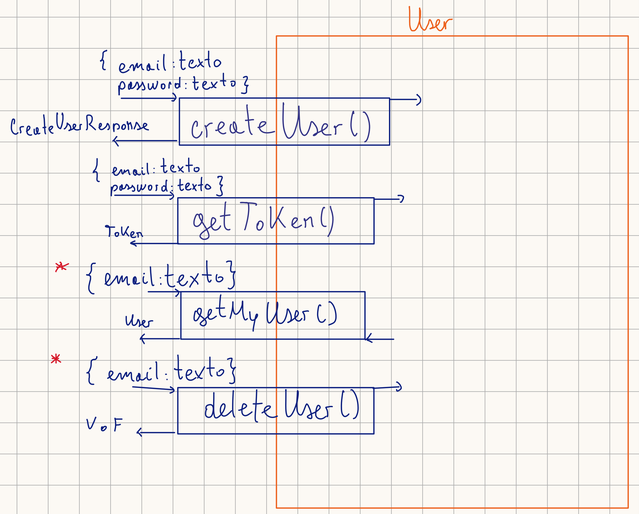
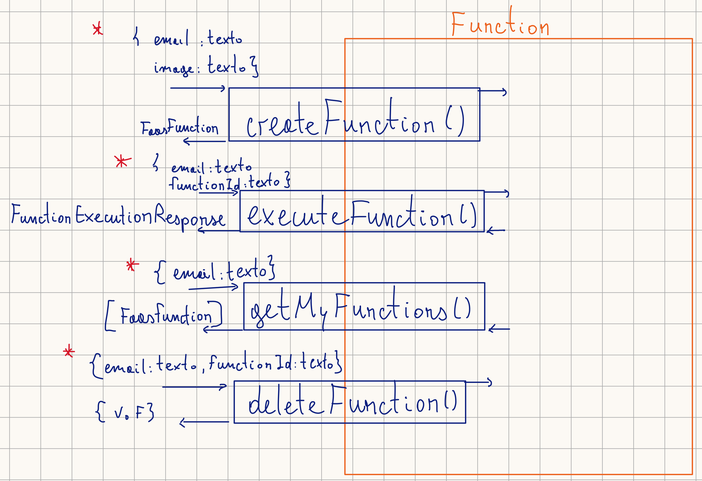
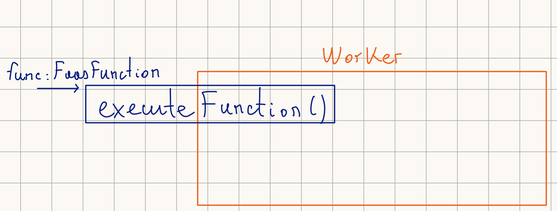
PSC. Diseño FAAS

Funciones y Módulos







* Nota: Todas las funciones marcadas van a estar protegidas con autenticación JWT en el servidor REST. El token llevará dentro el email del usuario. El servidor REST será quien descifre el token y llame a las funciones pasandoles el email y otros argumentos

Descripción Funciones

createUser:
1. Buscar en la BD un usuario con el email proporcionado
2. Si existe el usuario lanzamos una excepción
3. Salteamos y encriptamos la contraseña
4. Creamos el objeto usuario con el email y contraseña encriptada
5. Guardamos el usuario
6. Devolvemos un objeto con datos no sensibles del usuario (excluimos contraseña)

getToken:
1. Buscamos el usuario en la BD
2. si no existe lanzamos una excepción
3. Encriptamos y salteamos la contraseña recibida como parámetro
4. Comparamos las contraseñas, si no coinciden, lanzamos una excepción
5. Buscamos a ver si el usuario tiene un token
6. Si existe el token y es vigente lo devolvemos
7. Sino creamos uno, lo guardamos y lo devolvemos

getMyUser:
1. Buscamos el usuario en la BD
2. Si no existe lanzamos una excepcion
3. Devolvemos un objeto con los datos no sensibles del usuario

deleteUser:
1. Buscamos el usuario en la BD
2. Si no existe lanzamos una excepcion
3. Lo borramos y devolvemos un objeto con un booleano en funcion del resultado de la operación en la BD (normalmente será true)

createFunction: Creamos la función y la persistimos

getMyFunctions: Buscamos las funciones del usuario en la BD y se las devolvemos

deleteFunction:
1. Buscamos la Función en la BD
2. Si no existe lanzamos una excepción
3. Comparamos el id del usuario con el de la función
4. Si no coinciden lanzamos una excepcion
5. Borramos la función y retornamos

executeFunction:
1. Comprobamos en la BD el número de funciones activas
2. Si se supera el límite lanzamos una excepción
3. Nos suscribimos a un topico para recibir la respuesta
4. Mandamos la función al worker
5. Registramos la función como activa en la BD
6. Esperamos la respuesta durante un tiempo, si no llega lanzamos una excepción
7. En caso de que llegue, la devolvemos y marcamos la funcion como desactivada

esto se puede hacer directamente antes de devolver la respuesta o se puede lanzar un evento de dominio p.ej: FaasFunctionExecuted que tenga un listener encargado de actualizar el estado de la función de manera asíncrona.

(WORKER)

executeFunction:
1. Bajarse la imagen docker
2. Crear y lanzar el contenedor
3. Recoger logs del contenedor
4. Enviar respuesta con los logs
5. Eliminar el contenedor

Nota: hay que tener en cuenta el espacio del host donde se aloja Docker, puede interesar borrar imágenes descargadas en caso de recibir algún error por falta de espacio. No lo pongo por simplificar, pero en un sistema real que vaya a producción se debería tener un mecanismo para esto, ya sea en la lógica de la función o mediante cronjobs