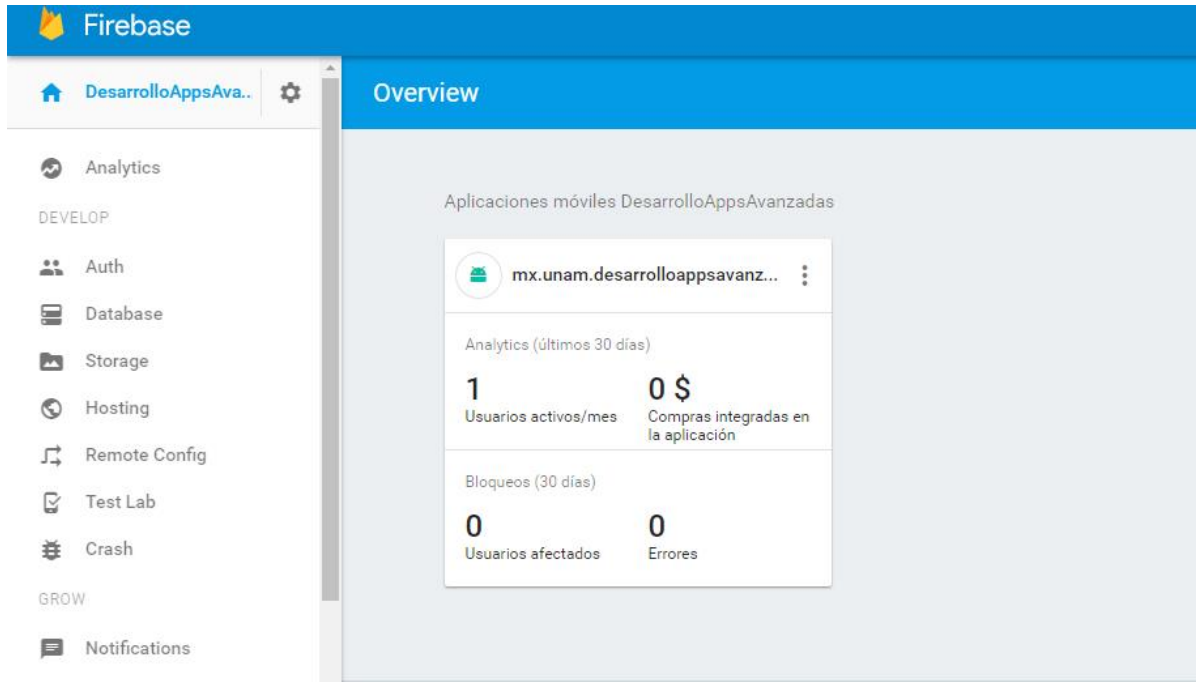
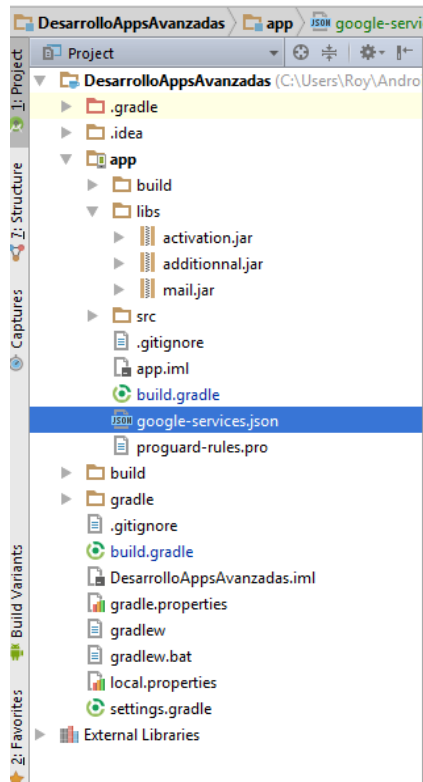


# Assignment: Configurando mi propio Endpoint en mi servidor.

- 1) Toma como base tu proyecto Petagram y registralo en Firebase de Google <https://firebase.google.com/>



- 2) Configura tu aplicación para almacenar el archivo con extensión google-services.json.



