Sending Cloud Messages from the Firebase Console

Mitch Tabian

Twitter: @mitch_tabian
Website: codingwithmitch.com
Youtube: youtube.com/c/mitchtabian

Source Code

“module_3\start\TabianConsulting”

Firebase console

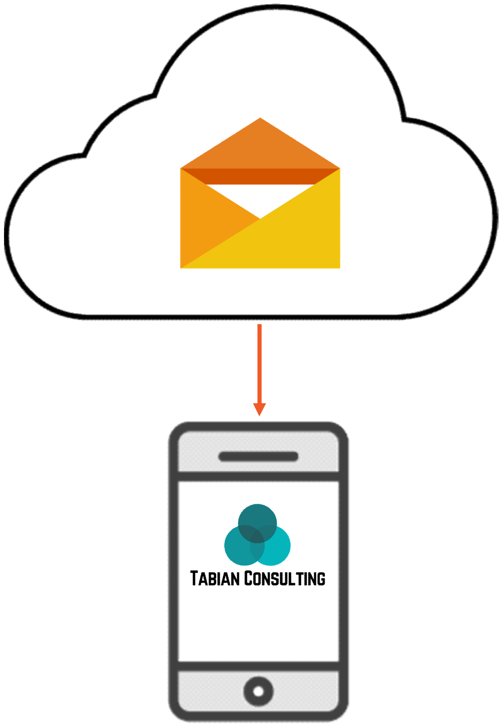
Android client via HTTP

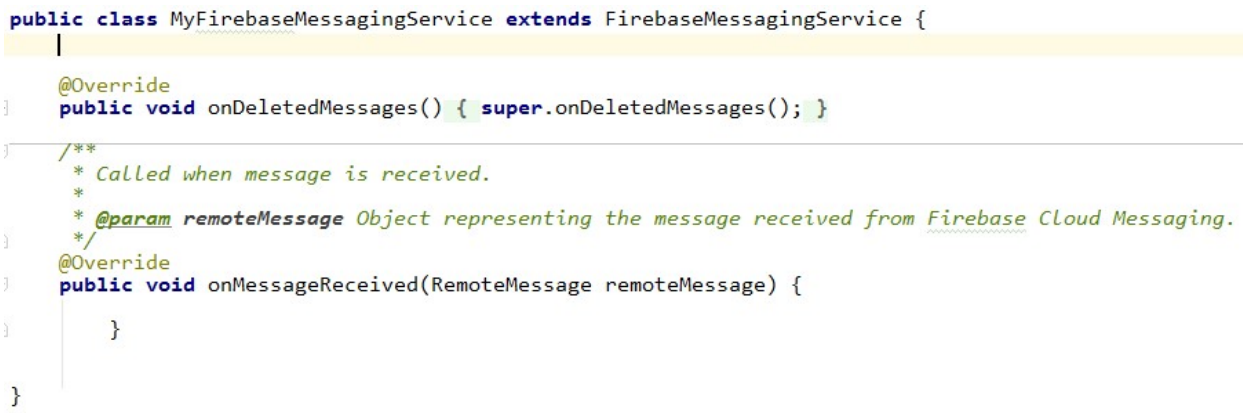
Cloud functions or server-side code

In the foreground

In the background

Closed





```
public class MyFirebaseMessagingService extends FirebaseMessagingService {


    @Override
    public void onDeletedMessages() { super.onDeletedMessages(); }

    /**
     * Called when message is received.
     *
     * @param remoteMessage Object representing the message received from Firebase Cloud Messaging.
     */
    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {

        }


}
```

