

Creando, Recibiendo y Actuando Push Notifications

Autor: German Echaiz Espinoza

Endpoints

```
public interface EndpointsFirebase {

    @FormUrlEncoded
    @POST(ConstantsRestApiFirebase.KEY_POST_ID_TOKEN)
    Call<UsuarioResponse> registrarTokenID(@Field("id_dispositivo") String
token, @Field("id_usuario_instagram") String insta ,
@Field("id_foto_instagram") String likes);

    @GET(ConstantsRestApiFirebase.KEY_LIKE_MASCOTA)
    Call<UsuarioResponse> likeMascota(@Path("id") String id, @Path("nombre")
String nombre);
}
```

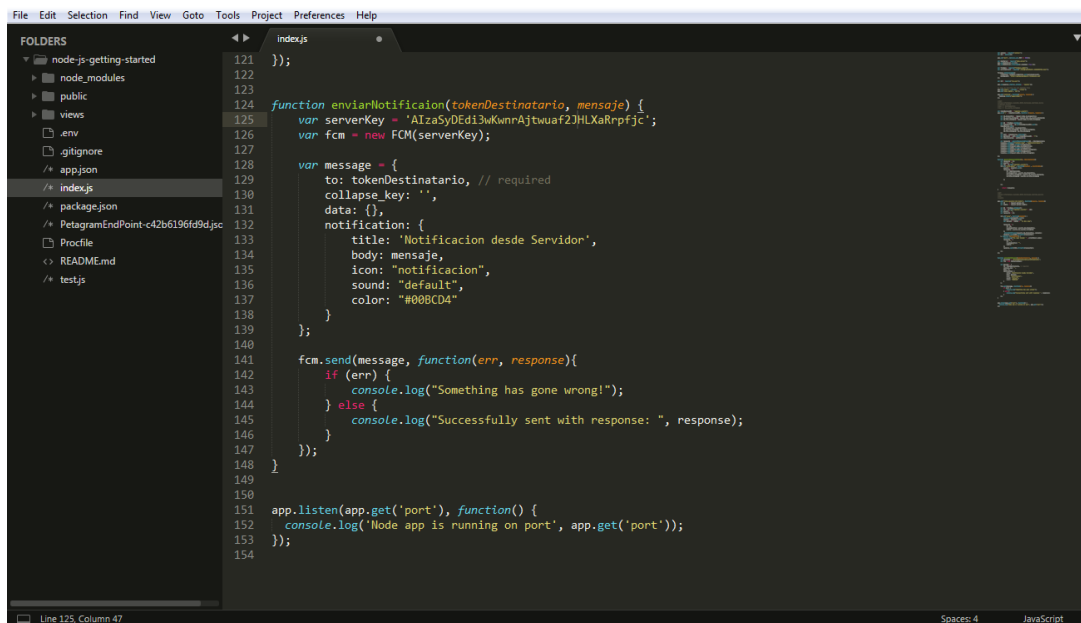
Constantes Restapi

```
public static final String ROOT_URL = "https://fathomless-lowlands-88456.herokuapp.com/";
public static final String KEY_POST_ID_TOKEN = "registar-usuario/";
public static final String KEY_LIKE_MASCOTA = "like-mascota/{id}/{nombre}";
```

Modificaciones al archivo index.js

```
FOLDERS | node.js-getting-started
  node_modules
  public
  views
  .env
  gitignore
  app.json
  index.js
  package.json
  PetagramEndPoint-c42b6196fd9d.js
  Profile
  README.md
  test.js

index.js
85 //GET
86 //https://fathomless-lowlands-88456.herokuapp.com/like-mascota
87 //idc
88 //nombre
89
90 app.get("/like-mascota/:id:nombre", function(request, response){
91   var id = request.params.id;
92   var nombre = request.params.nombre;
93
94   var db = firebase.database();
95   var ref = db.ref("registrar-usuario/" + id);
96   var usuario = "";
97   var respuesta = {};
98
99   ref.on("value", function(snapshot) {
100     console.log(snapshot.val());
101     usuario = snapshot.val();
102     var mensaje = nombre + " te dio Like";
103
104     respuesta = {
105       id: id,
106       id_dispositivo: usuario.id_dispositivo,
107       nombre: usuario.id_usuario_instagramc
108     };
109     enviarNotificaion(usuario.id_dispositivo, mensaje);
110     response.send(JSON.stringify(respuesta));
111   }, function (errorObject) {
112     console.log("The read failed: " + errorObject.code);
113     respuesta = {
114       id: "",
115       id_dispositivo: "",
116       nombre: ""
117     };
118     response.send(JSON.stringify(respuesta));
119   });
120 });
```