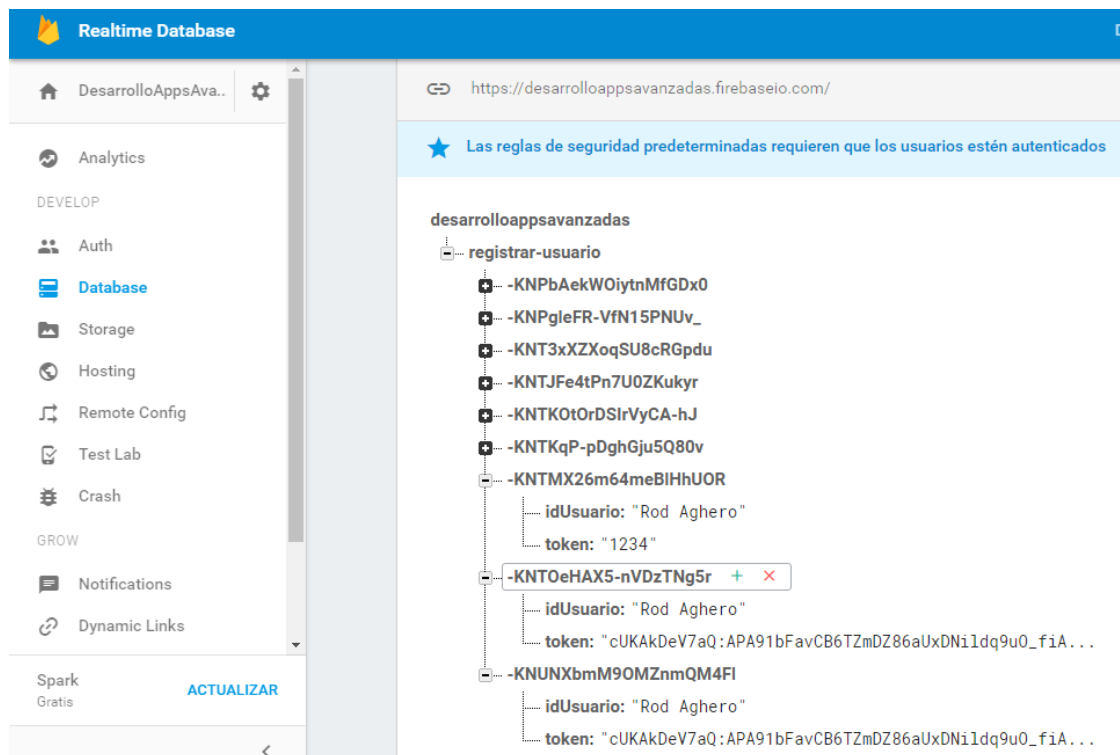
3) Con ayuda de Heroku selecciona tu lenguaje de programación favorito y:

Crea un Web Service que tenga como Endpoint el servicio de recibir el **identificador del dispositivo (token)** y el **id de usuario** que se configuró en la cuenta de la aplicación (esto lo hiciste en la actividad anterior). La finalidad es que estos sean almacenados en una base de datos, la cual, debe estar preparada para insertar los datos permanentemente. Puedes tomar como referencia el siguiente modelo:

#### Modelo Base de Datos WS

usuario_instagram
id_dispositivo
id_usuario_instagram

En esta imagen se observan los datos recibidos a partir de Android Studio (los últimos 2), el token (id\_dispositivo) y el id\_usuario de Instagram. También aparece un id para cada conjunto de datos enviado.

