Edit allowed IPs

Regenerate key

Delete

- Now in **Pushbots.xml** copy the **GCM Project Number** in the place of **PB_Sender ID** and **Pushbots Application ID** in the place of **PB_Appid**.
- Next copy the **API Key** which we generated in the above step in Pushbots > Dashboard > APP Settings > Platforms > Android GCM API Key field.

That's pretty much it.

Run your application now, if all goes well, your device should be registered and ready to receive push notifications. You can also find this tutorial on Pushbots website.

Sending Notifications:

You can Send 3 Types of Notifications:

1. General Notification
2. BigPictureStyle
3. BigTextStyle


Sending the above listed notifications depends on custom fields. Custom Fields are nothing but inputs or attachments to your notification (**Eg.** Picture, Text, Url etc)

General notification:







Just compose a notification by enetering the text in the field provided and Send.

BigPictureStyle: (Custom Fields) :

Sending a picture notification requires custom fields. In the image below to get to know how to add custom fields and send a notification. You can also have a look at the output of picture notification right after the custom fields image.

Custom Data Fields : 

Send additional data with the notification as key-value pair.

customNotificationTitle	Star Wars 7 TV Trailer	
BigPictureStyle	true	
imgUrl	http://okvizag.com/stream/wp-content/uploads/2016/07/star-wars-7-trailer-1080p-hd.mp4	
nextActivity	com.okvizag.ctrplusu.WebViewActivity	
url	http://okvizag.com/stream/star-wars-7-extended-1080p-hd.mp4	
largeIcon	http://okvizag.com/stream/wp-content/uploads/2016/07/star-wars-7-trailer-1080p-hd.mp4	

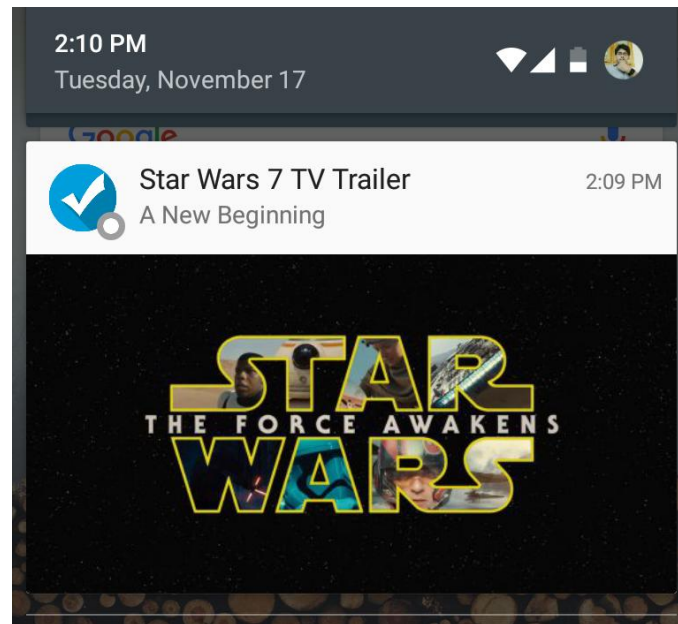


SEND A PUSH



SAVE PRESET

Output:



BigTextStyle: (Custom Fields) :

Sending a notification with 3-4 lines also requires custom fields. In the image below to get to know how to add custom fields for BigTextStyle notification. You can also have a look at the output of bigText notification right after the custom fields image.

Custom Data Fields :

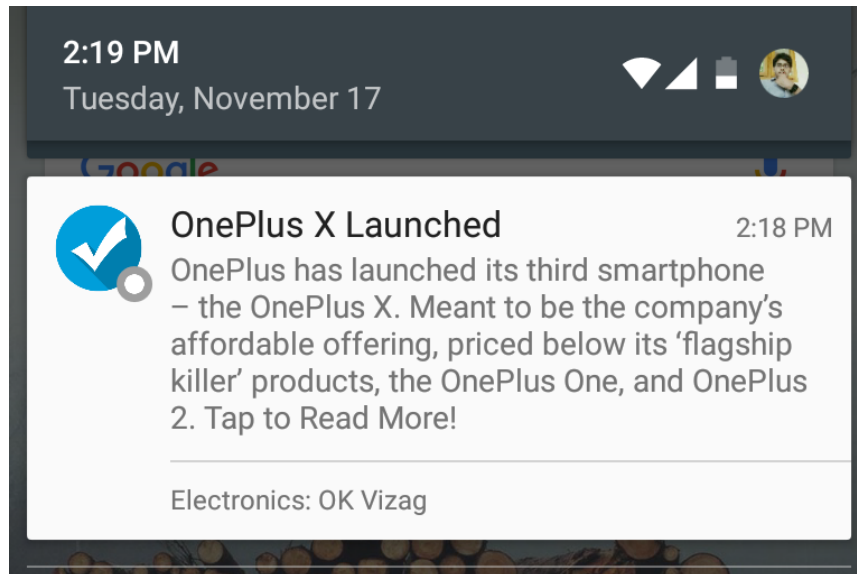
Send additional data with the notification as key-value pair.

customNotificationTitle	OnePlus X Launched	
BigTextStyle	true	
bigText	OnePlus has launched its third smartphone – the	
nextActivity	com.okvizag.ctrlplus.WebViewActivity	
url	http://okvizag.com/stream/oneplus-x-with-5-inch-	
largeIcon	http://okvizag.com/stream/wp-content/uploads/2016/08/oneplus-x-with-5-inch-	
summaryText	Electronics: OK Vizag	

SEND A PUSH

SAVE PRESET

Output:



You can save the custom fields as presets so that from next time you don't need to enter all the fields manually. Simply import the saved presets for "BigPicture" and "BigText" replace the custom fields with new ones and Send.

Export and Publishing on GooglePlay:

Now export your Android project as .apk and upload it on your [Google Play Publishers Dashboard](#). That's it.

The mobile theme which you have seen in the demo app is called "vibrant." It's a mobile only theme, using '[Theme Switcher](#)' Plugin for Wordpress I have set the theme to load only on Mobiles and Tablets. There are no. of themes out there on [Themeforest](#) just browse them all and get the one that best fits your business.

For any further assistance or support do contact us at nkssrikanth@outlook.com.

I have tried my best to explain each and every feature of the app and the implementation procedures, here in this documentation. Hope all your basic queries get solved here. Any assistance or support? kindly [email me](#).

Share love 😊

From:



Nexus and Me