

Fernando Rojas Hidalgo  
Semana 3. Tarea 2  
2018, Costa Rica.

## ENDPOINTS UTILIZADOS desde Java

```
package net.rohisa.mascotas.restApi;

import ...

/**
 * Created by frojash on 1/15/18.
 */

public interface EndpointsApi {

    //Trae las imagenes recientes de mi usuario
    @GET(ConstantsRestApi.URL_GET_RECENT_MEDIA_USER)
    Call<MascotaResponse> getRecentMedia();

    //Da like a una imagen de instagram
    @FormUrlEncoded
    @POST()
    Call<Gson> setLikeMedia(@Url String url, @Field("access_token") String token);

    //Trae las imagenes recientes de otro usuario
    @GET
    Call<MascotaResponse> getRecentMediaAnother(@Url String url);

    //Trae el dato de un like, con el fon de notificar
    @GET(ConstantsRestApi.KEY_GET_LIKE_NOTIFICATION)
    Call<UsuarioResponse> getLikeForNotification(@Path("id") String id, @Path("usuario") String usuario);

    //Trae la informacion de un usuario (id codigo), a partir de su username
    @GET
    Call<UsuarioResponse> getSearchUserCode(@Url String url);

    //Registra un token y su usuario de instagram
    @FormUrlEncoded
    @POST(ConstantsRestApi.KEY_POST_ID_TOKEN)
    Call<UsuarioResponse> registrarTokenID(@Field("token") String token, @Field("usuario") String usuario);

    //Registra un like en mi servidor o mi conteo personal
    @FormUrlEncoded
    @POST(ConstantsRestApi.KEY_POST_ID_LIKE)
    Call<UsuarioResponse> registrarLikeID(@Field("token") String token, @Field("usuario") String usuario, @Field("foto") String foto);
}
```

```
MainActivity.java x EndpointsApi.java x RestApiAdapter.java x DetalleAdaptador.java x JsonKeys.java x NotificationService.java x DetalleMascotaFragmentPresenter.java x

RestApiAdapter establecerConexionesRestApiInstagram()

package net.rohisa.mascotas.restApi.adapter;

import ...

/**
 * Created by frojash on 1/16/18.
 */

public class RestApiAdapter {

    //https://api.instagram[app] net.rohisa.mascotas.restApi t/?access_token=ACCESS-TOKEN
    //https://api.instagram[recent] recent/?access_token=ACCESS-TOKEN
    //https://api.instagram[usuario] usuario/?access_token=ACCESS-TOKEN
    public EndpointsApi establecerConexionesRestApiInstagram(Gson gson){
        Retrofit retrofit = new Retrofit.Builder()
            .baseUrl(ConstantsRestApi.ROOT_URL)
            .addConverterFactory(GsonConverterFactory.create(gson))
            .build();

        return retrofit.create(EndpointsApi.class);
    }

    //https://tranquil-garden-45815.herokuapp.com/registrar-usuario
    //https://tranquil-garden-45815.herokuapp.com/registrar-like
    public EndpointsApi establecerConexionesRestApiIdToken(){
        Retrofit retrofit = new Retrofit.Builder()
            .baseUrl(ConstantsRestApi.ROOT_URL_BASE)
            .addConverterFactory(GsonConverterFactory.create())
            .build();

        return retrofit.create(EndpointsApi.class);
    }

    public Gson construyeGsonDeserializadorMediaRecent(){
        GsonBuilder gsonBuilder = new GsonBuilder();
        gsonBuilder.registerTypeAdapter(MascotaResponse.class, new MascotaDeserializador());
        return gsonBuilder.create();
    }

    public Gson construyeGsonDeserializadorSearch(){
        GsonBuilder gsonBuilder = new GsonBuilder();
        gsonBuilder.registerTypeAdapter(UsuarioResponse.class, new UsuarioDeserializador());
        return gsonBuilder.create();
    }
}
```