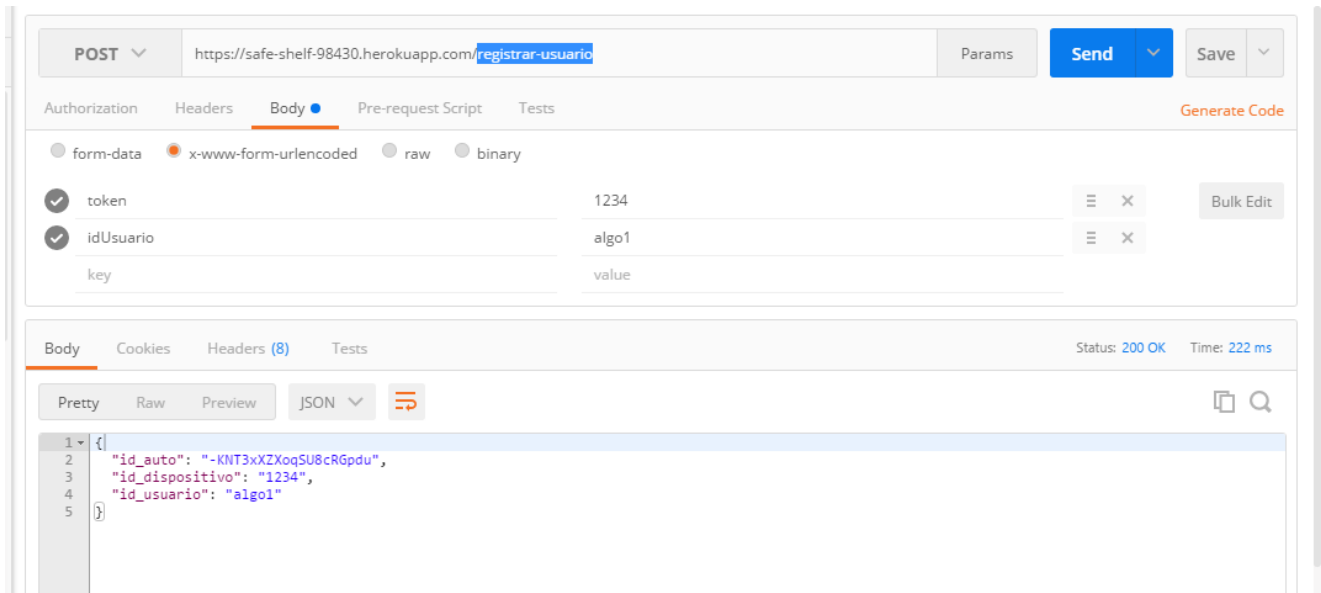


4) El Endpoint debe llamarse: **/registrar-usuario**

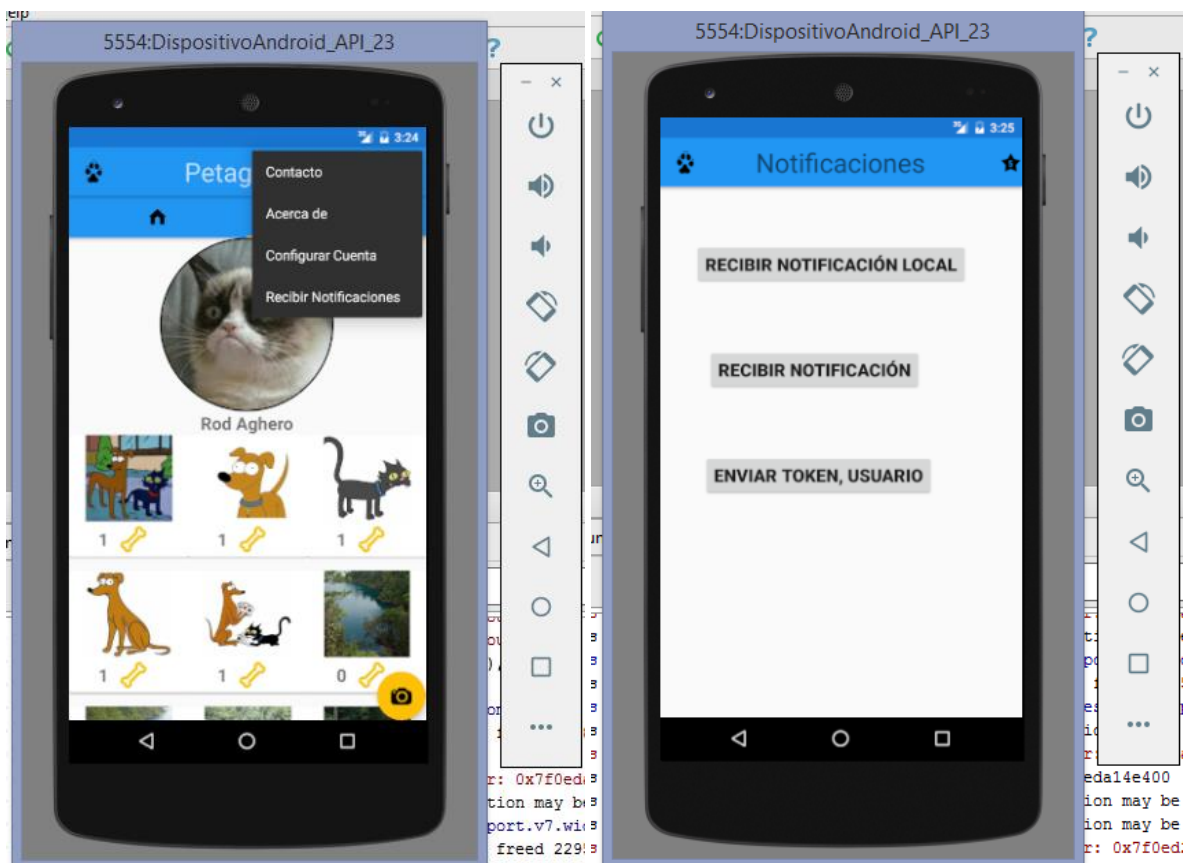
Como parámetros debe recibir:

- id\_dispositivo
- id\_usuario\_instagram

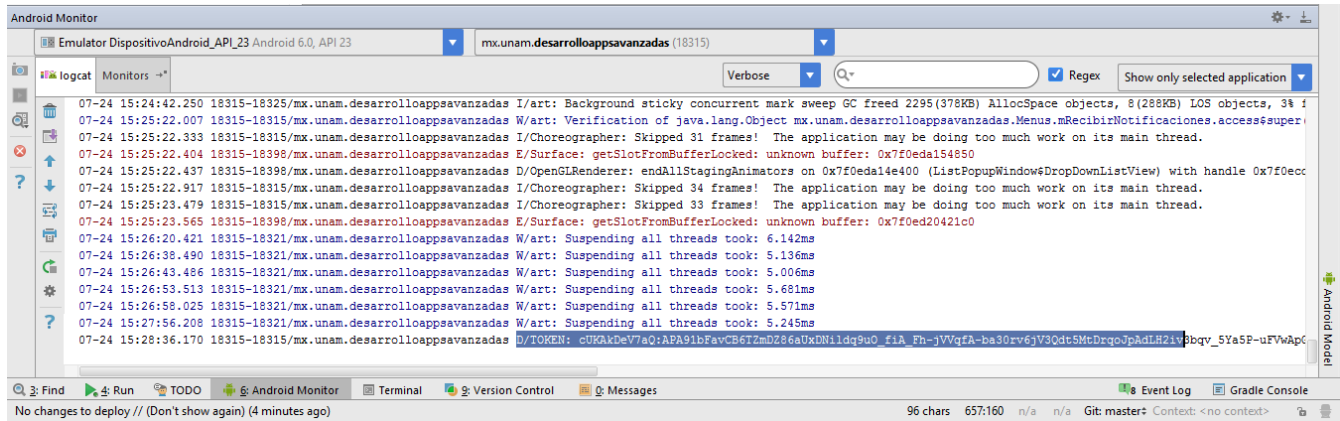
En esta imagen se muestra la prueba con Postman con el endpoint **registrar-usuario**, recibiendo los dos parámetros solicitados. En la imagen anterior se observa en la antepenúltima posición la recepción de estos datos.



