Laravel Developer Assessment

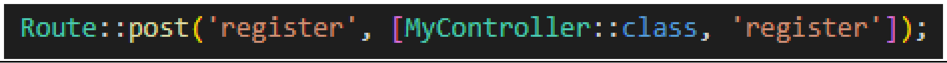
Test API

Membuat fungsi Rest API untuk :

1. **Register**

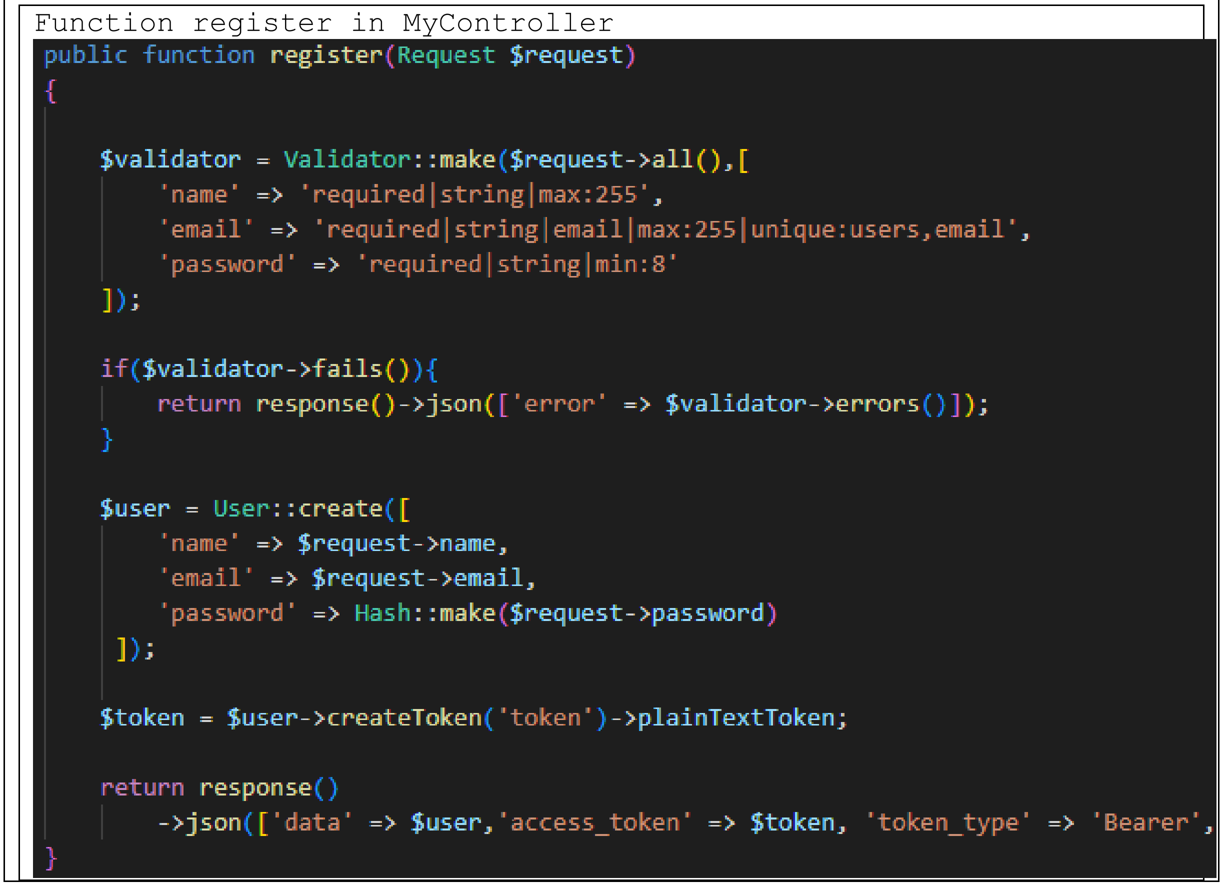
   Code:

Route



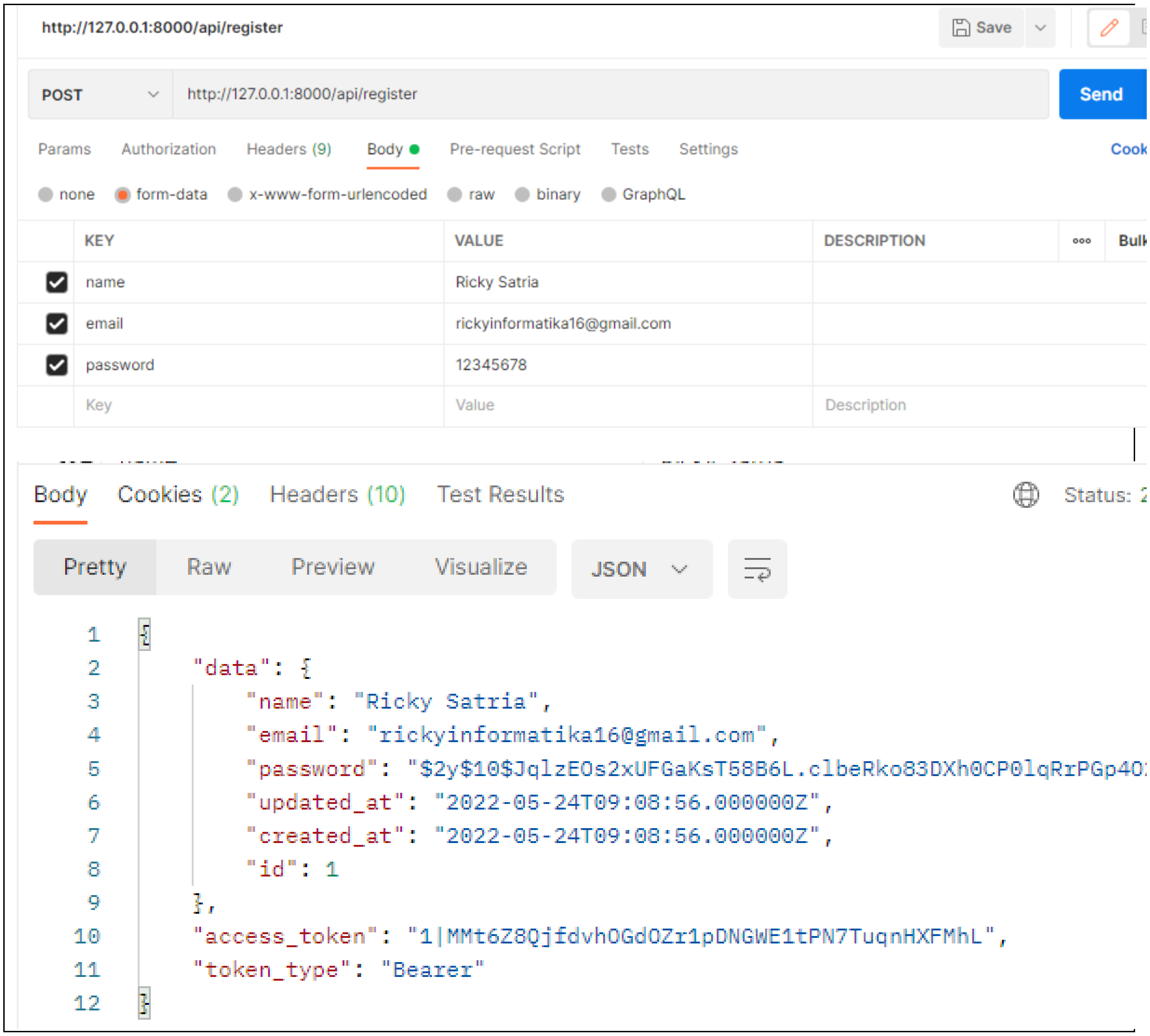
```php
Route::post('register', [MyController::class, 'register']);
```

Function register in MyController



```php
public function register(Request $request)
{

    $validator = Validator::make($request->all(),[
        'name' => 'required|string|max:255',
        'email' => 'required|string|email|max:255|unique:users,email',
        'password' => 'required|string|min:8'
    ]);

    if($validator->fails()){
        return response()->json(['error' => $validator->errors()]);
    }

    $user = User::create([
        'name' => $request->name,
        'email' => $request->email,
        'password' => Hash::make($request->password)
     ]);

    $token = $user->createToken('token')->plainTextToken;

    return response()
        ->json(['data' => $user,'access_token' => $token, 'token_type' => 'Bearer',
}
```

