Sending Cloud Messages from the Android Client

Mitch Tabian

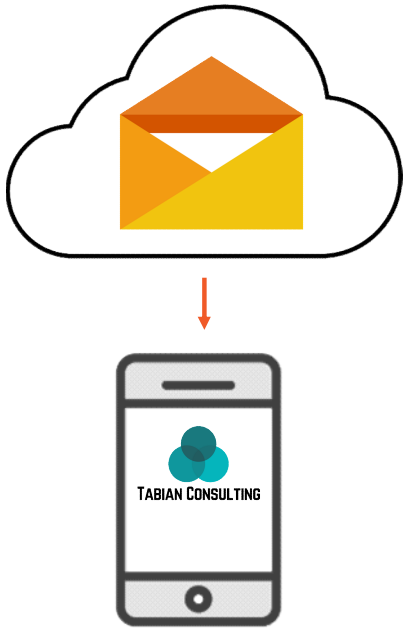
Twitter: @mitch_tabian
Website: codingwithmitch.com
Youtube: youtube.com/c/mitchtabian

Sending Cloud Messages from the Android Client

Source Code

“module_4\start\TabianConsulting”

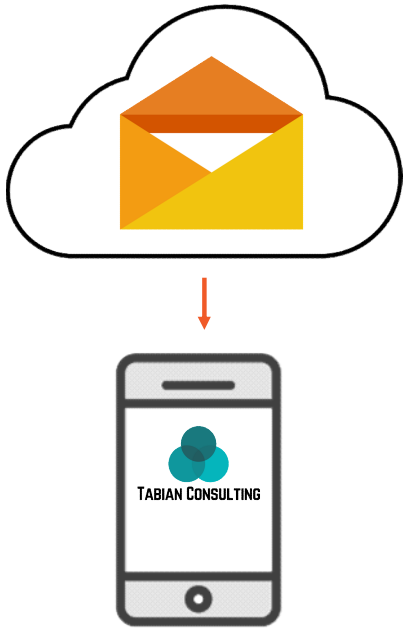
Sending Cloud Messages from Android Client



https://fcm.googleapis.com/fcm/send

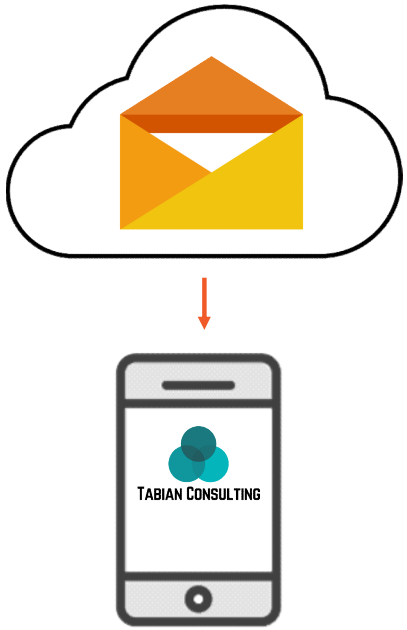
Retrofit

Sending Cloud Messages from Android Client




```
{
"to" : "<user FCM token>",
"data" : {
   "message" : "This is my data message",
   "title" : "This is my data title",
   “data_type" : “data_type_admin_broadcast",
 }
}
```

Sending Cloud Messages from Android Client




```
Content-Type:application/json
Authorization:key=<Server Key>
```

Sending Cloud Messages from Android Client

Retrieving the FCM Tokens and Server Key

Getting Started with Retrofit

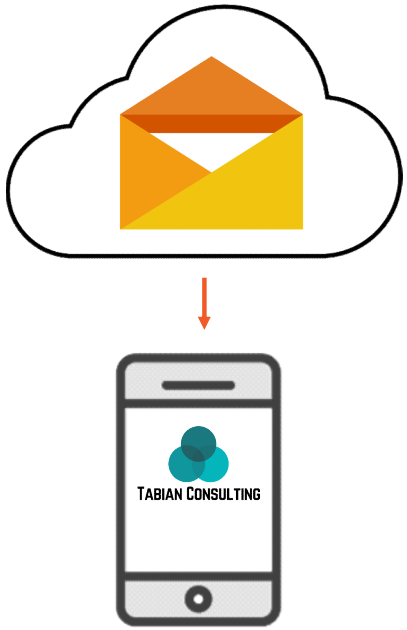
Testing HTTP Requests with Postman

Sending a Cloud Message from the Android Client

Cloud Messaging with Topics and Device Groups

Module Review

Sending Cloud Messages from Android Client



https://fcm.googleapis.com/fcm/send

```
Content-Type:application/json
Authorization:key=<Server Key>
{
"to" : "<user FCM token>",
"data" : {
   "message" : "This is my data message",
   "title" : "This is my data title",
   “data_type" : “data_type_admin_broadcast",
 }
}
```