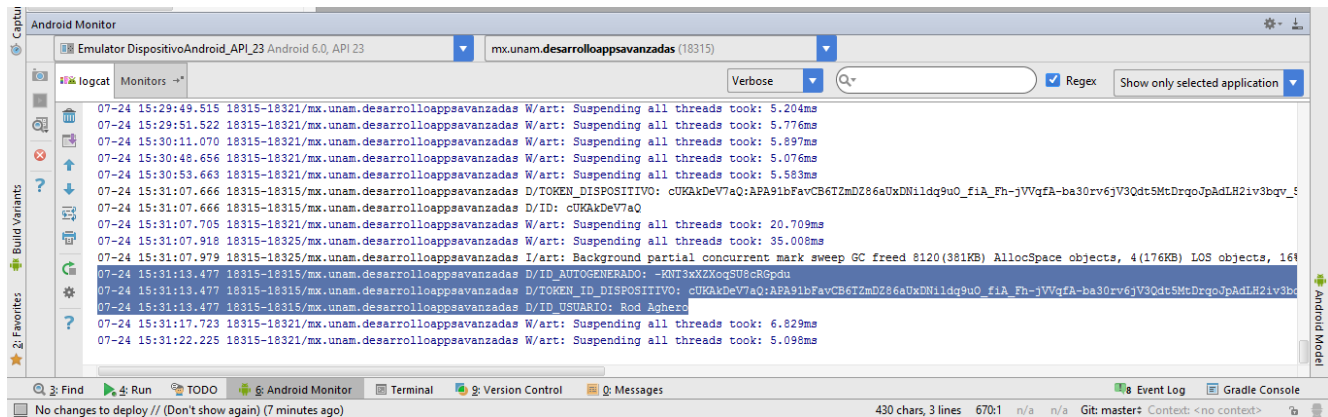
- 5) Implementa un nuevo ítem en tu menú de Opciones, "Recibir Notificaciones" como se muestra en la maqueta, el cual enviará los datos a GCM retornando tu Token o id de dispositivo. Haz las validaciones necesarias.



En las imágenes del emulador se observa la implementación del menú **Recibir Notificaciones**, dentro del cual aparecen 3 botones, el primero para la notificación local, el segundo para el token de dispositivo y el tercero para obtener el token y el usuario. Tuve algunos problemas para que saliera el token en la terminal por lo que replique paso a paso la lección. Al final fue un tema de actualización de los Google APIs que solicitaban la rev 15 en el SDK Manager.