## Endpoints Servidor

```
.get('/notificar-like/:id/:usuario', function(req, res) {
  var id = req.params.id;
  var usuario = req.params.usuario;

  var db = firebase.database();
  var ref = db.ref("registrar-like/" + id);
  var like = "";
  var respuesta = {};
  ref.on ("value", function(snapshot){
    like = snapshot.val();
    var mensaje = usuario + " te dio un like.";
    enviarNotificacion(like.id_dispositivo, mensaje)
    respuesta = {
      id : like.id_dispositivo,
      foto : like.id_foto_instagram,
      usuario : like.id_usuario_instagram
    };
    res.send(JSON.stringify(respuesta));
  }, function (errorObject){
    console.log("The read failed: " + errorObject.code);
    respuesta = {
      id : "",
      foto : "",
      usuario : ""
    };
    res.send(JSON.stringify(respuesta));
  });
})
```

```

    .post('/registrar-like', function(req, res) {
        var token = req.body.token;
        var foto = req.body.foto;
        var usuario = req.body.usuario;
        var db = firebase.database();
        var tokenDevices = db.ref("registrar-like").push();

        tokenDevices.set({
            id_foto_instagram : foto,
            id_usuario_instagram : usuario,
            id_dispositivo : token
        });

        var path = tokenDevices.toString();
        var pathSplit = path.split("registrar-like/");
        var idAutoGenerado = pathSplit[1];
        var respuesta = generarRespuestaLike(db, idAutoGenerado);

        res.setHeader("Content-Type", "application/json");
        res.send(JSON.stringify(respuesta));
    })

```

```

function enviarNotificacion(tokenDestinatario, mensaje){
    var serverKey = 'AAAASZvVJps:APA91bG0QCm13Smek3CV99SIlaD__AuT6S4cPe9VWLz6A93YQ5P';
    //var serverKey = 'AIzaSyB2ajUBD3nwEw99ui_4zqUU2Ei10Rd1ISA';

    var fcm = new FCM(serverKey);
    var message = {
        to: tokenDestinatario, // required fill with device token or topics
        collapse_key: '',
        data: { },
        notification: {
            title: 'Notificación desde servidor',
            body: mensaje,
            sound : "default",
            color : "#00BCD4",
            icon : "icons8_cat_footprint_filled_50"
        }
    };
}

```

## TODO EL CODIGO DEL NODE.js

```
const express = require('express')
const path = require('path')
const PORT = process.env.PORT || 5000
var bodyParser = require("body-parser");
var firebase = require("firebase");
var FCM = require('fcm-push');
firebase.initializeApp({
  serviceAccount: "Mascotas-67164317762d.json",
  databaseURL: "https://mascotas-51565.firebaseio.com"
});
```

```
express()
  .use(bodyParser.json())//soporte para codificar JSON
  .use(bodyParser.urlencoded({extended: true}))
  .use(express.static(path.join(__dirname, 'public')))
  .set('views', path.join(__dirname, 'views'))
  .set('view engine', 'ejs')
  .get('/android', (req, res) => res.render('pages/index'))
  .get('/notificar-like/:id/:usuario', function(req, res) {
    var id = req.params.id;
    var usuario = req.params.usuario;
```

```
var db = firebase.database();
var ref = db.ref("registrar-like/" + id);
var like = "";
var respuesta = {};
ref.on ("value", function(snapshot){
  like = snapshot.val();
  var mensaje = usuario + " te dio un like.";
  enviarNotificacion(like.id_dispositivo, mensaje)
  respuesta = {
    id : like.id_dispositivo,
    foto : like.id_foto_instagram,
```

```

        usuario : like.id_usuario_instagram
    };
    res.send(JSON.stringify(respuesta));
}, function (errorObject){
    console.log("The read failed: " + errorObject.code);
    respuesta = {
        id : "",
        foto : "",
        usuario : ""
    };
    res.send(JSON.stringify(respuesta));
});
})

