

JWT

Level 1: Install Laravel and Setup Basic User Tables

- Update the migration staff table migration file by opening create_staff_table.php in database/migrations. Inside the up function, type the lines below to build the table columns.

```
> php artisan make:migration create_staff_table
```

- Update the migration staff table migration file by opening create_staff_table.php in database/migrations. Inside the up function, type the lines below to build the table columns.

```
Schema::create('staff', function (Blueprint $table) {  
    $table->id();  
    $table->string("first_name");  
    $table->string("last_name");  
    $table->string("email");  
    $table->string("password");  
    $table->timestamps();  
});
```

Level 2: Install Laravel JWT Package :

- Install the package using composer,
- Publish the configuration file which is located in your config/jwt.php
- Generate a secret token. This secret token will be added to your .env file.

```
> composer require tymon/jwt-auth  
> php artisan vendor:publish --provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"  
> php artisan jwt:secret
```

- We need to tell Laravel to have two guards, one is for normal user and another one for the staff. Both of these guards will use jwt as the driver but the provider will be coming from two different tables. To do this, let's update **auth.php**

```
'defaults' => [  
    'guard' => 'api', // set the default guard to 'api'  
    'passwords' => 'users',  
],
```

```
'guards' => [  
    'web' => [  
        'driver' => 'session',  
        'provider' => 'users',  
    ],  
  
    'api' => [  
        'driver' => 'jwt', // set to jwt token  
        'provider' => 'users',  
        'hash' => false,  
    ],  
  
    'staff-api' => [ // add another guard for the staff  
        'driver' => 'jwt',  
        'provider' => 'staff', // set the provider  
        'hash' => false,  
    ],  
],
```

```
'providers' => [  
    'users' => [  
        'driver' => 'eloquent',  
        'model' => App\User::class,  
    ],  
    'staff' => [ // Add new staff provider  
        'driver' => 'eloquent',  
        'model' => App\Staff::class,  
    ],  
  
]
```

Level 3: Staff Authentication:

➤ `app/Http/Controllers/StaffAuthenticationController.php`

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Auth\Access\AuthorizationException;
```

```
use Illuminate\Http\Response;
```

```
class StaffAuthenticationController extends Controller
```

```
{
```

```
    public function store()
```

```
    {
```

```
        if ($token = auth('staff-api')->attempt(request()->only('email', 'password'))) {
```

```
            return response()->json([
```

```
                'token' => $token,
```

```
                'message' => "Login successful"
```

```
            ]);
```

```
        }
```

```
        throw new AuthorizationException("Invalid Credentials");
    }

    public function show()
    {
        return response()->json(['data' => ['user' => auth('staff-api')->user()]]);
    }

    public function destroy()
    {
        auth('staff-api')->logout();
        return response()->json(['message'=>'Logout'], Response::HTTP_ACCEPTED);
    }
}
```

➤ app/Staff.php

```
<?php

namespace App;

use Illuminate\Foundation\Auth\User as Authenticatable;
use Tymon\JWTAuth\Contracts\JWTSubject;

class Staff extends Authenticatable implements JWTSubject
{
    /**
     * @inheritDoc
     */
    public function getJWTIdentifier()
    {
        return $this->getKey();
    }

    /**
     * @inheritDoc
     */
    public function getJWTCustomClaims()
    {
        return [];
    }
}
```

➤ **api.php**

```

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});
Route::post('auth/login', "AuthController@store");
Route::post('/staff/auth/login', "StaffAuthenticationController@store");

Route::group(['middleware' => 'auth:api'], function () {
    Route::get('account/me', "AuthController@show");
    Route::delete('auth/logout', "AuthController@destroy");
});

Route::group(['middleware' => 'auth:staff-api'], function () {
    Route::get('staff/me', "StaffAuthenticationController@show");
    Route::delete('staff/auth/logout', "StaffAuthenticationController@destroy");
});

```

Resources:

- <https://dripcoder.com/posts/multi-table-laravel-jwt-authentication/>
- <https://www.positronx.io/laravel-jwt-authentication-tutorial-user-login-signup-api/>
- Commande : `composer require tymon/jwt-auth --ignore-platform-reqs`