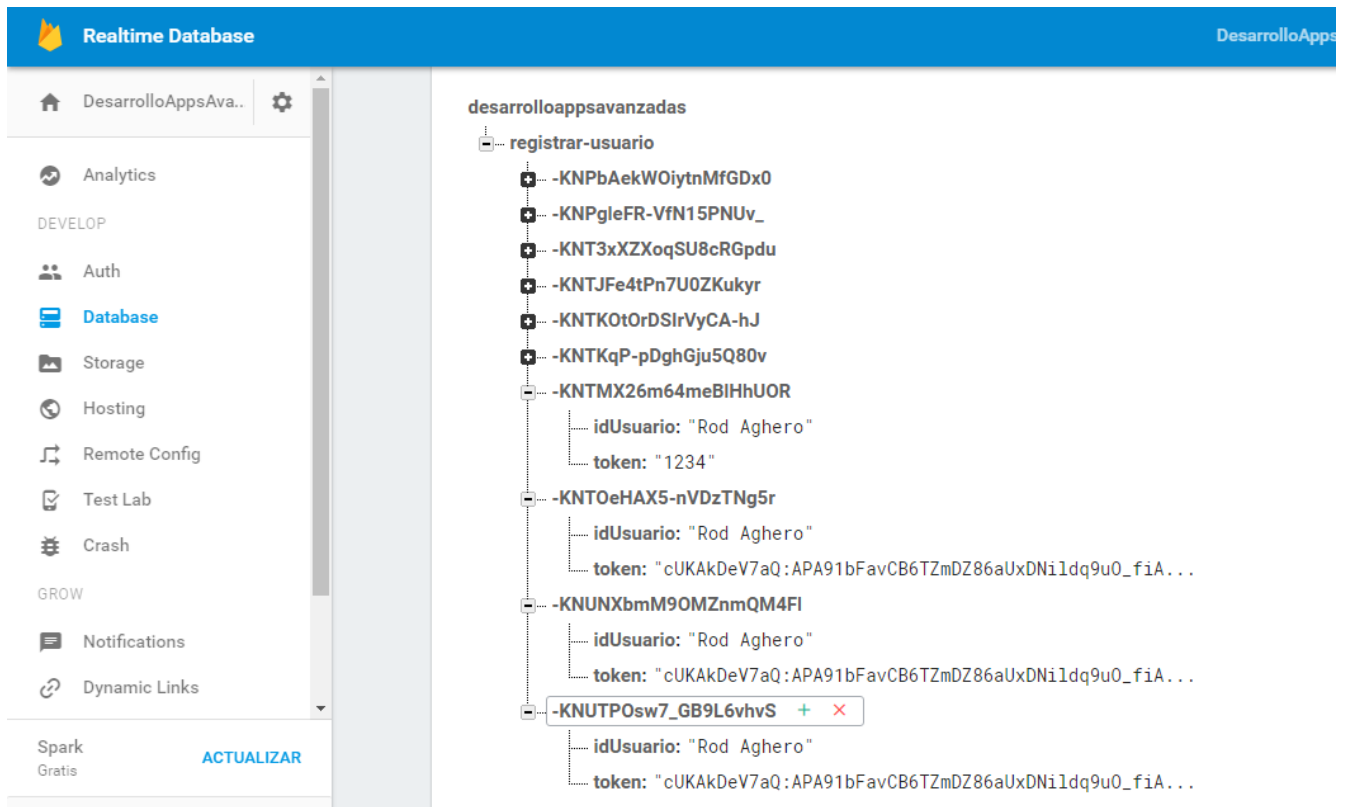


- 6) Una vez que recibas el id de tu dispositivo (token), envíalo junto con el id de usuario de la cuenta de Instagram seleccionada en la aplicación a tu Endpoint que configuraste.