URL and Output:



POST http://127.0.0.1:8000/api/register

	KEY	VALUE	DESCRIPTION
☑	name	Ricky Satria	
☑	email	rickyinformatika16@gmail.com	
☑	password	12345678	
	Key	Value	Description

```
{
    "data": {
        "name": "Ricky Satria",
        "email": "rickyinformatika16@gmail.com",
        "password": "$2y$10$JqlzEOs2xUFGaKsT58B6L.clbeRko83DXh0CP0lqRrPGp40
        "updated_at": "2022-05-24T09:08:56.000000Z",
        "created_at": "2022-05-24T09:08:56.000000Z",
        "id": 1
    },
    "access_token": "1|MMt6Z8QjfdvhOGdOZr1pDNGWE1tPN7TuqnHXFMhL",
    "token_type": "Bearer"
}
```

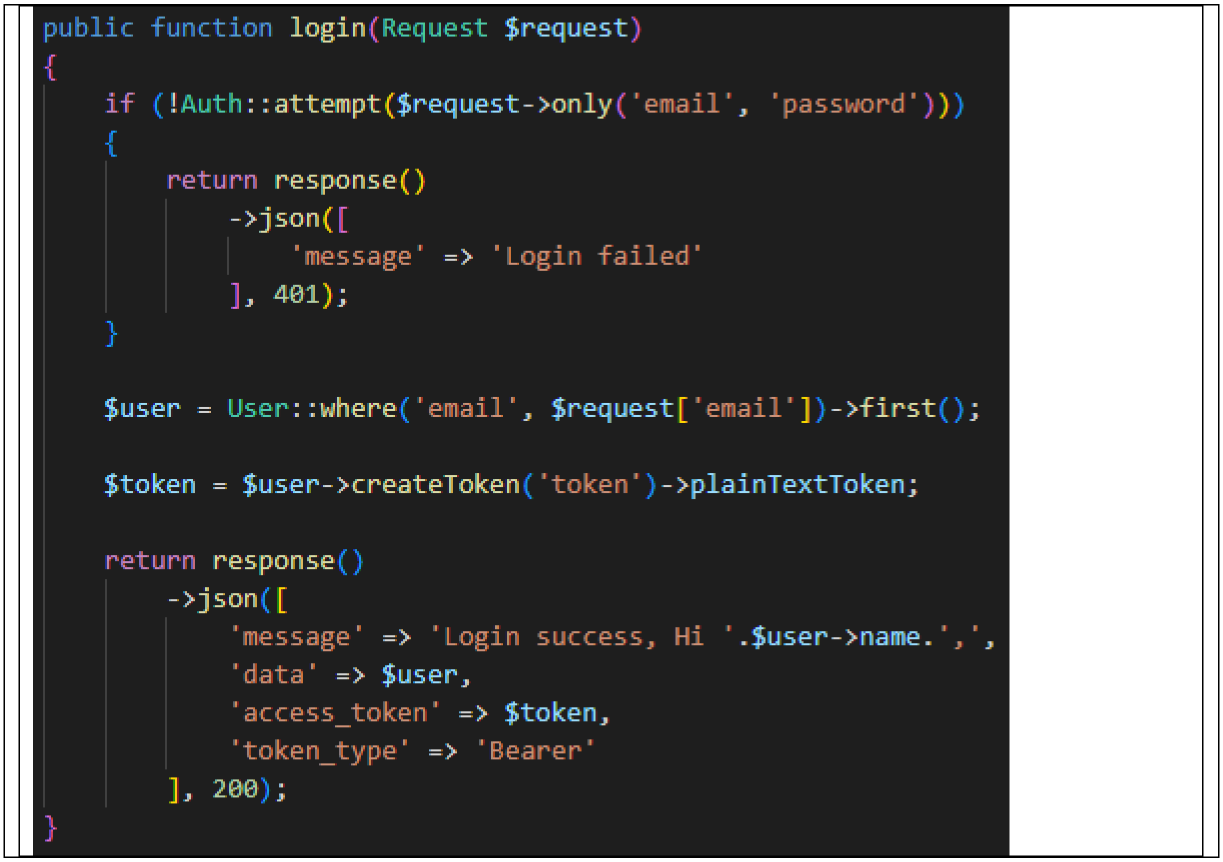
2. **Login**

Code:

Route

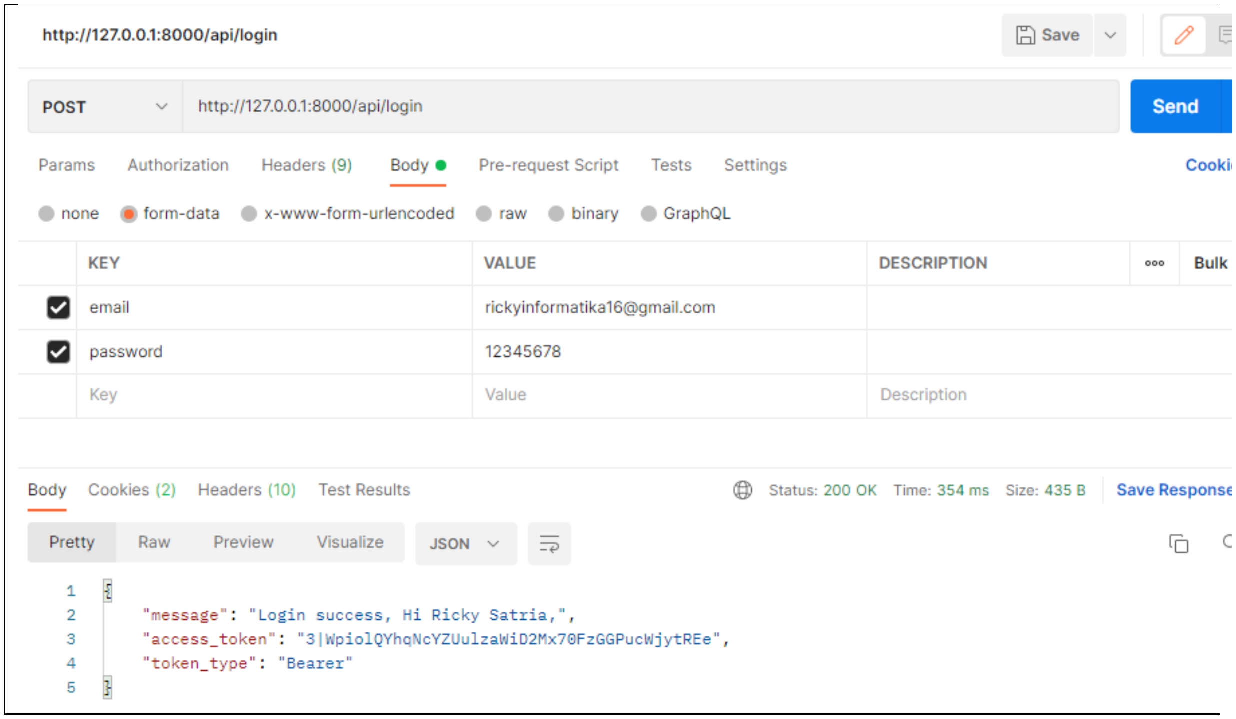
```
Route::post('login', [MyController::class, 'login']);
```

Function login in MyController



```php
public function login(Request $request)
{
    if (!Auth::attempt($request->only('email', 'password')))
    {
        return response()
            ->json([
                'message' => 'Login failed'
            ], 401);
    }

    $user = User::where('email', $request['email'])->first();

    $token = $user->createToken('token')->plainTextToken;

    return response()
        ->json([
            'message' => 'Login success, Hi '.$user->name.',',
            'data' => $user,
            'access_token' => $token,
            'token_type' => 'Bearer'
        ], 200);
}
```

URL and Output:



http://127.0.0.1:8000/api/login

POST http://127.0.0.1:8000/api/login

Body: ○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	email	rickyinformatika16@gmail.com	
☑	password	12345678	
	Key	Value	Description

