

The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on the left side are several concentric circles and arcs, some with degree markings (40, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260) and arrows, suggesting a circular or orbital theme.

INTRODUCTION TO LARAVEL PASSPORT

@MICHAELPEACOCK

PHP NORTH EAST, SEPTEMBER 2016

The background is a dark blue gradient with a subtle pattern of white dots, resembling a starry sky. Overlaid on this are several faint, light blue technical diagrams. In the top right, there is a large circular gauge with concentric circles and radial tick marks, some labeled with numbers like 100, 120, 140, 160, 180, and 200. In the bottom right, there is a smaller circular diagram with concentric circles and arrows indicating a clockwise direction. In the bottom left, there is a partial view of another circular diagram with an arrow. The overall aesthetic is technical and modern.

@MICHAELPEACOCK

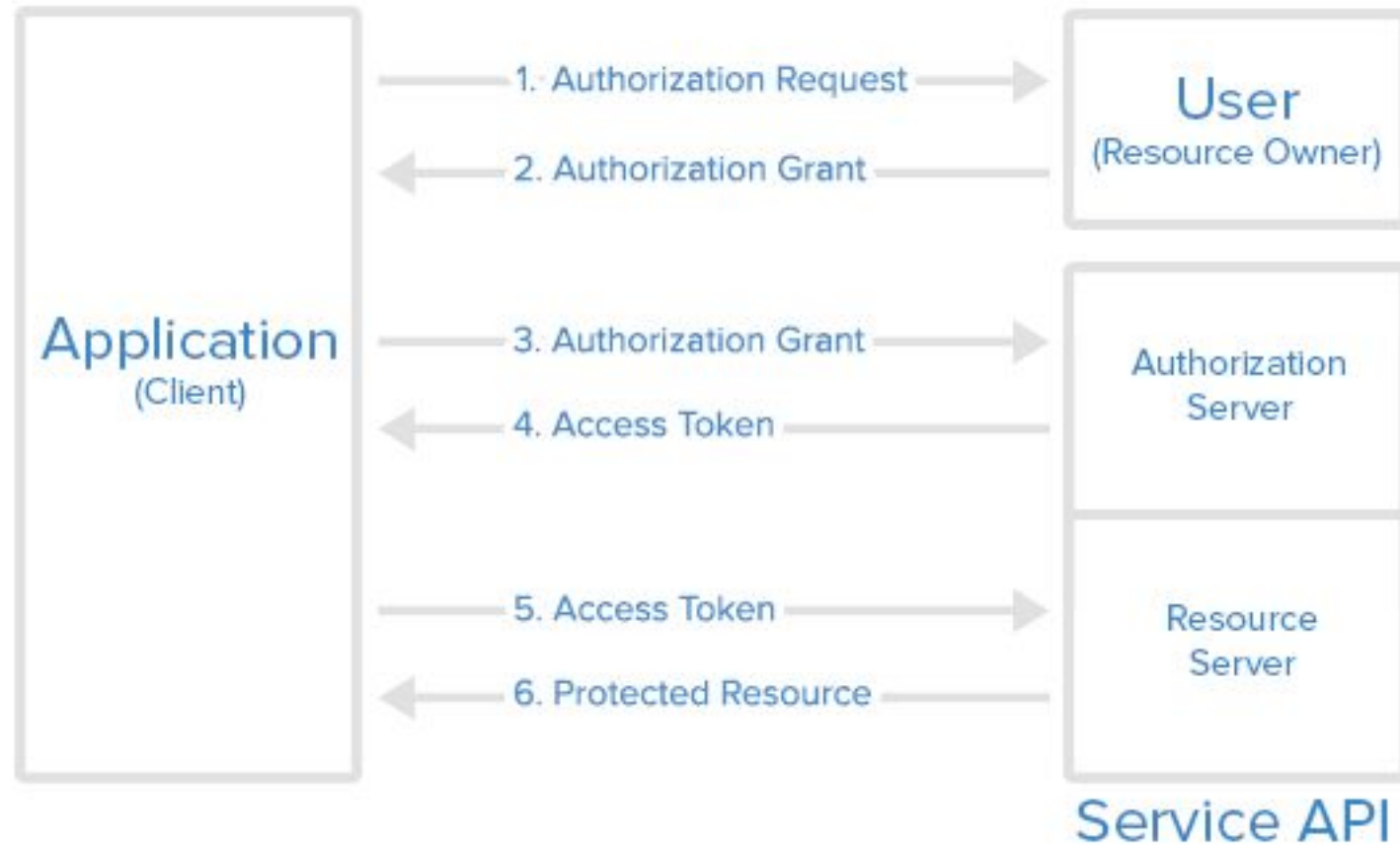
- Co-organiser of PHP North East & Laravel North East
- Software development consultant
- Primarily serve as a consultant CTO for a number of early stage startups

INTRODUCTION

- API Authentication
- Uses OAuth
- Built on top of the PHP League OAuth 2 Server

OAUTH?

Abstract Protocol Flow



SSL!

- OAuth 2 specification mandates that the authorization server uses SSL

INSTALLATION

1. Require laravel passport

```
composer require laravel/passport
```

2. Register passport service provider in config/app.php

```
Laravel\Passport\PassportServiceProvider::class,
```

3. Migrate: `php artisan migrate`

4. Install: `php artisan passport:install`

SETUP

1. **Add** `Laravel\Passport\HasApiTokens` **trait** to **User** class
2. **Call** `Laravel\Passport\Passport::routes()` **from** `AuthServiceProvider::boot` **method**
3. **Change the driver** for the api authentication guard to `passport` **in** `config/auth.php`

OAUTH ROUTES OUT OF THE BOX

| Method | URL | Description |
|--------|-------------------------------|--|
| GET | /oauth/clients | List clients / apps you have created |
| POST | /oauth/clients | Create a new client / app. Requires name and redirect. |
| PUT | /oauth/clients/{id} | Update client. Requires both name and redirect |
| DELETE | /oauth/clients/{id} | Delete a client. |
| GET | /oauth/authorize | Start the oAuth process, displays the accept / cancel dialogue |
| POST | /oauth/authorize | Accept the oAuth process |
| POST | /oauth/token | Exchange a code for a token or refresh a token |
| GET | /oauth/scopes | List all scopes |
| GET | /oauth/personal-access-tokens | List all personal access tokens |
| POST | /oauth/personal-access-tokens | Request a personal access token (name and scopes required) |
| DELETE | /oauth/personal-access- | Delete a personal access token |

PUBLISH VUE COMPONENTS & VIEWS

- `php artisan vendor:publish --tag=passport-components`
- `php artisan vendor:publish --tag=passport-views`

VUE INTEGRATION

```
Vue.component(  
  'passport-clients',  
  require('./components/passport/Clients.vue')  
);
```

```
Vue.component(  
  'passport-authorized-clients',  
  require('./components/passport/AuthorizedClients.vue')  
);
```

```
Vue.component(  
  'passport-personal-access-tokens',  
  require('./components/passport/PersonalAccessTokens.vue')  
);
```

...DON'T FORGET TO GULP

- (requires an npm install first!)

USE THE VUE COMPONENTS

- `<passport-clients></passport-clients>`
- `<passport-authorized-clients></passport-authorized-clients>`
- `<passport-personal-access-tokens></passport-personal-access-tokens>`

Dashboard

You are logged in!

OAuth Clients

[Create New Client](#)

| Client ID | Name | Secret | |
|-----------|------------|--|---|
| 3 | App Client | 9Ze2bt13P5MSmSgmFmzLdweW7BM4r8wvpnlWnxZH | Edit Delete |

Authorized Applications

| Name | Scopes | |
|------------|--------------|------------------------|
| App Client | write, email | Revoke |

Personal Access Tokens

[Create New Token](#)

| Name | |
|------------------|------------------------|
| A Personal Token | Delete |

DEMO 1

- VUE components

AUTHORIZE

```
Route::get('/connect', function () {  
    $query = http_build_query([  
        'client_id' => '3',  
        'redirect_uri' => 'http://...t.local/redirect',  
        'response_type' => 'code',  
        'scope' => '',  
    ]);  
  
    return redirect('http://...t.local/oauth/authorize?'.$query);  
});
```

EXCHANGE CODE FOR AN ACCESS TOKEN

```
use Illuminate\Http\Request;
```

```
Route::get('/redirect', function (Request $request) {  
    $http = new GuzzleHttp\Client;  
    $response = $http->post('http://...t.local/oauth/token', [  
        'form_params' => [  
            'grant_type' => 'authorization_code',  
            'client_id' => '3',  
            'client_secret' => '9Ze2bt13P5MSmSgmFmzLdweW7BM4r8wvpnlWnxZH',  
            'redirect_uri' => 'http://...t.local/redirect',  
            'code' => $request->query->get('code'),  
        ],  
    ]);  
  
    return json_decode((string) $response->getBody(), true);  
});
```

DEMO 2

- oAuth handshake

AUTHENTICATE

- Accept: application/json
- Authorization: Bearer bearer-token

REQUIRE A VALID ACCESS TOKEN

- ->middleware('auth:api')

DEFINING SCOPES

```
// AuthServiceProvider
Laravel\Passport\Passport::tokensCan([
    'read' => 'Read access to user account',
    'write' => 'Write access to user account',
    'email' => 'Send emails',
]);
```


REGISTER SCOPE MIDDLEWARE

```
// Http/Kernel.php Route Middleware
```

```
'scopes' => \Laravel\Passport\Http\Middleware\CheckScopes::class,  
'scope' => \Laravel\Passport\Http\Middleware\CheckForAnyScope::class,
```

REQUIRE ANY OF A LIST OF SCOPES

```
->middleware('scope:read,write');
```

REQUIRE A NUMBER OF SCOPES

```
->middleware('scopes:write,email');
```


DEMO

- Scopes

PASSWORD GRANT

- Enabled out of the box when we did `php artisan passport:install`
- Can be enabled via `php artisan passport:client --password`

```
Route::get('/connect-password-grant', function () {
    $http = new GuzzleHttp\Client;
    $response = $http->post(
        'http://laravel-passport.local/oauth/token',
        [
            'form_params' => [
                'grant_type' => 'password',
                'client_id' => '2',
                'client_secret' => 'xIQoQPimqpdVXQiRU81wyRa78X2mnSxSY9CD38EC',
                'username' => 'mkpeacock@gmail.com',
                'password' => 'password',
                'scope' => 'write email',
            ],
        ],
    );

    return json_decode((string) $response->getBody(), true);
});
```

DEMO

- Password grant

AUTOMATIC TOKENS FOR WEB REQUESTS

- Laravel can expose tokens (personal tokens) automatically for web users.
- Makes it easy for your application to communicate directly with its API
- Register to the web middleware group:
- `\Laravel\Passport\Http\Middleware\CreateFreshApiToken::class,`

DEMO

- Tokens for web users

CONCLUSION

- Getting an OAuth 2 server running with Laravel 5.3 is pretty easy
- The install and config is a little clunky and involves a few steps
- But with the amount of work that's done out of the box – its all good!

THANKS, QUESTIONS?

<https://bitbucket.org/mkpeacock/laravel-passport-demo>