**APIs de Autogestión de Usuarios**

1. ***Route::get('/user', 'UserController@getUserLoggedJWT');*** //Esta API también se llama después del login contra el portal, si EXISTE el usuario -> devuelve el objeto usuario, SI NO existe -> devuelve 404
   1. El token DEBE traer el user\_name cuando la API se llama inmediatamente después del loguin (todavía el front NO sabe el user\_id)
   2. El token DEBE traer el user\_id cuando la API se llama para que un usuario se autoadministre/visualice su cuenta.
2. ***Route::post('/user', 'UserController@updateUserLoggedJWT');***
   1. El token DEBE traer el user\_id, la API se llama para que un usuario se autoadministre/actualice su cuenta.
   2. *Valores pasados por el body:*

'name' => ['sometimes','string', 'max:255'],

'surname' => ['sometimes','string', 'max:255'],

'grade' => ['sometimes','nullable','string'],

'dni' => ['sometimes','integer'],

'location\_id' => ['sometimes','integer', 'exists:locations,id']

**APIs de Administración de Usuarios**

1. ***Route::post('/admin/user', 'UserController@createUserJWT');***
   1. El token DEBE traer el user\_id del ADMIN que se validará si es un usuario Administrador.
   2. *Valores pasados por el body:*

'user\_name' => ['required','string', 'max:255'],

'name' => ['sometimes','string', 'max:255'],

'surname' => ['sometimes','string', 'max:255'],

'grade' => ['sometimes','nullable','string'],

'dni' => ['sometimes','integer'],

'location\_id' => ['sometimes','integer', 'exists:locations,id']

1. ***Route::get('/admin/user/{user\_id}', 'UserController@getUserManagedJWT');***
   1. El token DEBE traer el user\_id del ADMIN que se validará si es un usuario Administrador.
   2. En la URL se envía el ID del usuario del cual se está pidiendo información.
2. ***Route::post('/admin/user/{user\_id}', 'UserController@updateUserManagedJWT');***
   1. El token DEBE traer el user\_id del ADMIN que se validará si es un usuario Administrador.
   2. En la URL se envía el ID del usuario que se desea actualizar.
   3. *Valores pasados por el body:*

'user\_name' => ['required','string', 'max:255'],

'name' => ['sometimes','string', 'max:255'],

'surname' => ['sometimes','string', 'max:255'],

'grade' => ['sometimes','nullable','string'],

'dni' => ['sometimes','integer'],

'location\_id' => ['sometimes','integer', 'exists:locations,id']

**NOTAS:**

* Cuando la actualización la está haciendo un Administrador, también se deja modificar el user\_name. Un usuario común NO puede actualizar su propio user\_name