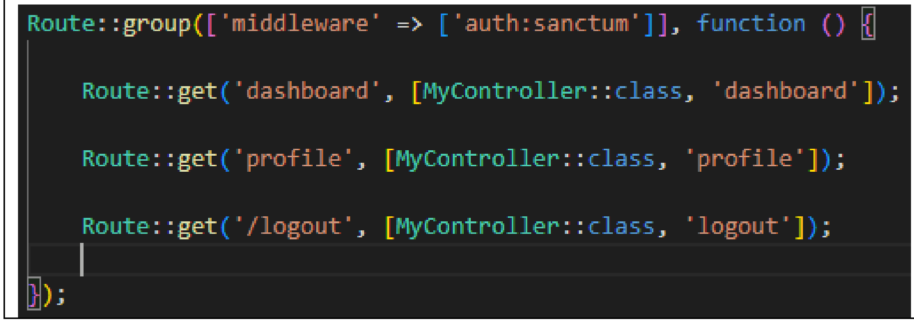
Status: 200 OK Time: 354 ms Size: 435 B

```
{
    "message": "Login success, Hi Ricky Satria,",
    "access_token": "3|WpiolQYhqNcYZUulzaWiD2Mx70FzGGPucWjytREe",
    "token_type": "Bearer"
}
```

3. Dashboard

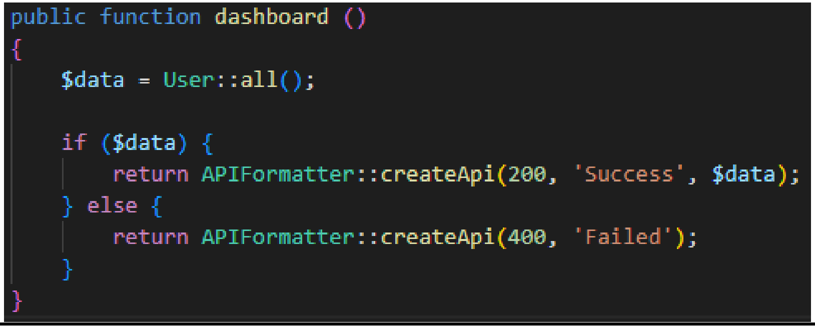
Code:

Route



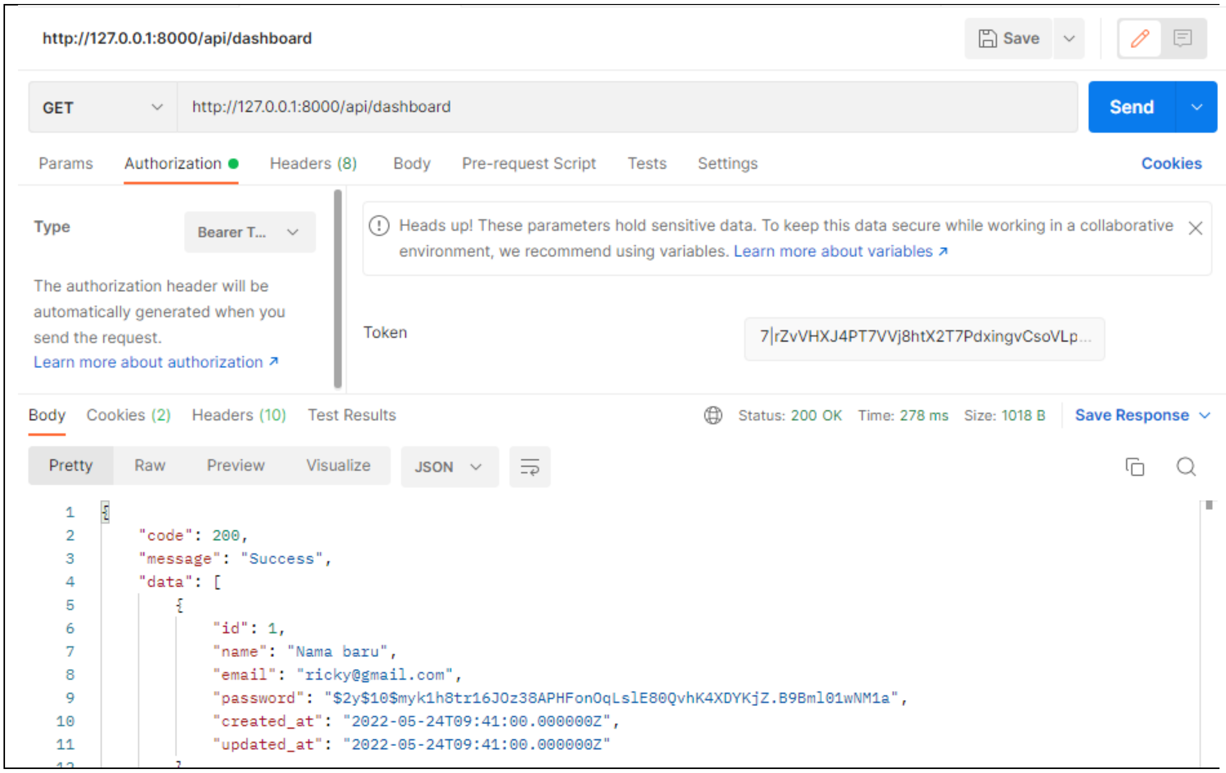
```
Route::group(['middleware' => ['auth:sanctum']], function () {

    Route::get('dashboard', [MyController::class, 'dashboard']);

    Route::get('profile', [MyController::class, 'profile']);

    Route::get('/logout', [MyController::class, 'logout']);

});
```