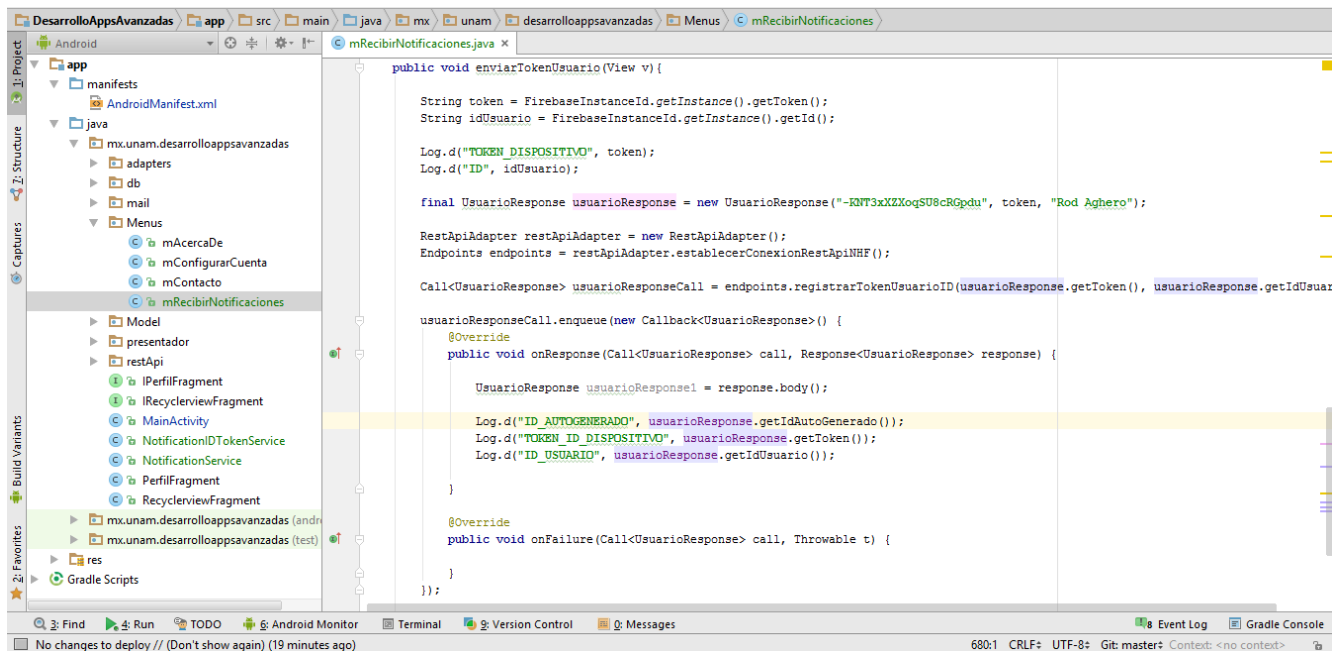


En éste último paso tengo una situación, al dar en el botón para obtener token con un **usuarioResponse1** vacío el proyecto se cierra, por lo que pasé los datos directo del **final UsuarioResponse** que utilicé para los parámetro del método del **endpoint.registrarTokenUsuarioID**.

Al hacer esto la base de datos de firebase recibe sin problema los datos, pero desconozco por qué no me deja usar un usuario vacío para recibir el **id\_autogenerado** de dichos datos.



Dejo el screenshot del método del botón y el index.js, aunque no creo que el error este por ahí dado que la base de datos si funciona.



```
File Edit View Selection Find Packages Help
node.js-getting-started index.js — Petagram/server/node.js-getting-: Petagram-dc57f09b72d3.json index.js — DesarrolloAppsAvanzadas Heroku! DesarrolloAppsAvanzadas-73837671a097.json
> .git
> node_modules
> public
> views
.env
.gitignore
app.json
index.js
package.json
Petagram-dc57f09b72d3.json
Profile
README.md
1 var express = require('express');
2 var app = express();
3
4 app.set('port', (process.env.PORT || 5000));
5
6 var bodyParser = require("body-parser");
7 app.use(bodyParser.json());
8 app.use(bodyParser.urlencoded( { extended:true } ));
9
10 var firebase = require("firebase");
11 firebase.initializeApp({
12   serviceAccount: "DesarrolloAppsAvanzadas-73837671a097.json",
13   databaseURL: "https://desarrolloappsavanzadas.firebaseio.com"
14 });
15
16 app.use(express.static(__dirname + '/public'));
17
18 // views is directory for all template files
19 app.set('views', __dirname + '/views');
20 app.set('view engine', 'ejs');
21
22 app.get('/android', function(request, response) {
23   response.render('pages/index');
24 });
25
26 // Método post
27 // https://sgfe-shelf-98430.herokuapp.com/
28 var tokenDevicesURI = "registrar-usuario";
29 app.post("/") + tokenDevicesURI, function(request, response) {
30
31   var token = request.body.token; // id_dispositivo
32   var idUsuario = request.body.idUsuario; // id_usuario
33
34   var db = firebase.database();
35   var tokenDevices = db.ref("/registrar-usuario").push(); // para no sobre-escribir
36   tokenDevices.set({
37     token: token,
38     idUsuario: idUsuario
39   });
40
41   var path = tokenDevices.toString();
42   var pathSplit = path.split(tokenDevicesURI + "/");
43   var idAutoGenerado = pathSplit[1];
44
45   var respuesta = generarRespuestaAToken(db, idAutoGenerado);
46   response.setHeader("Content-Type", "application/json");
47   response.send(JSON.stringify(respuesta));
48
49   function generarRespuestaAToken(db, idAutoGenerado) {
50     var respuesta = {};
51     var usuario = "";
52     var ref = db.ref(tokenDevicesURI);
53     ref.on("child_added", function(snapshot, prevChildKey) {
54       usuario = snapshot.val();
55
56       respuesta = {
57         id_auto: idAutoGenerado,
58         id_dispositivo: usuario.token,
59         id_usuario: usuario.idUsuario
60       };
61     });
62     return respuesta;
63   }
64
65   app.listen(app.get('port'), function() {
66     console.log('Node app is running on port', app.get('port'));
67   });
68
69
70
71
72
```

Agradeceré sus comentarios al respecto.