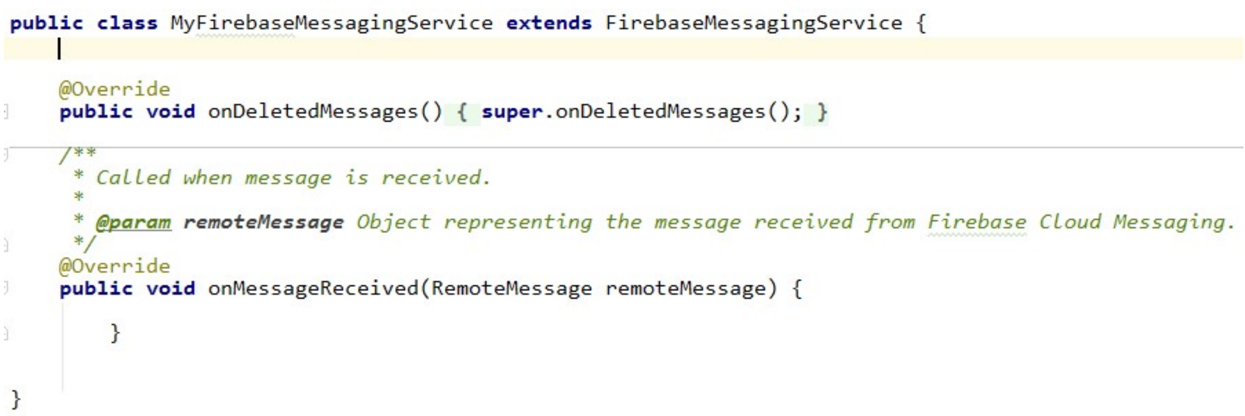
```
{
"to" : "APA91bEat6E1k0gIXXAXuGX6tpEwtFOiSTeYfpLQQslqUEWW41gdCMa",
 "notification" : {
    "body" : "This is a Firebase Cloud Messaging Topic Message!",
    "title" : "FCM Message",
 }
"data" : {
    "data_message" : "This is my data message",
    "data_title" : "This is my data title",
 }
}
```

An Example

```
{
"to" : "APA91bEat6E1k0gIXXAXuGX6tpEwtFOiSTeYfpLQQslqUEWW41gdCMa",
 "notification" : {
   "body" : "This is a Firebase Cloud Messaging Topic Message!",
   "title" : "FCM Message",
 }
"data" : {
   "data_message" : "This is my data message",
   "data_title" : "This is my data title",
 }
}
```

Case 1: Application in Foreground

```
{

"to" :
"APA91bEat6E1k0gIXXAXuGX6tpEwtFOiSTeYfpLQQslqUEWW41gdCMa",

 "notification" : {

   "body" : "This is a Firebase Cloud Messaging Topic
Message!",

   "title" : "FCM Message",

 }

"data" : {

   "data_message" : "This is my data message",

   "data_title" : "This is my data title",

 }

}
```

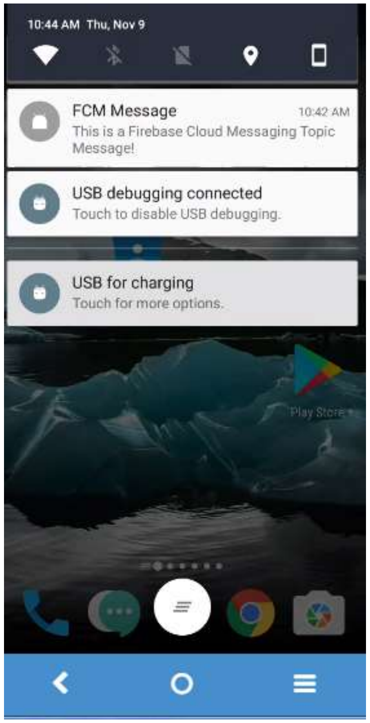


```
public class MyFirebaseMessagingService extends FirebaseMessagingService {

    @Override
    public void onDeletedMessages() { super.onDeletedMessages(); }

    /**
     * Called when message is received.
     *
     * @param remoteMessage Object representing the message received from Firebase Cloud Messaging.
     */
    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {

    }

}
```

```
remoteMessage.getdata()

remoteMessage.getNotification()
```

Case 2: Application in Background

```
{
"to" :
"APA91bEat6E1k0gIXXAXuGX6tpEwtFOiSTeYfpLQQslqUEWW41gdCMa",
 "notification" : {
   "body" : "This is a Firebase Cloud Messaging Topic
Message!",
   "title" : "FCM Message",
 }
"data" : {
   "data_message" : "This is my data message",
   "data_title" : "This is my data title",
 }
}
```



