

Canvis a l'API i perquè

Hem canviat dos mètodes a usuaris per a poder efectuar de manera correcta el login dels usuaris:

Post: Abans no estava implementat ja que no es demanava a la segona entrega i tampoc feia falta per a fer la primera versió del login per a la primera entrega, on la redirecció de google ja feia la funció de crear el nou usuari a sessions_controller. Ara crea l'usuari amb només els camps que són necessaris per a la nostra pàgina web quan no ha estat registrat (per a tenir exactament les mateixes funcionalitats que la primera versió).

Get: Abans, agafàvem tots el usuaris indiscriminadament ja que amb el scaffolding generat automàticament ens era suficient per a fer proves de testing de la nostra primera versió però mancava per a la nostra versió final ja que necessitàvem identificar l'usuari el qual volia fer login des de Google i ara el que es fa es enviar un paràmetre anomenat token (encara que és el id de Google, això s'ha escollit així ja que no es volia confondre a tots els integrants de l'equip) el qual fa retornar l'usuari que està registrat en el sistema.

Mecanisme de login

Per a explicar el mecanisme del login, s'explicarà amb un exemple pràctic i argumentant pas a pas el que passa lògicament a la pàgina web.

“La Maria va a <https://react-issue-tracker-asw.herokuapp.com/> i vol fer un login”

Primer de tot, Maria fa click al botó de Login, això fa que vagi a Google per a demanar les seves credencials, si és satisfactori el login va a una funció anomenada responseGoogle on té tota la informació de la resposta que ha fet Google, aquí el sistema fa un set del localStorage per a poder mantenir-se loguejat encara que es canviï de pàgina principal a crear una issue. Si no estava registrada, el sistema envia un post al servidor per a registrar-la al esquema per a futurs login. A continuació, es refresca la pàgina per a mostrar els camps actualitzats de l'usuari.

Diagrama de seqüência

