

En el paquete restAPI está todo lo relacionado con el webservice de Instagram

En el paquete restAPIFirebase, todo lo relacionado con el webservice a Heroku para el manejo de Firebase (base de datos y notificaciones)

Se añade evento onClickListener a la imagen del ViewHolder, para que lance la petición al webservice de Instagram y al de Heroku (donde guardará en base de datos Firebase, y enviará la notificación a través de Cloud Messaging)

EndPoint Instagram para dar Like

Para hacer like, se usa el EndPoint [/media/{media-id}/likes](#), donde por POST se pasa como parámetro el "access_token", pero en el link va el media-id, que es el id de la foto

Se monta un deserializador, para averiguar el código devuelto en el JSON, habiendo diseñado una clase "LikeInst"

Si el WS no devuelve el código 200 o hay algún otro problema, se envía un error. En caso contrario se actualiza la activity para ver el cambio

EndPoint en Heroku para añadir registros en Firebase

Devuelve el token del dispositivo, para enviar notificación

```
//https://warm-stream-32312.herokuapp.com/like-generado
//id_dispositivo {token}
//id_usuario_instagram
//id_foto_instagram
var likesInstagramURI="like-generado";
app.post("/"+likesInstagramURI, function(request, response){
    var id_disp = request.body.id_dispositivo; //token
    var id_user = request.body.id_usuario_instagram;
    var id_foto = request.body.id_foto_instagram;

    var db = firebase.database();
    var likesInstagram = db.ref(likesInstagramURI).push();
    likesInstagram.set({
        id_dispositivo: id_disp,
        id_usuario_instagram: id_user,
        id_foto_instagram: id_foto
    });

    //para coger el id de la bbdd
    var path = likesInstagram.toString();
    var pathSplit = path.split(likesInstagramURI+"/");
    var idAutoGenerado = pathSplit[1];
    //Devuelve lo que haya en la bbdd con ese ID
    var resp=generaRespuestaLikes(db, idAutoGenerado);
    response.setHeader("Content-Type", "application/json");
    response.send(JSON.stringify(resp));
})

function generaRespuestaLikes(db, idAuto){
    var respuesta={};
    var registro = "";
    var ref = db.ref(likesInstagramURI);
    ref.on("child_added", function(snapshot, prevChildKey){
        registro = snapshot.val();
        respuesta = {
            id_dispositivo: registro.id_dispositivo
        }
    });
    return respuesta;
}
```

EndPoint a Instagram para averiguar el dueño de una foto

https://api.instagram.com/v1/media/{media-id}?access_token=ACCESS-TOKEN

Se pasa el media-id de la foto, y mediante des-serialización averiguamos el id del usuario

Este id-usuario es el que pasaremos en el endpoint para enviar la notificación:

Notificaciones

Una vez averiguado el dueño de la foto, lo busco en la base de datos de Firebase de los que se hayan registrado, y si el dispositivo no es el mismo desde el que se está haciendo la petición se le envía el mensaje. (Lo que no sé es cómo averiguar cuál es el dispositivo original del dueño de la foto, ya que en el ejercicio anterior guardaba todos los que daban like o accedían al timeline de una cuenta, pero no se define si la cuenta era suya).

EndPoint notifica-usuario en Heroku

Es el encargado de conectar con Firebase para enviar la notificación. Sólo la envía si el token del dispositivo destino no es el que ha hecho like en la foto (para evitar que lo envía a sí mismo).

También se le pasan como parámetros el identificador de usuario y la URL de la foto de perfil de usuario, para que los envíe en la notificación como datos, y poder abrir la actividad con el time line del

Devuelve el primer id_usuario_instagram y nombre_usuario_instagram que encuentre con esas condiciones, para poder abrir la activity con el timeline de ese usuario propietario de la foto.

Importante:

En click hay que hacerlo en una foto (no en el hueso, por facilidad), y es cuando se aumenta el like (se recarga la actividad para que se vea), se guarda el like y se envía la notificación

Al hacer click en la notificación: si la aplicación está abierta se abre el timeline del usuario dueño de la foto; si no está abierta simplemente la abre con el usuario por defecto (no he sido capaz de enviar parámetros extras al PendingIntent; si alguien ha sabido cómo, le agradezco la ayuda....)

Una vez averiguado el dueño de la foto, lo busco en la base de datos de Firebase de los que se hayan registrado, y si el dispositivo no es el mismo desde el que se está haciendo la petición se le envía el mensaje. (Lo que no sé es cómo averiguar cuál es el dispositivo original del dueño de la foto, ya que en el ejercicio anterior guardaba todos los que daban like o accedían al timeline de una cuenta, pero no se define si la cuenta era suya).

Video

<https://youtu.be/XaSdQrLnozU>

Hay dos emuladores, cada uno con un usuario registrado:

En el Nexus 5X, Javier

En el Nexus 6, Leo.App



1.- Se hace click en una foto (no en el hueso) de Javier, y el Nexus 5X recibe la notificación, que abre la app

2.- Una vez abierta la app, se va al usuario de leo.app y se hace click en una foto suya, y entonces el NEXus 6 (registrado por Leo) recibe la notificación, que al pulsarla se abre el time line de la cuenta del dueño de la foto (Leo.App)

Cada vez que se hace like, se ve cómo se guarda en la bbdd de Firebase los registros correspondientes

2.- y se envía la notificación al usuario dueño de la foto (registrado previamente)