10:44 AM Thu, Nov 9

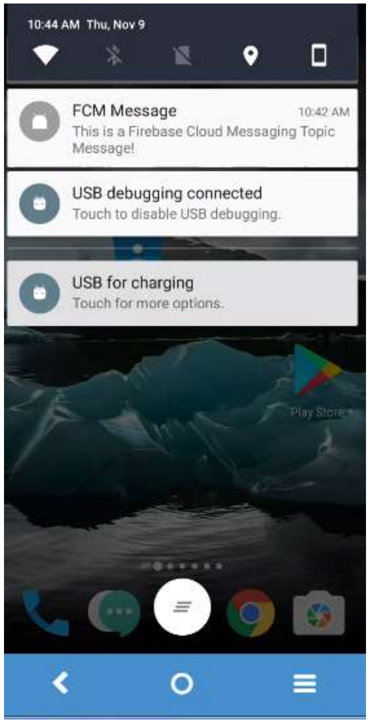
FCM Message 10:42 AM
This is a Firebase Cloud Messaging Topic Message!

USB debugging connected
Touch to disable USB debugging.

USB for charging
Touch for more options.

Case 3: Application is Closed

```
{

"to" :
"APA91bEat6E1k0gIXXAXuGX6tpEwtFOiSTeYfpLQQslqUEWW41gdCMa",

 "notification" : {

   "body" : "This is a Firebase Cloud Messaging Topic
Message!",

   "title" : "FCM Message",

 }

"data" : {

   "data_message" : "This is my data message",

   "data_title" : "This is my data title",

 }

}
```



10:44 AM Thu, Nov 9

FCM Message 10:42 AM
This is a Firebase Cloud Messaging Topic Message!

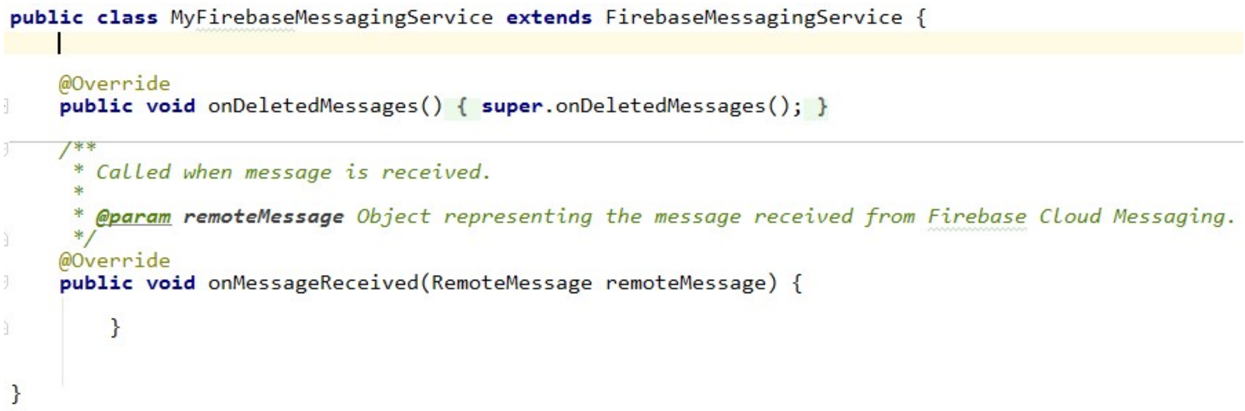
USB debugging connected
Touch to disable USB debugging.

USB for charging
Touch for more options.

