

He usado Heroku con Node.js, al igual que el ejemplo del curso.

El vídeo de cómo funciona:

<https://youtu.be/mkRqyPNqeMw>

En el EndPoint “registrar-usuario” se envía id_dispositivo junto con el id_usuario_instagram y el nombre_usuario_instagram, que quedan guardados en Firebase

```
//https://warm-stream-32312.herokuapp.com/registrar-usuario
//usuario_instagram
//id_dispositivo (token)
//id_usuario_instagram
var usuariosInstagramURI="registrar-usuario";
app.post("/"+usuariosInstagramURI, function(request, response){
  var id_disp = request.body.id_dispositivo; //token
  var id_user = request.body.id_usuario_instagram;
  var nom_usuario = request.body.nombre_usuario_instagram;

  var db = firebase.database();
  var usuariosInstagram = db.ref(usuariosInstagramURI).push();
  usuariosInstagram.set({
    id_dispositivo: id_disp,
    id_usuario_instagram: id_user,
    nombre_usuario_instagram: nom_usuario
  });

  //para coger el id de la bbdd
  var path = usuariosInstagram.toString();
  var pathSplit = path.split(usuariosInstagramURI+"/");
  var idAutoGenerado = pathSplit[1];

  var resp=generaRespuesta(db, idAutoGenerado);
  response.setHeader("Content-Type", "application/json");
  response.send(JSON.stringify(resp));

  //response.send(request.body.id_dispositivo);
})
```

La respuesta sólo devuelve el id_dispositivo (token) (aunque está preparado para devolver el JSON completo):

```
function generaRespuesta(db, idAuto){
  var respuesta={};
  var registro = "";
  var ref = db.ref(usuariosInstagramURI);
  ref.on("child_added", function(snapshot, prevChildKey){
    registro = snapshot.val();
    respuesta = {
      /* //sólo se devuelve el id_dispositivo (token) en el JSON
      id: idAuto,
      */
      id_dispositivo: registro.id_dispositivo
      /* //sólo se devuelve el id_dispositivo (token) en el JSON
      ,id_usuario_instagram: registro.id_usuario_instagram
      ,nombre_usuario_instagram: registro.nombre_usuario_instagram
      */
    }
  });
  return respuesta;
}
```