

Tareas calificadas por los compañeros: Creando, Recibiendo y Actuando Push Notifications

Explicacion del Endpoint

Instagram

El Endpoint que se implementó permite dar un like a una imagen de una cuenta de Instagram, se le pasa el Access token y el id de la imagen que se le quiere mandar un token

POST/media/media-id/likes

```
curl -F 'access_token=ACCESS-TOKEN' \  
https://api.instagram.com/v1/media/{media-id}/likes
```

```
@POST(ConstantsRestApi.URL_SET_LIKES)  
Call<LikeInstagramResponse> setLikeInstagram(@Path("media-id") String media_id,  
                                              @Query("access_token") String accessToken);
```

Heroku

El Endpoint que utilice para generar las notificaciones y guardar el toque, es el siguiente:

<https://kinder-hockey-59168.herokuapp.com/like-instagram>

Dicho endpoint recibe 2 parámetros, como se pueden ver a continuación:

/like-instagram/:fotoInstagram/:usuarioInstagram

Donde:

fotoInstagram, es la URL de la foto de Instagram a la que se le dio un toque.

UsuarioInstagram, perfil de usuario al que se le dio un toque a su foto.

Funcionamiento del EndPoint:

El endpoint recibe dos parámetros, la foto a la que se le dio el toque, el nombre de usuario a quien pertenece la foto. Primero guarda en Firebase el id del usuario que dio un toque, la URL de la foto a la que se le dio un toque:



Inmediatamente busca el token del dispositivo, en <https://kinder-hockey-59168.herokuapp.com/token-device>, relacionado con *usuarioInstagram* para enviarle la notificación a su Smartphone. Para buscarlo lo hace mediante un `forEach`.

Lo puedes revisar mediante el siguiente código.

```
//GET
//https://kinder-hockey-59168.herokuapp.com/like-instagram
//fotoInstagram
//usuarioInstagram

app.get("/like-instagram/:fotoInstagram/:usuarioInstagram", function (request,
response) {
    var foto          = request.params.fotoInstagram;
    var usuarioI      = request.params.usuarioInstagram;

    var db = firebase.database();

    //guarda los datos del like
    var likeInstagram = db.ref("like-instagram").push();
    likeInstagram.set({
        usuario: usuarioI,
        foto: foto
    });

    //-- buscar por USUARIO del dispositivo (token) para enviar la notificación
    var refToken = db.ref("token-device");
    var tokenObj = "";
    var dispositivoToken = "";
    var respuesta = {};
    refToken.on("value", function (snapshot) {
        tokenObj = snapshot.val();
        snapshot.forEach(function (childSnapshot) {
            var child = childSnapshot.val();
            if(child.instagram == usuarioI)
            {
                console.log("Usuario encontrado: ", child.instagram);
                dispositivoToken = child.token;
            }
        });
    });

    var mensaje = "Te dieron un like";
    enviarNotificacion(dispositivoToken, mensaje);
    respuesta = {
        fotoInstagram: foto,
        usuarioInstagram: usuarioI,
        dispositivo: dispositivoToken
    }
});
```

```

    }

    response.send(JSON.stringify(respuesta));

  });

function enviarNotificacion(tokenDestinatario, mensaje) {
  var serverKey = 'AIzaSyDvQ6kCG1l15k3ls6tTVxS-n812qrkKq_k';
  var fcm = new FCM(serverKey);

  var message = {
    to: tokenDestinatario,
    collapse_key: '',
    data: {},
    notification: {
      title: 'Notificación desde Servidor',
      body: mensaje,
      icon: "ic_action_name",
      sound: "default",
      color: "#03A9F4"
    }
  };

  fcm.send(message, function (err, response) {
    if (err) {
      console.log("Something has gone wrong!", response);
    } else {
      console.log("Successfully sent with response: ", response);
    }
  });
}

```

Plataforma que usaste para montarlo

Las plataformas que se utilizaron para hacer la tarea de PushNotifications, fueron las mismas que se utilizaron en curso, es decir, Heroku para el servidor y alojar los Web Services, en lenguaje Node JS, y FireBase para guardar la información de la aplicación. Igual se utilizaron Endpoints de Instagram.