Summary of the Scenarios

A Better Way

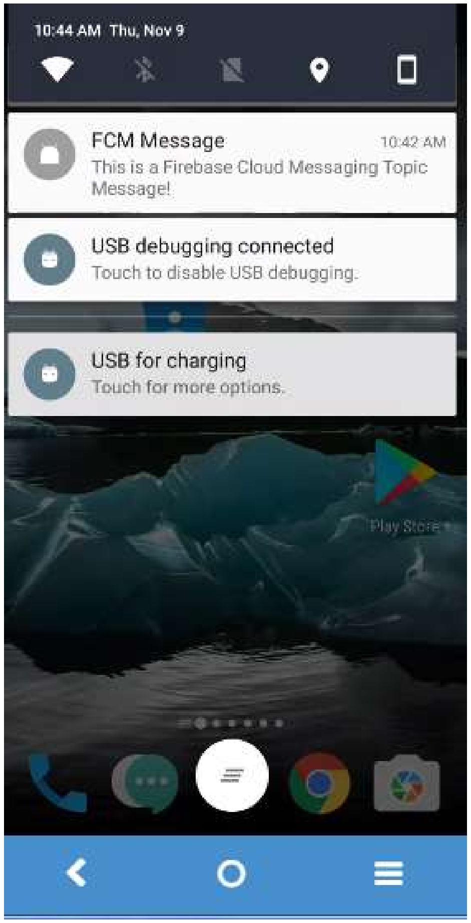
```
{

"to" :
"APA91bEat6E1k0gIXXAXuGX6tpEwtFOiSTeYfpLQQslqUEWW41gdCMa"
,

"data" : {

   "message" : "This is my data message",

   "title" : "This is my data title",

 }

}
```



```
public class MyFirebaseMessagingService extends FirebaseMessagingService {

    @Override
    public void onDeletedMessages() { super.onDeletedMessages(); }

    /**
     * Called when message is received.
     *
     * @param remoteMessage Object representing the message received from Firebase Cloud Messaging.
     */
    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {

        }

}
```

`remoteMessage.getdata()`

Which is better?

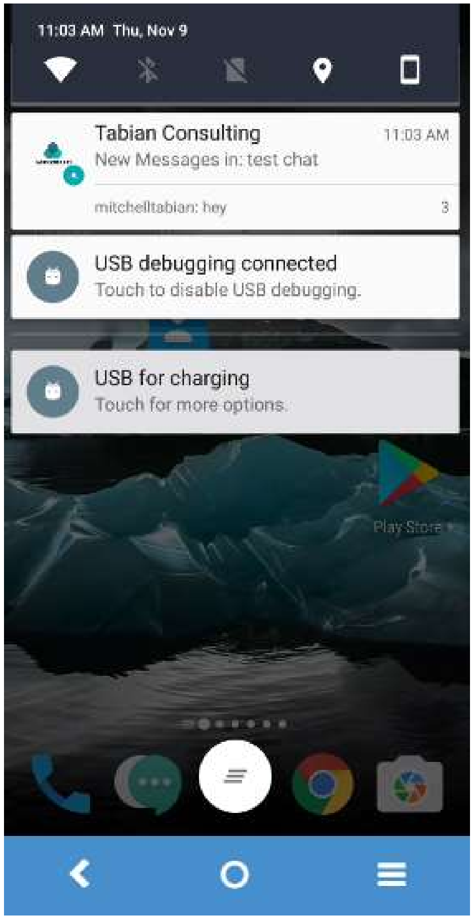


10:44 AM Thu, Nov 9

FCM Message 10:42 AM
This is a Firebase Cloud Messaging Topic Message!

USB debugging connected
Touch to disable USB debugging.

USB for charging
Touch for more options.



11:03 AM Thu, Nov 9

Tabian Consulting 11:03 AM
New Messages in: test chat
mitchelltabian: hey 3

USB debugging connected
Touch to disable USB debugging.

USB for charging
Touch for more options.