Retrofit

HTTP requests

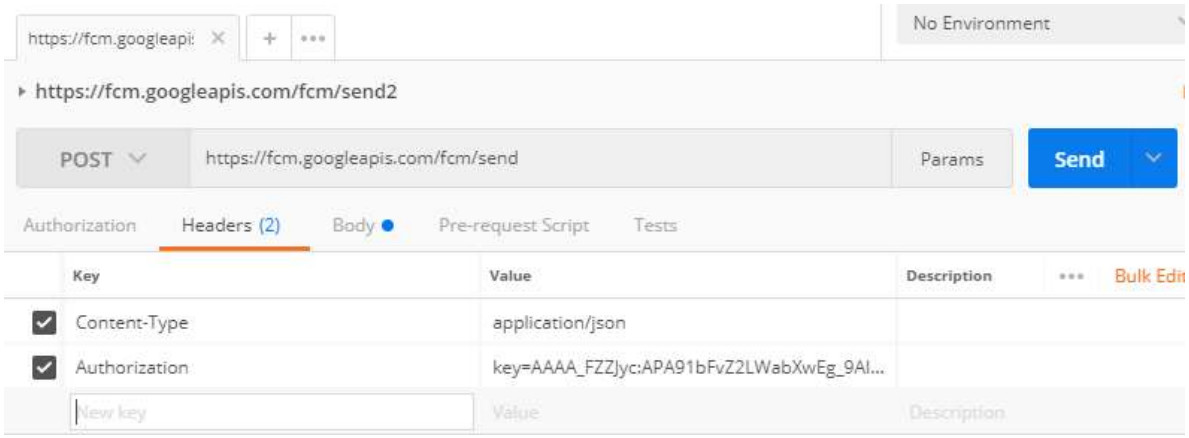
- JSON
- XML
- JACKSON

Postman

Test the request

Review the data structure

Postman


https://fcm.googleapi
No Environment
https://fcm.googleapis.com/fcm/send2
POST
https://fcm.googleapis.com/fcm/send
Params
Send
Authorization
Headers (2)
Body
Pre-request Script
Tests
Key
Value
Description
Bulk Edit
Content-Type
application/json
Authorization
key=AAAA_FZZJyc:APA91bFvZ2LWabXwEg_9AI...
New key
Value
Description

Postman

https://fcm.googleapis.com/fcm/send2

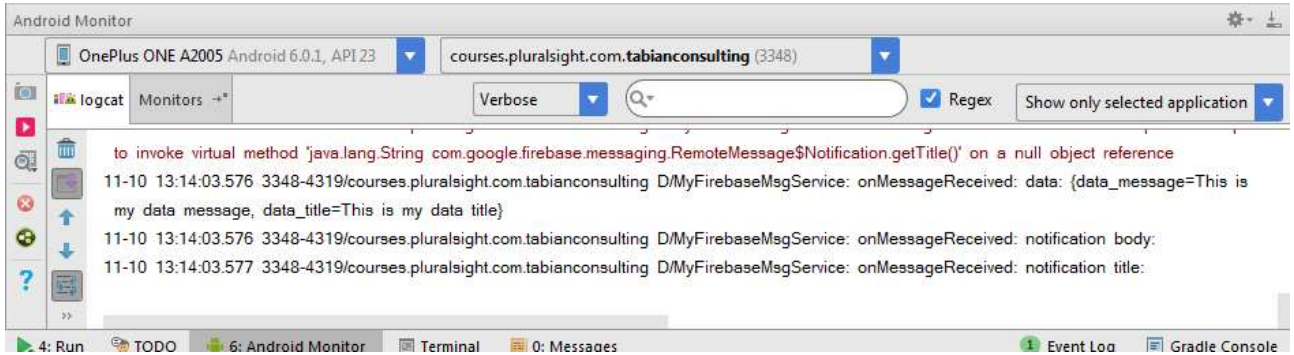
POST https://fcm.googleapis.com/fcm/send Params **Send**

