Nosotros ya tenemos el eurito que han generado.

Obviamente cuando esto falla por ejemplo si pongo otro número no tenemos lo que es el mensaje de la

editora nos dispara un error y esto es bastante conveniente.

Vamos a trabajar con esas dos partes en este momento.

Regresemos a nuestra aplicación.

Cerremos el locking component y trabajemos en el servis okey tanto el locking como el nuevo usuario

pasan por este posteo.

Este posteo ustedes ya saben ya lo han visto anteriormente.

Esto regresa un observable.

Ok voy a mostrarles esto regresa un observable en el cual yo puedo hacer ciertas modificaciones antes

de notificar a lo que la página de registro o la página de login es decir cuando el locking lance la

suscripción yo puedo ejecutar un código antes y ahí es donde voy a almacenar el token.

Si es que viene entonces voy a crearme dos métodos aka voy a quedarme un método privado Rivet voy a

ponerlo a escribirlo bien Rivet espacio guardar token necesito mandar el Heydi token o esto va a ser

de tipo string Abreus Rojas lo que voy a hacer aquí son dos cosas uno en la clase como tal servis voy

a crearme una nueva propiedad llamada token o user token token de tipo string estén ensalzados como

uno y eso está bien o sea on the fine no tiene ningún valor.

Ahora este usar token.

Aquí es donde yo voy a colocar si ya existe Leydi token.

De esa manera lo voy a tener rápidamente a la mano entonces digamos disputo y usar token va a ser igual

a Heydi token que yo estoy recibiendo como argumento a este método y adicionalmente voy a llamar en

locales Storage punto set Aitken parentesis y vamos a almacenarlo en una propiedad llamada token con

la llave token.

Y el valor que almacenamos tiene que ser un string y el string perfectamente es el Eirik token.

Eso es básicamente toda la parte de grabación del token.

Ahora necesito otro método para leerlo de locales Storage sea llamar leer token parentesis abrir sólo

llaves no necesito ningún argumento pero sí necesito antes verificar si tengo información ahí entonces

y si en el local Storage punto que hoy tenemos lo que es el token recuerde que tiene que ser esta misma

llave con el que yo estoy grabándolo.

Si esto existe entonces voy a decir que disputo y usar token va a ser igual a la obtención de Cresto

H.B Haytham token ok.

Eso sería básicamente todo y si no existe el token entonces podemos inicializar al user token como un

estro invasivo.

Ok entonces tendríamos una pequeña validación ahí muy fácil de hacer eventualmente.

Ok si el token existe entonces lo leemos de locales Storch y si no existe mandamos retornamos un estro

invasivo podríamos retornar un protón del disputo y usar token si nosotros quisiéramos también podríamos

hacer la privada hay muchas cosas que se pueden hacer acá pero este es mi método para leer el token

ya tengo los dos métodos ahora en donde voy a llamar el de guardar el toque yo sé en teoría en teoría

déjenme mostrarles en teoría yo sé que aquí tengo el token pero en el login component y la verdad es

que el componen no le interesa mucho saber si el token ya está acá porque aquí en este punto ya me parece

un poco tarde para hacer la asignación de leer token se podría hacer pero no lo veo conveniente cargar

mi Lovin component con esa información.

Entonces lo voy a hacer del lado del cervix.

Empecemos con el nuevo usuario que el nuevo usuario yo puedo pasar toda esta petición del HTTP apuntó

pasó por un el PIB parentesis Enter.

Aquí falta un punto y coma en el PIB puedo utilizar el operador Mapp porque el mapa de los RX J es el

cual él me va a permitir obtener la respuesta cuando se obtenga de este post de una suscripción y pasarla

por el MAP el MAP la puede transformar puede regresar lo que ustedes quieran o simplemente servir como

intermediario o simplemente leer la respuesta.

Y si bien el token entonces ahí lo voy a almacenar.

La ventaja del MAP es que si esto regreso un error el MAP nunca se dispara.

Eso ya lo vamos a ver entonces el mapa está marcando un error necesito importarlo aquí arriba importa

algo from entre comillas RX JS Slash Operators con ese punto y coma.

Aquí vamos a importar el mapa perfecto.

Eso es básicamente todo lo que hacemos aquí abajo en el MAP obtendría la respuesta.

Y aquí viene una función común y corriente en esta función.

Recuerden que sólo se va a ejecutar si tenemos éxito ejecutando esta primera petición.

Si esto regresa un error entonces eso nunca se va a disparar.

Si ustedes quisiera manejar o atrapar o atrapar perdón el error acá tendríamos que usar otro operador

llamado Catch Catch error que también es de los RDX J3.

Pero no lo vamos a usar ahí porque ya tenemos la validación acá.

Aquí estamos error.

Entonces si todo sale bien.

Aquí está la misma respuesta que tiene toquín y todas las cosas.

Lo único que voy a hacer aquí es disputo guardar toquen y voy a mandar la respuesta de la respuesta.

El Heydi toquen noten como lo estoy capitalizando adicionalmente para que este no me bloqueé la respuesta.

Yo voy a tener que retornar nuevamente la respuesta.

Esto simplemente es como poner un pequeño pedazo de una tubería y filtrar toda la información que entra

por ahí por este mapa.

Ok también puedo poner acá lo que diga entró en el mapa del RX J.S.

Recuerden que solo estamos en el nuevo usuario podemos copiar todo este código.

Todo ese pack lo vamos a copiar y busquemos nuestro blogging quiten este punto y coma y peguen esto

ahí no se preocupen por este error ustedes lo pueden deshabilitar pero es una sugerencia del JBS SIGINT

me dice que esto es literal ahí no es permitido pero bueno si es permitido seguramente es alguna regla

que desconozco pero eso está bien.

Voy a grabar los cambios y regresemos a la aplicación.

Podríamos probar con uno válido entonces uno incorrecto del 1 al 6 tocó ingresar noten que entró en

el mapa del RX J.V y tenemos el token nuevamente.

Si yo me voy a Orange Application me voy aquí al local Storage aquí tengo esta información no deberíamos

de tener pero aquí tengo mi toque.

Si quieren lo podemos volver a hacer porque yo tenía información para que me crean.

Vuelvo a tocar el botón de ingresar y aquí aparece el token.

Si no les apareciera entonces toca el botón de refresh sólo para estar seguros de que tenemos la información

ahí perfecto.

Ahora probemos crearnos un nuevo usuario limpiar el local storage.

No tengo nada ahí voy a crear una nueva cuenta pero recuerde que tiene que ser una cuenta de correo

electrónico que no exista.

Melissa Flores va a pedir del 1 al 6 también tocó crear cuenta y debería aparecerme el token ahí perfecto.

Ya no me sirve solo lo puedo borrar.

Al igual que estoy aquí abajo lo puedo borrar.

Básicamente ya tenemos la parte de almacenar el token en lo que es el local Storch podríamos hacer otra

cosa leer token lo podríamos ejecutar esa instrucción.

Justo cuando iniciamos el servicio de observes aquí simplemente voy a ponerlo como disputo leer token.

Así justo cuando ingresa ya sabemos si tenemos un toque o no ok.

El lockout es relativamente sencillo ya lo vamos a trabajar cuando creamos la pantalla del home y la

pantalla protegida pero ya no vamos a trabajar por los momentos.

Ya tenemos lo que es el almacenamiento de nuestro token en el local Storage asegúrense grabar todos los cambios.