Function dashboard in MyController



```
public function dashboard ()
{
    $data = User::all();

    if ($data) {
        return APIFormatter::createApi(200, 'Success', $data);
    } else {
        return APIFormatter::createApi(400, 'Failed');
    }
}
```

URL and Output:



http://127.0.0.1:8000/api/dashboard

GET http://127.0.0.1:8000/api/dashboard

Type: Bearer T...

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

Token: 7|rZvVHXJ4PT7VVj8htX2T7PdxingvCsoVLp...

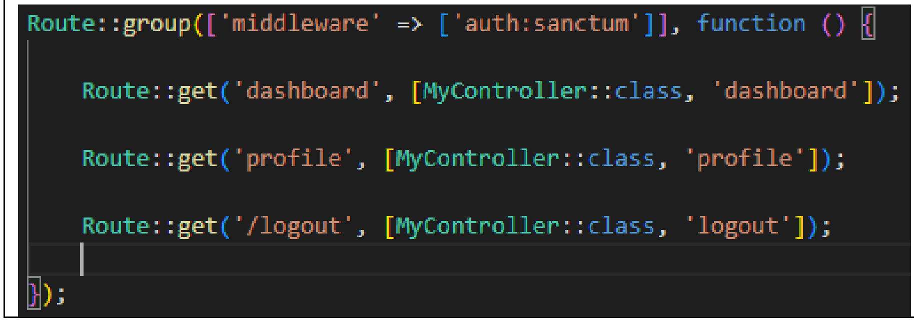
Status: 200 OK   Time: 278 ms   Size: 1018 B

```
{
    "code": 200,
    "message": "Success",
    "data": [
        {
            "id": 1,
            "name": "Nama baru",
            "email": "ricky@gmail.com",
            "password": "$2y$10$myk1h8tr16JOz38APHFonOqLslE80QvhK4XDYKjZ.B9Bml01wNM1a",
            "created_at": "2022-05-24T09:41:00.000000Z",
            "updated_at": "2022-05-24T09:41:00.000000Z"
```

4. Profile dan logout

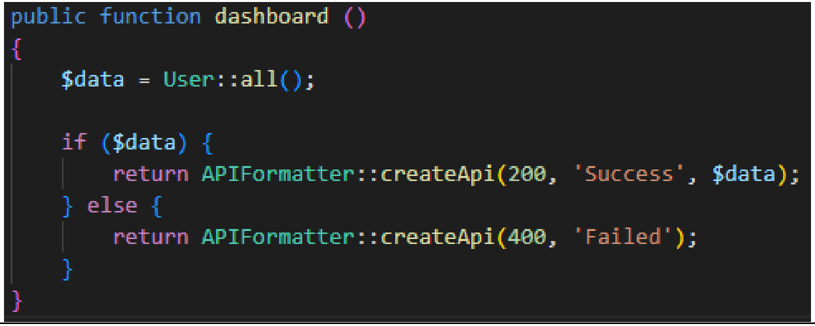
Code:

Route



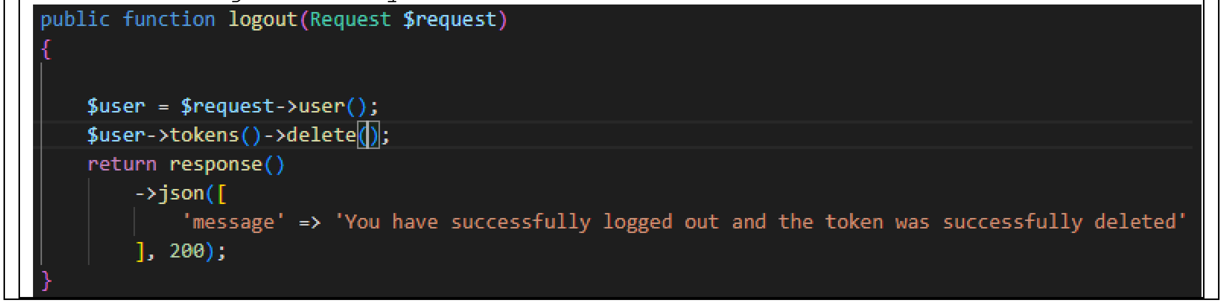
```
Route::group(['middleware' => ['auth:sanctum']], function () {

    Route::get('dashboard', [MyController::class, 'dashboard']);

    Route::get('profile', [MyController::class, 'profile']);

    Route::get('/logout', [MyController::class, 'logout']);

});
```

Function profile in MyController



```
public function dashboard ()
{
    $data = User::all();

    if ($data) {
        return APIFormatter::createApi(200, 'Success', $data);
    } else {
        return APIFormatter::createApi(400, 'Failed');
    }
}
```

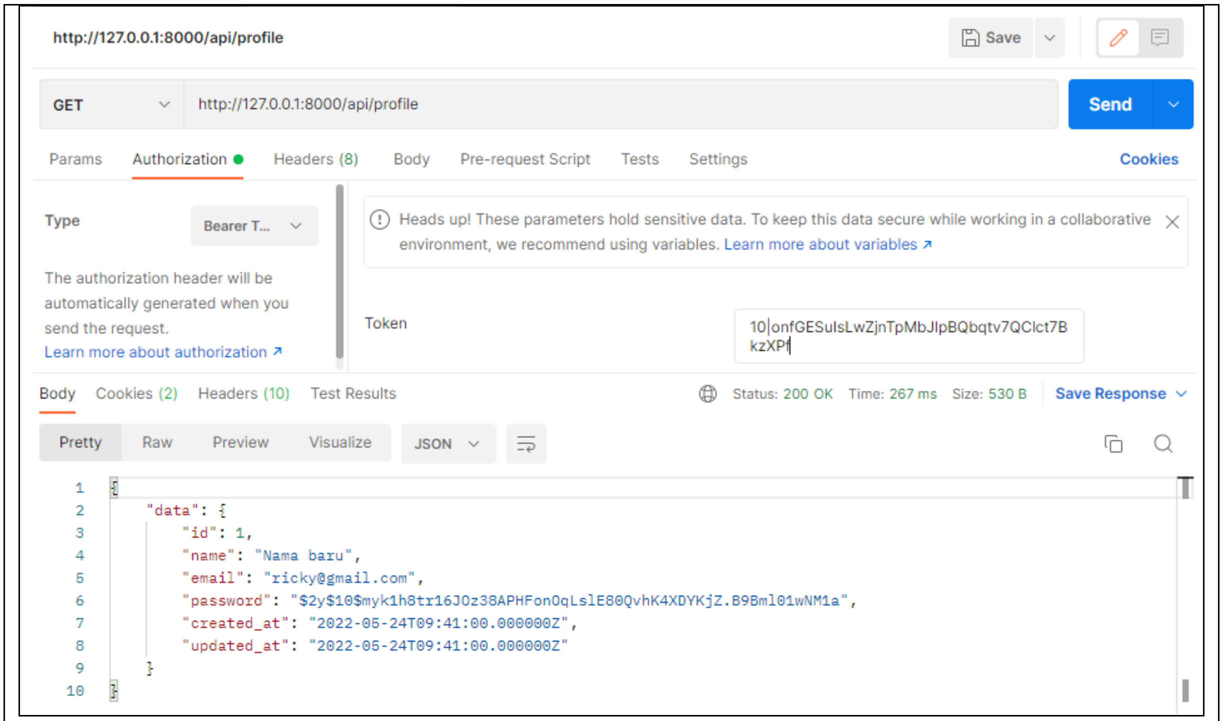
Function logout in MyController



```
public function logout(Request $request)
{

    $user = $request->user();
    $user->tokens()->delete();
    return response()
        ->json([
            'message' => 'You have successfully logged out and the token was successfully deleted'
        ], 200);
}
```