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'node_modules', 'public', 'views', and files like 'index.js', 'package.json', 'PetagramEndPoint-c42b6196f9d9d.js', 'Procfile', 'README.md', and 'test.js'. The code editor shows the following JavaScript code:

```
121 });
122
123
124 function enviarNotificaion(tokenDestinatario, mensaje) {
125     var serverKey = 'AIzaSyDEdi3wKwnrAjtwaaf2JHLXaRpfjc';
126     var fcm = new FCM(serverKey);
127
128     var message = {
129         to: tokenDestinatario, // required
130         collapse_key: '',
131         data: {},
132         notification: {
133             title: 'Notificacion desde Servidor',
134             body: mensaje,
135             icon: 'notification',
136             sound: 'default',
137             color: "#000CD4"
138         }
139     };
140
141     fcm.send(message, function(err, response){
142         if (err) {
143             console.log("Something has gone wrong!");
144         } else {
145             console.log("Successfully sent with response: ", response);
146         }
147     });
148 }
149
150
151 app.listen(app.get('port'), function() {
152     console.log('Node app is running on port', app.get('port'));
153 });
154
```

Método para enviar notificación

```
private void enviarLikeMascota(String id, String id_dispositivo, String
idInsta) {

    final UsuarioResponse usuarioResponse = new
UsuarioResponse(id,id_dispositivo,idInsta);
    RestApiAdapterFirebase restApiAdapterFirebase = new
RestApiAdapterFirebase();
    EndpointsFirebase endpointsFirebase =
restApiAdapterFirebase.establecerConexionRestAPI();
    Call<UsuarioResponse> usuarioResponseCall =
endpointsFirebase.likeMascota(usuarioResponse.getId() ,
usuarioResponse.getId_usuario_instagram());
    usuarioResponseCall.enqueue(new Callback<UsuarioResponse>() {
        @Override
        public void onResponse(Call<UsuarioResponse> call,
Response<UsuarioResponse> response) {
            UsuarioResponse usuarioResponse1 = response.body();
            Log.d("ID_FIREBASE", usuarioResponse1.getId());
            Log.d("TOKEN_FIREBASE", usuarioResponse1.getId_dispositivo());

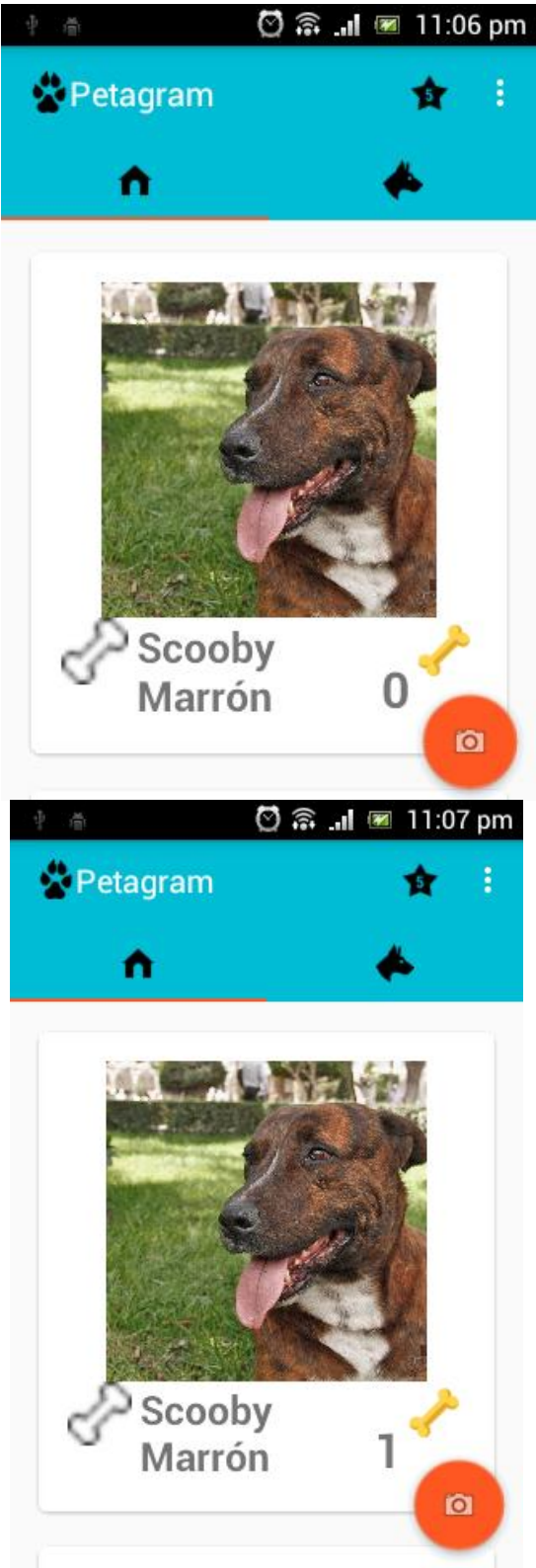
        }

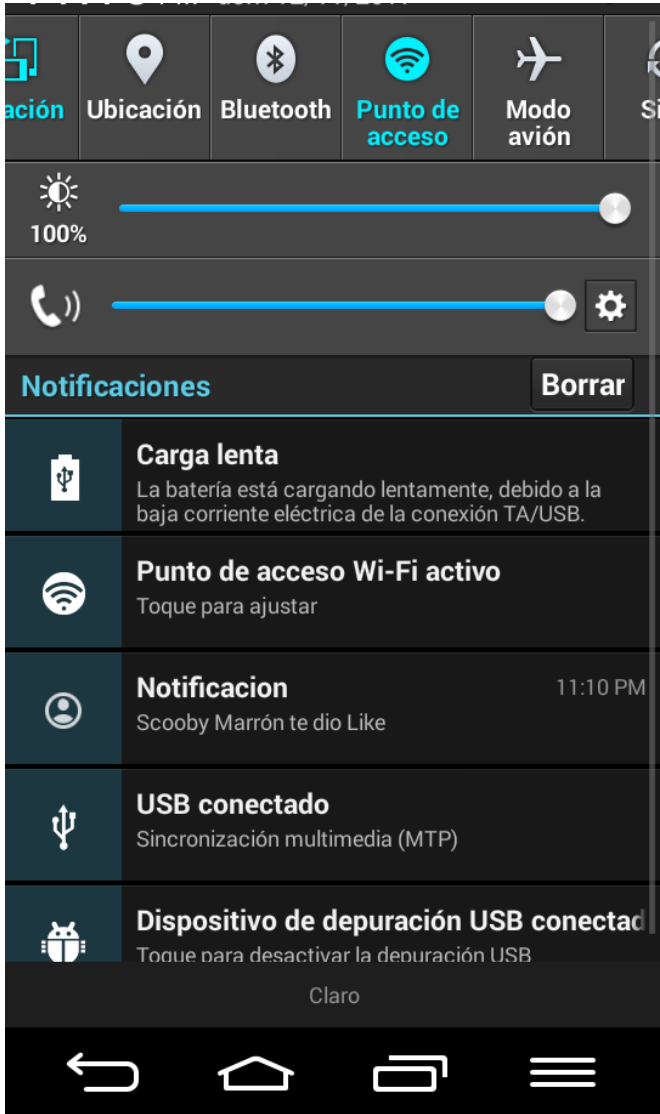
        @Override
        public void onFailure(Call<UsuarioResponse> call, Throwable t) {

        }

    });
}
```

CAPTURAS







Scooby Marrón



0



0



0

