# Petagram
## Configurando mi propio Endpoint en mi Servidor

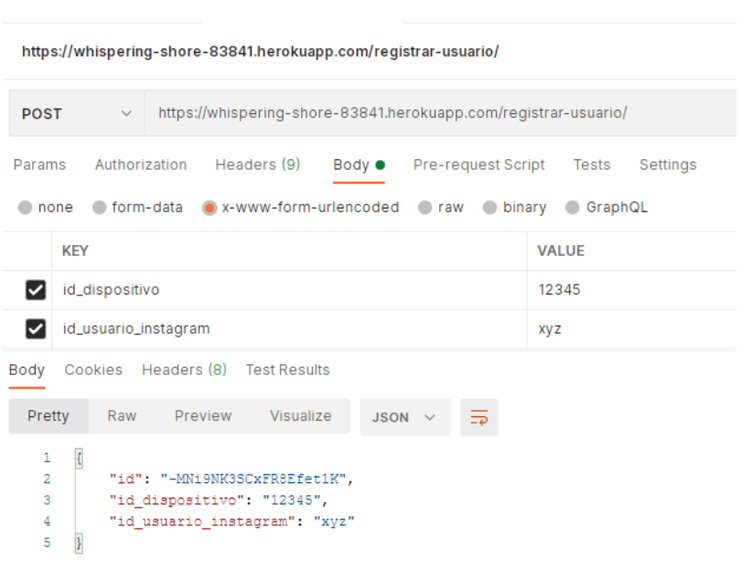
Siguiendo los pasos indicados en el video del curso, y adaptando el código a las actualizaciones de las clases de java implementadas, registrando mi aplicación en Firebase de Google, creo el Endpoint en Heroku con Node.js siguiendo los pasos del video de Coursera. Quedando registrado en la base de datos creada en Firebase, los datos con la estructura requerida.

Creación del Endpoint con Node.js

```
var tokenDevicesURI = 'registrar-usuario' //'token-device'
app.post('/' + tokenDevicesURI, (req, res) =>{
    //var token = req.body.token
    var id_dispositivo = req.body.id_dispositivo
    var id_usuario_instagram = req.body.id_usuario_instagram
    var db = firebase.database()
    var tokenDevices = db.ref(tokenDevicesURI).push()
    tokenDevices.set({
        id_dispositivo: id_dispositivo,
        id_usuario_instagram: id_usuario_instagram
    })
    var pathRegistro = tokenDevices.toString()
    var idReg = pathRegistro.split(tokenDevicesURI+'/')[1]
    var respuesta = generarRespuestaAToken(db,idReg)
    res.setHeader('Content-Type','aplication/json')
    res.send(JSON.stringify(respuesta))
} )

function generarRespuestaAToken(db,idReg){
    var respuesta = {}
    var registro= ''
    var ref = db.ref('registrar-usuario')
    ref.on('child_added',function(snapshot, prevChildKey){
        registro = snapshot.val()
        respuesta = {
            id: idReg,
            id_dispositivo: registro.id_dispositivo,
            id_usuario_instagram: registro.id_usuario_instagram
        }
    })
    return respuesta
}
```

Prueba desde Postman:



```
public static final String ROOT_URL_MI_API = "https://whispering-shore-83841.herokuapp.com/";
public static final String KEY_POST_ID_TOKEN = "registrar-usuario/";
```

Recibe los parámetros: id_dispositivo, id_usuario_instagram

*Método de la Interfaz im*plementada:

```
@FormUrlEncoded
@POST(ConstantesRestApi.KEY_POST_ID_TOKEN)
Call<UsuarioResponse> registrarTokenID(@Field("id_dispositivo") String id_dispositivo,
                                       @Field("id_usuario_instagram") String id_usuario_instagram);
```

**El** id_dispositivo es el Token regresado de la clase: FirebaseMessaging.

```
private void enviarNotificacion() {
    // Get token
    // [START log_reg_token]
    FirebaseMessaging.getInstance().getToken()
            .addOnCompleteListener(new OnCompleteListener<String>() {
                @Override
                public void onComplete(@NonNull Task<String> task) {
                    if (!task.isSuccessful()) {
                        Log.w(TAG, "Fetching FCM registration token failed", task.getException());
                        return;
                    }

                    // Get new FCM registration token
                    String token = task.getResult();

                    // Log and toast
                    String msg = getString(R.string.msg_token_fmt, token);
                    Log.d(TAG, msg);
                    Toast.makeText(MascotasActivity.this, msg, Toast.LENGTH_SHORT).show();
                    NotificationService notificationService = new NotificationService();
                    notificationService.onNewToken(token);
                }
            });
    // [END log_reg_token]
}
```

Y sobrecargando el método

```
@Override
public void onNewToken(@NonNull String token) {
    // super.onNewToken(s);
    Log.i("onNewToken",token);
    enviarTokenRegistro(token);
}
```

De mi clase de Notificación

```
public class NotificationService extends FirebaseMessagingService
```

Para registrar los  id_dispositivo, id_usuario_instagram en la base de datos de Firebase

```
private void enviarTokenRegistro(String id_dispositivo) {
    Log.d(TAG,id_dispositivo);
    RestApiAdapter restApiAdapter = new RestApiAdapter();
    IEndpoinsApi iEndpoinsApi = restApiAdapter.establecerConexionRestMiApi();
    Call<UsuarioResponse> usuarioResponseCall = iEndpoinsApi.registrarTokenID(id_dispositivo, Consta
    usuarioResponseCall.enqueue(new Callback<UsuarioResponse>() {
        @Override
        public void onResponse(Call<UsuarioResponse> call, Response<UsuarioResponse> response) {
            UsuarioResponse usuarioResponse = response.body();
            Log.d("ID_FIREBASE",usuarioResponse.getId());
            Log.d("Id_dispositivo",usuarioResponse.getId_dispositivo());
            Log.d("Id_usuario_instagram",usuarioResponse.getId_usuario_instagram());
        }

        @Override
        public void onFailure(Call<UsuarioResponse> call, Throwable t) {
            //Toast.makeText(this,"Error Notificación",Toast.LENGTH_SHORT).show();
            Log.e("FIREBASE",t.getStackTrace().toString());
        }
    });
}
```

Registro en la base de datos Firebase de Google.