Implementing the Firebase Messaging ID Service

Implementing the Firebase Messaging Service

Module Review

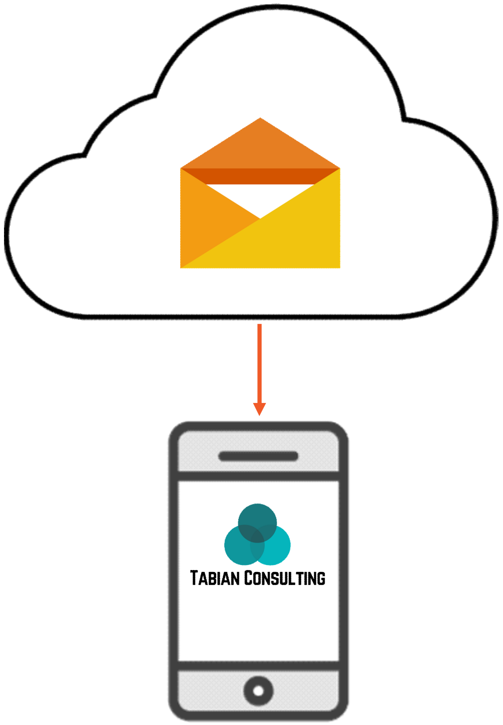
In the foreground

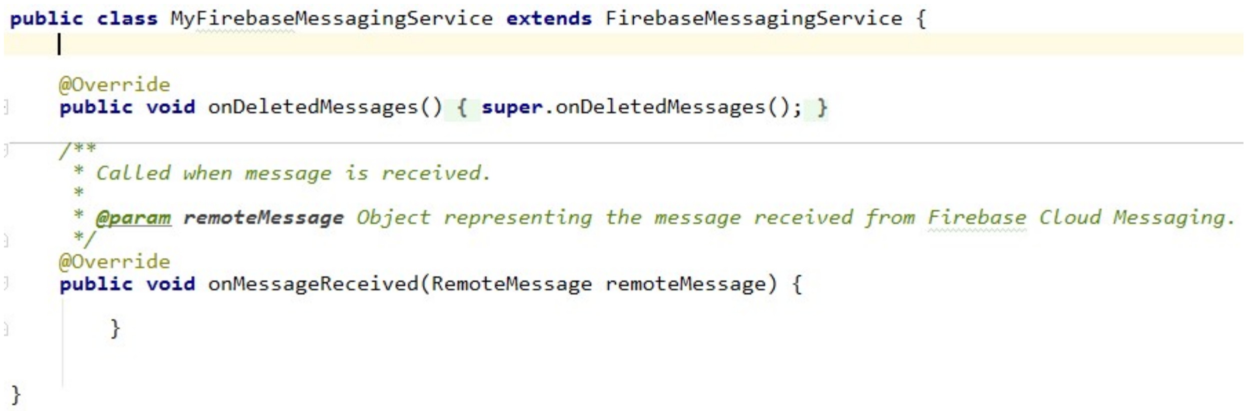
In the background

Closed

```
{
 "to" : "<Token or Topic>",
 "notification" : {
   "body" : "This is a Firebase Cloud Messaging Topic Message!",
   "title" : "FCM Message",
 }
"data" : {
   "my_custom_key1" : "Something custom I want to send",
   "my_custom_key2" : "Another thing I want to send",
 }
}
```

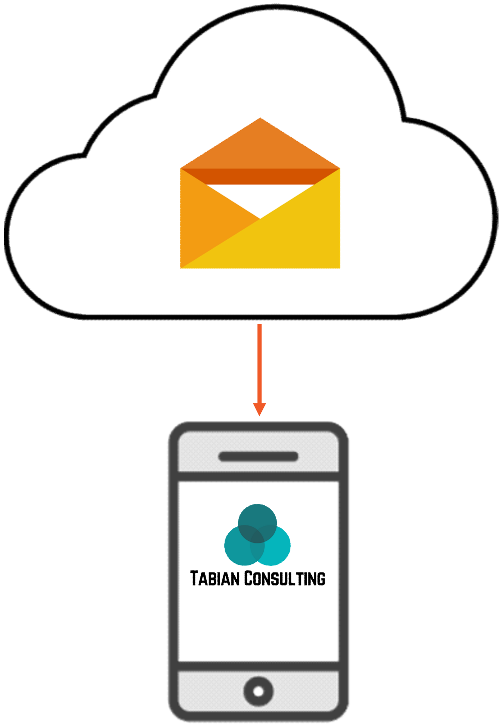
```
{
"to" : "APA91bEat6E1k0gIXXAXuGX6tpEwtFOiSTeYfpLQQslqUEWW41gdCMa",
"data" : {
   "message" : "This is my data message",
   "title" : "This is my data title",
 }
}
```



