API Token Authentication in Laravel 5.2 & 5.3

by **JacobBennett**

I recently had the need to write a small url shortening application. I am aware that this problem has been solved quite a few times before, but what is being a developer if not reinventing the wheel just for the heck of it? Custom CMS anyone?

Knowing that this was going to be a tiny RESTful API and also knowing that Laravel 5.2 had API rate limiting built in, I was eager to give it a try.
Taylor Otwell being Taylor Otwell shipped 5.2 with the rate limiting defaults set up out of the box and I had my application building out short url's in a matter of minutes.
The problem for me came when I wanted to start associating those short urls with a user.

Typically my applications have a UI and authentication is done through a simple login page. Obviously for a RESTful API, having a login page isn't ideal. Instead, my hope was to have users append an `api_token` to the end of their query string and use that to authenticate their request.
I was happy to find that 5.2 also ships with a `TokenGuard`(link) class that allows you to do exactly that, but the documentation on getting it to work was a bit thin, so here you go.

Set up Token based Authenticaton

1. Add an api_token

The first think you need to do is to add an `api_token` column to your `users` table. If you are just starting your application you can likely get away with modifying the user migration that ships with Laravel to include your new column.

```
// add this to your users_table migration
```

```
        $table->string('api_token', 60)->unique();
```

Note: Be sure to generate and assign an api_token to new users. Something like `str_random(60)` should be sufficient.

2. Wrap your routes

Second, we need to make sure that any routes that will be using Token Authentication are being protected by the `auth:api` middleware.
Use the following route group as an example of what your routes might look like.

```
        Route::group(['prefix' => 'api/v1', 'middleware' =>
'auth:api'], function () {

        Route::post('/short', 'UrlMapperController@store');

        });
```

Note: Typically when protecting routes from unauthenticated users, we use the `auth` middleware, but by appending `:api` to the end we are telling Laravel that we want to use the driver for the `api` guard which is set up in the `config/auth.php` and is defaulted to `token`.

At this point, any routes wrapped with your `auth:api` middleware are only accessible to those that visit the route with a valid `api_token` in their request.

3. Getting the User

To get the authenticated user for this API request, use the following snippet:

```
        Auth::guard('api')->user();
```

Just like when we called the middleware, we have to let Laravel know that we want the `api` guard instead of the default `web` guard.

Extras

In the `App\Http\Middleware\Authenticate` middleware, you might want to change the following lines:

Update: This has been merged into 5.2. Check out the current `Authenticate` Middleware here

```
// Original

    if ($request->ajax()) {

        return response('Unauthorized.', 401);

    } else {

        return redirect()->guest('login');

    }


// Updated

    if ($request->ajax() || $request->wantsJson()) {

        return response('Unauthorized.', 401);

    } else {

        return redirect()->guest('login');

    }
```

This will return a 401 status code to unauthorized API requests instead of redirect it to a login page.

Lastly, if you are planning on primarily using the `TokenGuard` to authenticate requests, change the default guard in `config/auth.php` to be `api` instead of `web`. That should prevent you from having to tell Laravel to use the `api` version of the middleware or guard since Laravel will use by default what you have set in `config/auth.php`.

Wrapping Up

Well there you have it, authenticating users to your api using nothing more than an `api_token` in the request and an `api_token` column on your user table. Hopefully this will save you the time it took me digging through the `TokenGuard` and Issues to figure it out.

Created 3 years ago | Updated 3 months ago
View on GitHub