

# Laravel Sanctum

Install Laravel

```
composer create-project --prefer-dist laravel/laravel:^11 laravel_sanctum_app
```

pindah directory

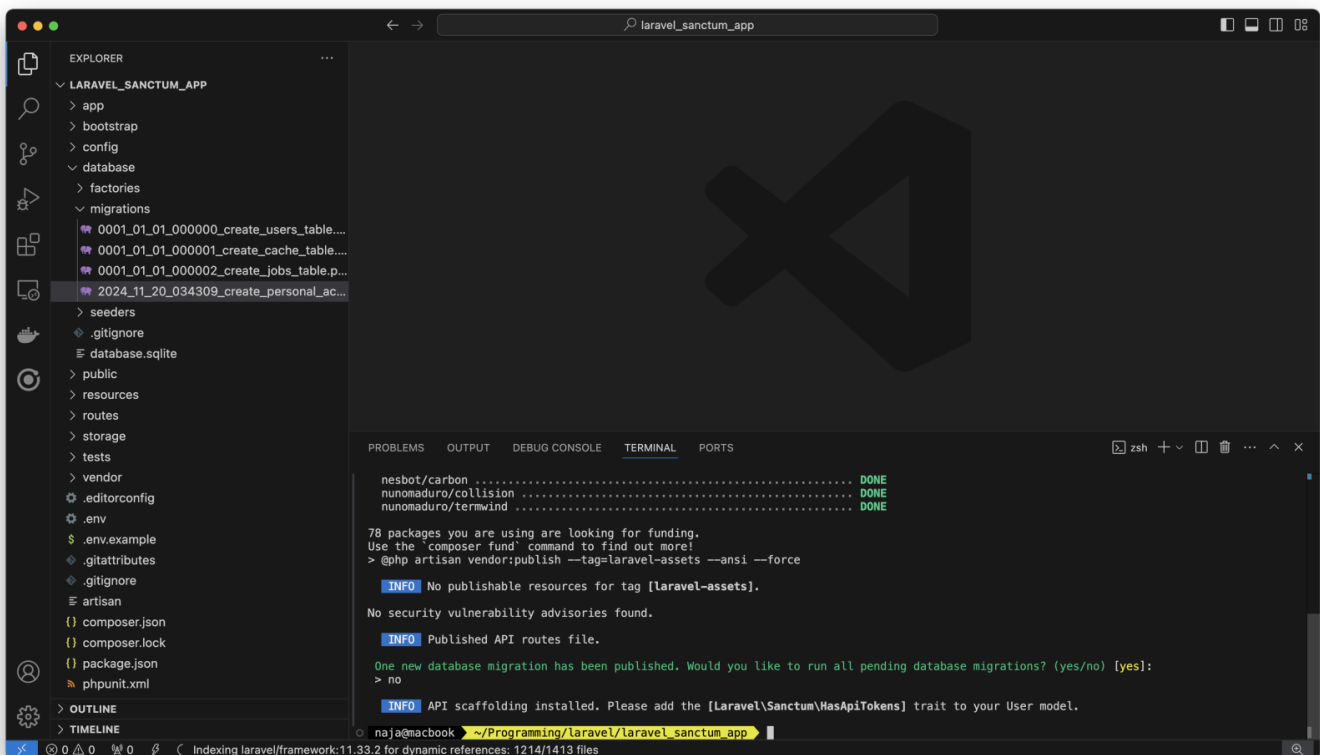
```
cd laravel_sanctum_app
```

buka dengan vscode

```
code .
```

setting env untuk koneksi database dan kemudian install sanctum

```
php artisan install:api
```



The screenshot shows a VS Code window with the Explorer view on the left displaying the file structure of the 'laravel\_sanctum\_app' project. The main editor area shows the output of the 'php artisan install:api' command in the terminal. The output includes the following text:

```
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE

78 packages you are using are looking for funding.
Use the "composer fund" command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

[INFO] No publishable resources for tag [laravel-assets].

No security vulnerability advisories found.

[INFO] Published API routes file.

One new database migration has been published. Would you like to run all pending database migrations? (yes/no) [yes]:
> no

[INFO] API scaffolding installed. Please add the [Laravel\Sanctum\HasApiTokens] trait to your User model.
```

jalankan migrate

```
php artisan migrate
```

tambahkan

```
HasApiTokens
```

pada model User.php

```
php artisan make:controller API/AuthController
```

```
public function register(Request $request)
{
    $validator = Validator::make($request->all(), [
        'name' => 'required',
        'email' => 'required|email',
        'password' => 'required',
        'confirm' => 'required|same:password'
    ]);

    if($validator->fails()){
        return response()->json([
            'success' => false,
            'message' => 'Ada kesalahan',
            'data' => $validator->errors()
        ]);
    }

    $input = $request->all();
    $input['password'] = bcrypt($input['password']);
    $user = User::create($input);

    $success['token'] = $user->createToken('auth_token')->plainTextToken;
    $success['name'] = $user->name;

    return response()->json([
        'success' => true,
        'message' => "Sukses Register",
        'data' => $success
    ]);
}
```

```
public function login(Request $request)
{
    if (Auth::attempt(['email' => $request->email, 'password' => $request->password]))
    {
        $auth = Auth::user();
        $success['token'] = $auth->createToken('auth_token')->plainTextToken;
        $success['name'] = $auth->name;

        return response()->json([
            'success' => true,
            'message' => 'Login sukses',
            'data' => $success
        ]);
    } else {
        return response()->json([
            'success' => false,
            'message' => 'Cek email dan password lagi',
            'data' => null
        ]);
    }
}
```

---

Revision #1

Created 20 November 2024 03:37:37 by M. Najamudin Ridha

Updated 20 November 2024 05:53:23 by M. Najamudin Ridha