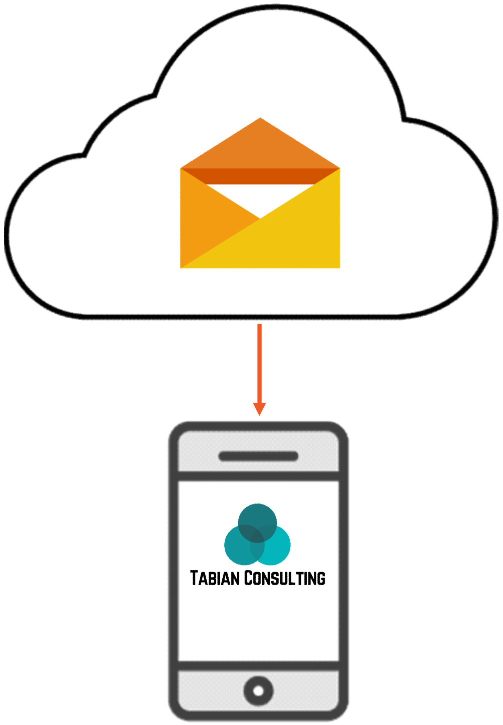


```java
public class MyFirebaseMessagingService extends FirebaseMessagingService {
    |

    @Override
    public void onDeletedMessages() { super.onDeletedMessages(); }

    /**
     * Called when message is received.
     *
     * @param remoteMessage Object representing the message received from Firebase Cloud Messaging.
     */
    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {

        }

}
```

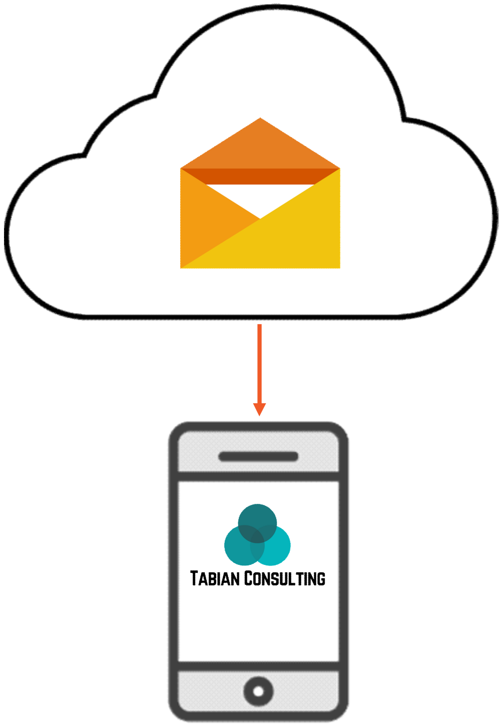
```xml
<service
 android:name=".utility.MyFirebaseMessagingService">
 <intent-filter>
    <action
android:name="com.google.firebase.MESSAGING_EVENT"/>
 </intent-filter>
</service>
```



```java
public class MyFirebaseInstanceIDService extends FirebaseInstanceIdService
{

    private static final String TAG = "MyFirebaseIIDService";

    @Override
    public void onTokenRefresh() {
        // Get updated InstanceID token.
        String refreshedToken =
          FirebaseInstanceId.getInstance().getToken();
        Log.d(TAG, "Refreshed token: " + refreshedToken);

        //send token to server
        sendRegistrationToServer(refreshedToken);
    }
}
```



```
<service
    android:name=".utility.MyFirebaseInstanceIDService">
    <intent-filter>
        <action
        android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
    </intent-filter>
</service>
```



```
private void sendRegistrationToServer(String token) {
    Log.d(TAG, "sendRegistrationToServer: sending token to server: " +
      token);
    DatabaseReference reference =
      FirebaseDatabase.getInstance().getReference();
      reference.child(getString(R.string.dbnode_users))
            .child(FirebaseAuth.getInstance().getCurrentUser().getUid())
            .child(getString(R.string.field_messaging_token))
            .setValue(token);
}
```