Authorization Headers (2) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
{
  "to" : "dJOBuyL2H1o:APA91bFtjfWW7up-No2OtOfzKIYVg9kBwE0cDWyGw6NCy1ux12SfELAPg_-wFzX93uigVdnz7kqBHs2H2N3Q1Juf
     -69yVxRvu41DDjFvDZHJ4pewwy2urgrcwy1XFdyn7sA9Wk5WnB8L",
 "data" : {
    "message" : "This is my data message",
    "title" : "This is my data title",
    "data_type": "data_type_admin_broadcast"
  }
}
```

Postman


Android Monitor
OnePlus ONE A2005 Android 6.0.1, API 23
courses.pluralsight.com.tabianconsulting (3348)
logcat
Monitors
Verbose
Regex
Show only selected application
to invoke virtual method 'java.lang.String com.google.firebase.messaging.RemoteMessage$Notification.getTitle()' on a null object reference
11-10 13:14:03.576 3348-4319/courses.pluralsight.com.tabianconsulting D/MyFirebaseMsgService: onMessageReceived: data: {data_message=This is my data message, data_title=This is my data title}
11-10 13:14:03.576 3348-4319/courses.pluralsight.com.tabianconsulting D/MyFirebaseMsgService: onMessageReceived: notification body:
11-10 13:14:03.577 3348-4319/courses.pluralsight.com.tabianconsulting D/MyFirebaseMsgService: onMessageReceived: notification title:
4: Run
TODO
6: Android Monitor
Terminal
0: Messages
Event Log
Gradle Console

Data.java

```
public class Data {

  private String title;
  private String message;
  private String data_type;

  public Data(String title, String message, String data_type) {
     this.title = title;
     this.message = message;
     this.data_type = data_type;
  }
  public Data() {

  }
  public String getData_type() {
     return data_type;
  }

  public void setData_type(String data_type) {
     this.data_type = data_type;
  }
  public String getTitle() {
     return title;
  }
  public void setTitle(String title) {
     this.title = title;
  }
  public String getMessage() {
     return message;
  }
  public void setMessage(String message) {
     this.message = message;
  }
}
```

FirebaseCloudMessage.java

```
public class FirebaseCloudMessage {

  private String to;
  private Data data;

  public FirebaseCloudMessage(String to, Data data) {
     this.to = to;
     this.data = data;
  }

  public FirebaseCloudMessage() {

  }

  public String getTo() {
     return to;
  }

  public void setTo(String to) {
     this.to = to;
  }


  public Data getData() {
     return data;
  }

  public void setData(Data data) {
     this.data = data;
  }
}
```

Retrofit Request

```
Retrofit retrofit = new Retrofit.Builder()
        .baseUrl(BASE_URL)
        .addConverterFactory(GsonConverterFactory.create())
        .build();

//create the interface
FCM fcmAPI = retrofit.create(FCM.class);

//attach the headers
HashMap<String, String> headers = new HashMap<>();
headers.put("Content-Type", "application/json");
headers.put("Authorization", "key=" + mServerKey);

//send the message to all the tokens
for(String token : mTokens){

    Log.d(TAG, "sendMessageToDepartment: sending to token: " + token);
    Data data = new Data();
    data.setMessage(message);
    data.setTitle(title);
    data.setData_type(getString(R.string.data_type_admin_broadcast));
    FirebaseCloudMessage firebaseCloudMessage = new FirebaseCloudMessage();
    firebaseCloudMessage.setData(data);
    firebaseCloudMessage.setTo(token);

    Call<ResponseBody> call = fcmAPI.send(headers, firebaseCloudMessage);

    call.enqueue(new Callback<ResponseBody>() {
        @Override
        public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {
            Log.d(TAG, "onResponse: Server Response: "  + response.toString());

        }

        @Override
        public void onFailure(Call<ResponseBody> call, Throwable t) {
            Log.e(TAG, "onFailure: Unable to send the message." + t.getMessage() );
            Toast.makeText(AdminActivity.this, "error", Toast.LENGTH_SHORT).show();
        }
    });
}
```

Testing the Request