URL and Output:

Output Profile



http://127.0.0.1:8000/api/profile

GET http://127.0.0.1:8000/api/profile

Type: Bearer T...

The authorization header will be automatically generated when you send the request.
Learn more about authorization ↗

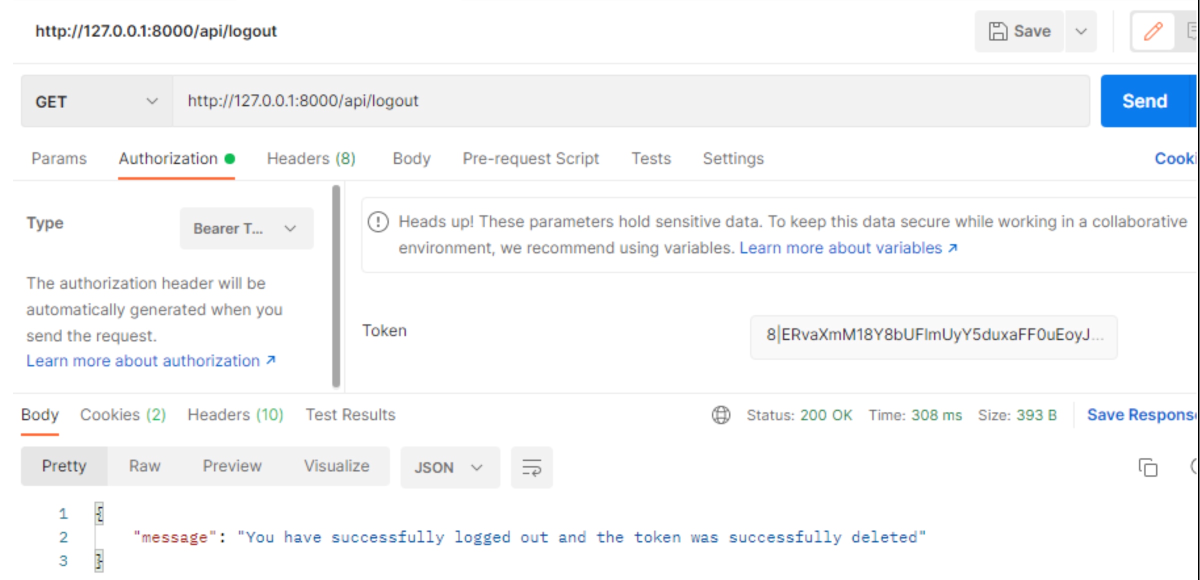
Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

Token: 10|onfGESulsLwZjnTpMbJlpBQbqtv7QClct7BkzXPf

Status: 200 OK Time: 267 ms Size: 530 B

```
{
    "data": {
        "id": 1,
        "name": "Nama baru",
        "email": "ricky@gmail.com",
        "password": "$2y$10$myk1h8tr16J0z38APHFon0qLslE80QvhK4XDYKjZ.B9Bml01wNM1a",
        "created_at": "2022-05-24T09:41:00.000000Z",
        "updated_at": "2022-05-24T09:41:00.000000Z"
    }
}
```

Output Logout



http://127.0.0.1:8000/api/logout

GET http://127.0.0.1:8000/api/logout

Type: Bearer T...

The authorization header will be automatically generated when you send the request.
Learn more about authorization ↗

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

Token: 8|ERvaXmM18Y8bUFlmUyY5duxaFF0uEoyJ...

Status: 200 OK Time: 308 ms Size: 393 B

```
{
    "message": "You have successfully logged out and the token was successfully deleted"
}
```