

User Refresh Token

Leer primero <https://incentro.atlassian.net/wiki/spaces/LPCR/pages/4721541134> Can't find link .

Ruta: (post) api/v1/user/refreshToken (middlewares que observan esta ruta [AuthenticationChannel]), la cual apunta al controlador UserController, método RefreshToken.

Lo que hacemos con esta api es exactamente lo mismo que en remember token a excepción de la parte de usar el remember y usar un registro de la tabla remember tokens, en su lugar vamos a utilizar el refresh token value del campo de user tokens que va a refrescar el registro de user token (en el linke al comienzo se ve como se generaba) y borrar el anterior, esto quiere decir.

Tenemos nuestro registro en la base de datos de user token (img 1).



id	user_id	token	expired_at	refresh_token	refresh_expired_at	created_at	updated_at	deleted_at
8	18	d5DzG0pdyf9hSEUfEYODuZnZChYvWOW5SamJmSDA2hZVX...	2024-06-06 14:31:47	R1FwXdBaGdSWXZkUnZSZ2ZIZzNITVR5SFZMWG15blpzQ2U4a1BnSjJpVU9sdzVkaLB2bmpRaUZXyVZWc0pYZXVSSHlYajF6bEJRb2t...	2024-06-06 15:01:47	2024-06-06 14:31:47	2024-06-06 14:31:47	2024-06-06 14:31:47

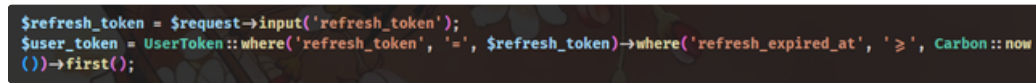
img 1

el parámetro que espera la api es refresh_token en vez de remember_token.

en este ejemplo sería:

R1FwXdBaGdSWXZkUnZSZ2ZIZzNITVR5SFZMWG15blpzQ2U4a1BnSjJpVU9sdzVkaLB2bmpRaUZXyVZWc0pYZXVSSHlYajF6bEJRb2t
vQ2JPMUVkSFVDanIRNUFTaUhUV0IPSmpXRUFsenlicXptak9UaFZBTIQzeElyWW5jUG1yWWR6WTNhS1NPTEhqeFUxbzdmRndtSm44
RGZSbUpSOHlEdVBCUEduOGoyTjF0QkYwckduSjRqOGFkSjZ0WUQ3azh2UXZ0d3U=

De la tabla user token sacamos el usuario que tenga en el campo refresh token el valor del token que acabamos de enviar y que el expired_at sea superior a la hora actual (que no esté caducado) (img 2),



```
$refresh_token = $request->input('refresh_token');  
$user_token = UserToken::where('refresh_token', '=', $refresh_token)->where('refresh_expired_at', '>=', Carbon::now())->first();
```

img 2

extracción de user y company , creación de nueva data igual que en refresh token.

ejecutamos la consulta y se espera que se borre el registro con el refresh token que enviamos como param y se cree un registro nuevo con los nuevos valores (img 3).



id	user_id	token	expired_at	refresh_token	refresh_expired_at	created_at	updated_at	deleted_at
8	18	d5DzG0pdyf9hSEUfEYODuZnZChYvWOW5SamJmSDA2hZVX...	2024-06-06 14:31:47	R1FwXdBaGdSWXZkUnZSZ2ZIZzNITVR5SFZMWG15blpzQ2U4a1BnSjJpVU9sdzVkaLB2bmpRaUZXyVZWc0pYZXVSSHlYajF6bEJRb2t...	2024-06-06 15:01:47	2024-06-06 14:31:47	2024-06-06 14:31:47	2024-06-06 14:31:47
9	21	8iFpSdGp0pdyf9hSEUfEYODuZnZChYvWOW5SamJmSDA2hZVX...	2024-06-06 15:01:47	R1FwXdBaGdSWXZkUnZSZ2ZIZzNITVR5SFZMWG15blpzQ2U4a1BnSjJpVU9sdzVkaLB2bmpRaUZXyVZWc0pYZXVSSHlYajF6bEJRb2t...	2024-06-06 15:01:47	2024-06-06 15:01:47	2024-06-06 15:01:47	2024-06-06 15:01:47

img 3

Vemos que se generó el nuevo registro, este nuevo registro generado la data será enviada como respuesta (img 4) y el registro anterior vemos que en deleted_at se aplicó el soft delete.

```
{
  "error": 200,
  "data": [
    {
      "user_id": 0,
      "token": "Bearer
d1F5aDgzcn0ScjB3MEp0TzcycnVja0xiQXNlVFZOMjRBNVpKamhIVU1uZ1dzb3V0bWVhcn1Z6aU1maEZzZ3k2Tm1dTBBDNg4cnFYbHFnZn1B9Tz4dE1KXQjVem1OU3VpQjB6b1ZwdmNzdkc5YmhCdW1
sR3Btc2N2R8bnQet5aFuaWlpYmEhM2ZwaGdyZ2V1ZG9VIRvbfFnykY1mR0VGcxOXRhanh9mccbnRnOGNlVn1zSVV3S1VXNDJlVVR0O0Iyb2RyUEM=",
      "refresh_token":
"OUh1bEg3VhGR0BqTEQ3QKRATjczRm16NFVQcTJecU3t9WpZnp6cEdRZGZzTEVBT0FwZk1oa1o1cjhCeu1jTG1QVEM5Y0M1S1V1VnhBTUSkYk3ab0h1Y3ZqdWpZZEtBZE5pb3JpaE9wW1BmU1NkT0
Q4Wl0Z2Fd1OH8zdGc5N2ZVemo2az1FVE3pUP1Mv3PHZn3CVnp6UWFTTVZOHFkbFabWd1SEVFEpUZZEwTDE3emRRWGFpNGVFE53Vgxx50HKeWVldHY=",
      "profile_completed": 1
    }
  ]
}
```

img 4