```

```

.post('/registrar-usuario', function(req, res) {
    var token = req.body.token;
    var usuario = req.body.usuario;
    var db = firebase.database();
    var tokenDevices = db.ref("registrar-usuario").push();

```

```

    tokenDevices.set({
        id_dispositivo : token,
        id_usuario_instagram : usuario
    });

```

```

    var path = tokenDevices.toString();
    var pathSplit = path.split("registrar-usuario/");
    var idAutoGenerado = pathSplit[1];
    var respuesta = generarRespuestaToken(db, idAutoGenerado);

```

```

    res.setHeader("Content-Type", "application/json");
    res.send(JSON.stringify(respuesta));
})
.post('/registrar-like', function(req, res) {
    var token = req.body.token;
    var foto = req.body.foto;
    var usuario = req.body.usuario;

```

```
var db = firebase.database();  
var tokenDevices = db.ref("registrar-like").push();
```

```
tokenDevices.set({  
  id_foto_instagram : foto,  
  id_usuario_instagram : usuario,  
  id_dispositivo : token  
});
```

```
var path = tokenDevices.toString();  
var pathSplit = path.split("registrar-like/");  
var idAutoGenerado = pathSplit[1];  
var respuesta = generarRespuestaLike(db, idAutoGenerado);
```

```
res.setHeader("Content-Type", "application/json");  
res.send(JSON.stringify(respuesta));  
})  
.listen(PORT, () => console.log(`Listening on ${ PORT }`))
```

```
function generarRespuestaToken(db, idAutoGenerado){  
  var respuesta = {};  
  var usuario = "";  
  var ref = db.ref("registrar-usuario");  
  ref.on ("child_added", function(snapshot,prevChildKey){  
    usuario = snapshot.val();  
    respuesta = {  
      id : idAutoGenerado,  
      token : usuario.id_dispositivo,  
      usuario : usuario.id_usuario_instagram  
    };  
  })  
  return respuesta;  
}
```

```
function enviarNotificacion(tokenDestinatario, mensaje){
```

```
var serverKey =  
'AAAASZvVJps:APA91bG0QCm13Smek3CV99SIlad__AuT6S4cPe9VWLz6A93YQ5P  
V1eRuwSLebUhn80PmIMuZiTXnRl48uBJdK0XWyrG7ETTTZvXDsmL20esu3Vcs8W9  
rmgH_HpfXv1uuf7HhkzARfvm0';  
//var serverKey = 'AIzaSyB2ajUBD3nwEw99ui_4zqUU2Ei10Rd1ISA';
```

```
var fcm = new FCM(serverKey);  
var message = {  
  to: tokenDestinatario, // required fill with device token or  
  topics:  
  collapse_key: '',  
  data: { },  
  notification: {  
    title: 'Notificación desde servidor',  
    body: mensaje,  
    sound : "default",  
    color : "#00BCD4",  
    icon : "icons8_cat_footprint_filled_50"  
  }  
};  
//callback style  
fcm.send(message, function(err, response){  
  if (err) {  
    console.log("Something has gone wrong!");  
  } else {  
    console.log("Successfully sent with response: ",  
response);  
  }  
});  
//promise style  
fcm.send(message)  
  .then(function(response){  
    console.log("Successfully sent with response: ",  
response);  
  })  
  .catch(function(err){
```

```
        console.log("Something has gone wrong!");  
        console.error(err);  
    })  
}
```

```
function generarRespuestaLike(db, idAutoGenerado){  
    var respuesta = {};  
    var usuario = "";  
    var ref = db.ref("registrar-like");  
    ref.on ("child_added", function(snapshot,prevChildKey){  
        usuario = snapshot.val();  
        respuesta = {  
            id : idAutoGenerado,  
            token : usuario.id_dispositivo,  
            usuario : usuario.id_usuario_instagram,  
            foto : usuario.id_foto_instagram  
        };  
    })  
    return respuesta;  
}
```