

# Canvis a la API

---

Els canvis que heu hagut de fer a la vostra API i per què.

- **Hem afegit una operació de login per poder gestionar l'autenticació d'usuaris.** Des del front podem obtenir l'usuari de Google, però sense aquesta operació de login no es podria crear l'usuari a la base de dades, necessària per a que l'usuari tingui karma, about i el user\_id, que és el que es relaciona amb la resta de models del sistema.
- **Hem afegit un bool a les contribucions que diu si l'usuari que fa la petició li ha donat a like.** Necessitàvem tenir una forma de saber si l'usuari actual té als seus likes cada contribució, i fins ara, no retornàvem aquesta condició, i per solucionar aquest problema, hauríem d'haver fet una crida a l'api per veure tots els likes de l'usuari loguejat, i si alguna de les contribucions corresponia a algún like. Això suposa una crida extra i informació duplicada, així que la millor solució ha estat passar aquest atribut "liked".
- Abans quan s'intentava crear una contribució amb una URL ja existent, es retornava un 302. Hem descobert que el 302 el gestiona automàticament el navegador redirigint a la URL del header location. Com no volem això, ja que seria recarregar la pàgina i no és convenient a una Single Page Application, **hem canviat el codi a 200.** El resultat final és que quan es crea una contribució retorna un 201 Created, i si no es crea perquè la URL està repetida retorna un 200 OK (i evidentment retorna la contribució existent amb aquesta URL, això ja ho feia).
- **Hem afegit un atribut a les replies anomenat parent\_type.** El que conté aquest atribut és el tipus del comentari pare (Comment o Reply). Necessitem aquest bool per a que el link 'parent' que hi ha a les replies vagi a /replies/:id o a /comments/:id. Sense aquest bool, hauríem de fer un get a /api/comments/:id del pare i si no dóna error és un comment, si dóna error és reply (això seria una mica arriscat perquè pot donar error per un altre motiu). L'altre opció seria fer dues peticions, una a /api/comments/:id i l'altre a /api/replies/:id. Amb aquest bool solucionem aquest problema sense peticions extres i d'una forma molt més còmoda.

# Mecanisme d'autenticació

---

Com funciona el mecanisme d'autenticació, des del moment que l'usuari prem el botó de login fins que finalment veu que ja està loguejat. Ajudeu-vos d'un diagrama de seqüència per fer ho més entenedor.

Quan l'usuari prem el botó de *login*, es crida a la funció d'una llibreria externa que es comunica amb Google per obtenir l'usuari de Google, és a dir, la id corresponent a aquest usuari.

Quan tenim l'usuari de Google, guardem el `googleId` al `localStorage` i l'enviem a `/api/login`. Aquesta funció el primer que fa és mirar si aquest usuari és a la base de dades. Si no hi és, el crea i retorna el `userId`. Si existeix, retorna el `userId`. Quan tenim el `userId`, el guardem també a `localStorage`. Llavors, en el lloc del botó de *login*, apareixen dos botons: un per accedir al perfil de l'usuari i l'altre per fer *logout*.

# Diagrama de seqüência

---

