

Social Media Aggregator Challenge

Santiago Federella

En grandes razgos la aplicación está conformada por:

1. Tres microservicios, que son APIs REST hechas con Scala Play:
 - a. social-media-aggregator-security
 - b. social-media-aggregator-users
 - c. social-media-aggregator-posts
2. Un cliente web hecho en React.
3. Bases de datos MongoDB.

Explicaré el funcionamiento en un gran caso de uso:

1. El usuario entra al cliente y se logea con su usuario y contraseña de twitter. Este proceso lo realiza el cliente contra la API de *security* que a su vez se conecta con la API de twitter y obtiene los tokens de acceso, y son retornados al browser seteando una cookie de sesión.
2. Una vez con la sesión iniciada, el cliente hace una llamada Ajax al servicio de *users* para obtener los datos del usuario logeado. Para verificar que los tokens son correctos, *users* hace una consulta a *security*. Si bien por ahora las únicas rutas aseguradas se encuentran en *users*, tener la seguridad como servicio aparte permiten que varios servicios puedan mantener una única sesión.
3. Una vez que se obtienen los datos del usuario, el cliente consulta al servicio de *posts* por los post correspondientes a los intereses del usuario. Asimismo, se registra en el stream donde este irá publicando todos los nuevos posts que vayan entrando usando SSE. Esto será un broadcast así que dependerá del cliente mostrar o no mostrar los posts que correspondan.
4. Para modificar sus intereses o agregar nuevos boards, el usuario accederá en el botón de "Editar". Aquí podrá:
 - a. Agregar, eliminar y editar boards existentes.
 - b. En cada board seguir a usuarios o a temas. Los usuarios se buscarán en un select que irá buscando en twitter usuarios existentes, en cambio los temas se deberán escribir simplemente.
 - c. Dejar de seguir los usuarios y los temas simplemente eliminándolos.
5. Todo el punto 4 se realiza a través de interacciones con el servicio de *users*. Cuando un usuario comienza a seguir a otro usuario o a un tema, el servicio *users* envía esta información al servicio de *posts*, quién almacena todas las cosas que se siguen y corre cada un minuto (configurable) un proceso en background para obtener tweets de las mismas.

Arquitectura:

