Software Security

IE5042 – Assignment 2

PATRICK ROY

Contents

[Introduction 2](#_Toc70508648)

[Technical Specification 2](#_Toc70508649)

[Approach 2](#_Toc70508650)

[UML Sequence Diagram 3](#_Toc70508651)

[Detail description of codes 4](#_Toc70508652)

[Use of Laraval Libraries 4](#_Toc70508653)

[Call back from OAuth Service 5](#_Toc70508654)

[Validate the session for token 6](#_Toc70508655)

[Retrieve files and folders 7](#_Toc70508656)

[Display the dropbox contents 7](#_Toc70508657)

[Appendix – 1 8](#_Toc70508658)

[Appendix – 2 9](#_Toc70508659)

[Appendix – 3 10](#_Toc70508660)

[Appendix – 4 10](#_Toc70508661)

[Appendix – 5 11](#_Toc70508662)

# Introduction

This document details the use of OAuth access token to interact with dropbox to retrieve files & folders and to list it on client end. In this web application it is demonstrated how you could use Laraval framework to explore its fullest capabilities to achieve the task.

# Technical Specification

Web programming language : Laravel framework 8.0 |

<https://laravel.com/docs/8.x>

OAuth Server : Dropbox OAuth server | <https://socialiteproviders.com/Dropbox/>

# Approach

The app is built on Laravel framework 8.0 and Laravel eco system has providers for OAuth services such as Dropbox. Have chosen the Dropbox features to celebrate the functionality , and the OAuth authentication is used to access the token. Then, it was used Dropbox API to get the files and folder list.

​

<https://www.dropbox.com/developers/documentation/http/documentation>

​

# UML Sequence Diagram

Application

Description automatically generated with medium confidence

# Detail description of codes

# Use of Laraval Libraries

1. Used Laravel and two ecosystem libraries to integrate OAuth with Dropbox and use Dropbox HTTP API.

{

"name": "laravel/laravel",

"type": "project",

"description": "The Laravel Framework.",

"keywords": [

"framework",

"laravel"

],

"license": "MIT",

"require": {

"php": "^7.3|^8.0",

"fideloper/proxy": "^4.4",

"fruitcake/laravel-cors": "^2.0",

"guzzlehttp/guzzle": "^7.0.1",

"laravel/framework": "^8.12",

"laravel/tinker": "^2.5",

"socialiteproviders/dropbox": "^4.2",

"spatie/dropbox-api": "^1.17"

},

"require-dev": {

"facade/ignition": "^2.5",

"fakerphp/faker": "^1.9.1",

"laravel/sail": "^1.0.1",

"mockery/mockery": "^1.4.2",

"nunomaduro/collision": "^5.0",

"phpunit/phpunit": "^9.3.3"

},

"config": {

"optimize-autoloader": true,

"preferred-install": "dist",

"sort-packages": true

},

"extra": {

"laravel": {

"dont-discover": []

}

},

"autoload": {

"psr-4": {

"App\\": "app/",

"Database\\Factories\\": "database/factories/",

"Database\\Seeders\\": "database/seeders/"

}

},

"autoload-dev": {

"psr-4": {

"Tests\\": "tests/"

}

},

"minimum-stability": "dev",

"prefer-stable": true,

"scripts": {

"post-autoload-dump": [

"Illuminate\\Foundation\\ComposerScripts::postAutoloadDump",

"@php artisan package:discover --ansi"

],

"post-root-package-install": [

"@php -r \"file\_exists('.env') || copy('.env.example', '.env');\""

],

"post-create-project-cmd": [

"@php artisan key:generate --ansi"

]

}

}

# Call back from OAuth Service

1. Route is the entry point of the Laravel, and a route is defined to handle call back from OAuth service (Dropbox), as for this demo purpose, define the logic in closure of the method. And it just gets the user details from the service with token and store into session. (In typical we store this into database and use in future requests). Then we redirect to the root URL. The root (/) is protected with Dropbox middleware.

Route::get('/', [DocumentController::class, 'index'])->middleware('dropbox');

Route::get('oauth/dropbox/callback', function () {

$user = Socialite::driver('dropbox')->user();

session()->push('user.token', $user->token);

session()->push('user.name', $user->getName());

session()->push('user.email', $user->getEmail());

return redirect()->to('/');

});

# Validate the session for token

1. This middleware always executed and validated the session and if the session doesn’t have the token, then redirect to the Dropbox authentication URL (Where the user login to the Dropbox and authorize the app to communicate each other).

If the session already has the token, then it will call the DocumentController::index method.

class DropboxAuth

{

/\*\*

\* Handle an incoming request.

\*

\* @param \Illuminate\Http\Request $request

\* @param \Closure $next

\* @return mixed

\*/

public function handle(Request $request, Closure $next)

{

if (!$request->session()->has('user')) {

return Socialite::driver('dropbox')

->scopes(['account\_info.read', 'files.metadata.write'])

->redirect();

}

return $next($request);

}

}

# Retrieve files and folders

1. This method gets the files and folders from the root path initially from the Dropbox and list in the HTML list.

class DocumentController extends Controller

{

public function index()

{

$authorizationCode = session()->get('user.token.0');

$client = new Client($authorizationCode);

$folder = \request()->input('folder', '/');

$lists = $client->listFolder($folder);

$lists = Arr::get($lists, 'entries', []);

return view('welcome', compact('lists'));

}

}

# Display the dropbox contents

1. And it link to the path if the type is folder, and allow user to click and show the contents (sub folder or files).

@foreach($lists as $list)

<tr>

<td>

@if($list['.tag'] == 'folder')

<a href="/?folder={{ $list['path\_lower'] }}">{{ $list['name'] }}</a>

@else

{{ $list['name'] }}

@endif

</td>

</tr>

@endforeach

# Appendix – 1

https://github.com/MS21917202/oauth-submission/blob/master/composer.json

{

"name": "laravel/laravel",

"type": "project",

"description": "The Laravel Framework.",

"keywords": [

"framework",

"laravel"

],

"license": "MIT",

"require": {

"php": "^7.3|^8.0",

"fideloper/proxy": "^4.4",

"fruitcake/laravel-cors": "^2.0",

"guzzlehttp/guzzle": "^7.0.1",

"laravel/framework": "^8.12",

"laravel/tinker": "^2.5",

"socialiteproviders/dropbox": "^4.2",

"spatie/dropbox-api": "^1.17"

},

"require-dev": {

"facade/ignition": "^2.5",

"fakerphp/faker": "^1.9.1",

"laravel/sail": "^1.0.1",

"mockery/mockery": "^1.4.2",

"nunomaduro/collision": "^5.0",

"phpunit/phpunit": "^9.3.3"

},

"config": {

"optimize-autoloader": true,

"preferred-install": "dist",

"sort-packages": true

},

"extra": {

"laravel": {

"dont-discover": []

}

},

"autoload": {

"psr-4": {

"App\\": "app/",

"Database\\Factories\\": "database/factories/",

"Database\\Seeders\\": "database/seeders/"

}

},

"autoload-dev": {

"psr-4": {

"Tests\\": "tests/"

}

},

"minimum-stability": "dev",

"prefer-stable": true,

"scripts": {

"post-autoload-dump": [

"Illuminate\\Foundation\\ComposerScripts::postAutoloadDump",

"@php artisan package:discover --ansi"

],

"post-root-package-install": [

"@php -r \"file\_exists('.env') || copy('.env.example', '.env');\""

],

"post-create-project-cmd": [

"@php artisan key:generate --ansi"

]

}

}

# Appendix – 2

https://github.com/MS21917202/oauth-submission/blob/master/routes/web.php#L20

<?php

use App\Http\Controllers\DocumentController;

use Illuminate\Support\Facades\Route;

use Laravel\Socialite\Facades\Socialite;

/\*

|--------------------------------------------------------------------------

| Web Routes

|--------------------------------------------------------------------------

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

|

\*/

Route::get('/', [DocumentController::class, 'index'])->middleware('dropbox');

Route::get('oauth/dropbox/callback', function () {

$user = Socialite::driver('dropbox')->user();

session()->push('user.token', $user->token);

session()->push('user.name', $user->getName());

session()->push('user.email', $user->getEmail());

return redirect()->to('/');

});

# Appendix – 3

https://github.com/MS21917202/oauth-submission/blob/master/app/Http/Middleware/DropboxAuth.php#L20

<?php

namespace App\Http\Middleware;

use Closure;

use Illuminate\Http\Request;

use Laravel\Socialite\Facades\Socialite;

class DropboxAuth

{

/\*\*

\* Handle an incoming request.

\*

\* @param \Illuminate\Http\Request $request

\* @param \Closure $next

\* @return mixed

\*/

public function handle(Request $request, Closure $next)

{

if (!$request->session()->has('user')) {

return Socialite::driver('dropbox')

->scopes(['account\_info.read', 'files.metadata.write'])

->redirect();

}

return $next($request);

}

}

# Appendix – 4

https://github.com/MS21917202/oauth-submission/blob/master/app/Http/Controllers/DocumentController.php#L11

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Arr;

use Spatie\Dropbox\Client;

class DocumentController extends Controller

{

public function index()

{

$authorizationCode = session()->get('user.token.0');

$client = new Client($authorizationCode);

$folder = \request()->input('folder', '/');

$lists = $client->listFolder($folder);

$lists = Arr::get($lists, 'entries', []);

return view('welcome', compact('lists'));

}

}

# Appendix -5

https://github.com/MS21917202/oauth-submission/blob/master/resources/views/welcome.blade.php#L399

<!DOCTYPE html>

<html lang="{{ str\_replace('\_', '-', app()->getLocale()) }}">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Laravel</title>

<!-- Fonts -->

<link href="https://fonts.googleapis.com/css2?family=Nunito:wght@400;600;700&display=swap" rel="stylesheet">

<!-- Styles -->

<style>

/\*! normalize.css v8.0.1 | MIT License | github.com/necolas/normalize.css \*/

html {

line-height: 1.15;

-webkit-text-size-adjust: 100%

}

body {

margin: 0

}

a {

background-color: transparent

}

[hidden] {

display: none

}

html {

font-family: system-ui, -apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Helvetica Neue, Arial, Noto Sans, sans-serif, Apple Color Emoji, Segoe UI Emoji, Segoe UI Symbol, Noto Color Emoji;

line-height: 1.5

}

\*, :after, :before {

box-sizing: border-box;

border: 0 solid #e2e8f0

}

a {

color: inherit;

text-decoration: inherit

}

svg, video {

display: block;

vertical-align: middle

}

video {

max-width: 100%;

height: auto

}

.bg-white {

--bg-opacity: 1;

background-color: #fff;

background-color: rgba(255, 255, 255, var(--bg-opacity))

}

.bg-gray-100 {

--bg-opacity: 1;

background-color: #f7fafc;

background-color: rgba(247, 250, 252, var(--bg-opacity))

}

.border-gray-200 {

--border-opacity: 1;

border-color: #edf2f7;

border-color: rgba(237, 242, 247, var(--border-opacity))

}

.border-t {

border-top-width: 1px

}

.flex {

display: flex

}

.grid {

display: grid

}

.hidden {

display: none

}

.items-center {

align-items: center

}

.justify-center {

justify-content: center

}

.font-semibold {

font-weight: 600

}

.h-5 {

height: 1.25rem

}

.h-8 {

height: 2rem

}

.h-16 {

height: 4rem

}

.text-sm {

font-size: .875rem

}

.text-lg {

font-size: 1.125rem

}

.leading-7 {

line-height: 1.75rem

}

.mx-auto {

margin-left: auto;

margin-right: auto

}

.ml-1 {

margin-left: .25rem

}

.mt-2 {

margin-top: .5rem

}

.mr-2 {

margin-right: .5rem

}

.ml-2 {

margin-left: .5rem

}

.mt-4 {

margin-top: 1rem

}

.ml-4 {

margin-left: 1rem

}

.mt-8 {

margin-top: 2rem

}

.ml-12 {

margin-left: 3rem

}

.-mt-px {

margin-top: -1px

}

.max-w-6xl {

max-width: 72rem

}

.min-h-screen {

min-height: 100vh

}

.overflow-hidden {

overflow: hidden

}

.p-6 {

padding: 1.5rem

}

.py-4 {

padding-top: 1rem;

padding-bottom: 1rem

}

.px-6 {

padding-left: 1.5rem;

padding-right: 1.5rem

}

.pt-8 {

padding-top: 2rem

}

.fixed {

position: fixed

}

.relative {

position: relative

}

.top-0 {

top: 0

}

.right-0 {

right: 0

}

.shadow {

box-shadow: 0 1px 3px 0 rgba(0, 0, 0, .1), 0 1px 2px 0 rgba(0, 0, 0, .06)

}

.text-center {

text-align: center

}

.text-gray-200 {

--text-opacity: 1;

color: #edf2f7;

color: rgba(237, 242, 247, var(--text-opacity))

}

.text-gray-300 {

--text-opacity: 1;

color: #e2e8f0;

color: rgba(226, 232, 240, var(--text-opacity))

}

.text-gray-400 {

--text-opacity: 1;

color: #cbd5e0;

color: rgba(203, 213, 224, var(--text-opacity))

}

.text-gray-500 {

--text-opacity: 1;

color: #a0aec0;

color: rgba(160, 174, 192, var(--text-opacity))

}

.text-gray-600 {

--text-opacity: 1;

color: #718096;

color: rgba(113, 128, 150, var(--text-opacity))

}

.text-gray-700 {

--text-opacity: 1;

color: #4a5568;

color: rgba(74, 85, 104, var(--text-opacity))

}

.text-gray-900 {

--text-opacity: 1;

color: #1a202c;

color: rgba(26, 32, 44, var(--text-opacity))

}

.underline {

text-decoration: underline

}

.antialiased {

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale

}

.w-5 {

width: 1.25rem

}

.w-8 {

width: 2rem

}

.w-auto {

width: auto

}

.grid-cols-1 {

grid-template-columns:repeat(1, minmax(0, 1fr))

}

@media (min-width: 640px) {

.sm\:rounded-lg {

border-radius: .5rem

}

.sm\:block {

display: block

}

.sm\:items-center {

align-items: center

}

.sm\:justify-start {

justify-content: flex-start

}

.sm\:justify-between {

justify-content: space-between

}

.sm\:h-20 {

height: 5rem

}

.sm\:ml-0 {

margin-left: 0

}

.sm\:px-6 {

padding-left: 1.5rem;

padding-right: 1.5rem

}

.sm\:pt-0 {

padding-top: 0

}

.sm\:text-left {

text-align: left

}

.sm\:text-right {

text-align: right

}

}

@media (min-width: 768px) {

.md\:border-t-0 {

border-top-width: 0

}

.md\:border-l {

border-left-width: 1px

}

.md\:grid-cols-2 {

grid-template-columns:repeat(2, minmax(0, 1fr))

}

}

@media (min-width: 1024px) {

.lg\:px-8 {

padding-left: 2rem;

padding-right: 2rem

}

}

@media (prefers-color-scheme: dark) {

.dark\:bg-gray-800 {

--bg-opacity: 1;

background-color: #2d3748;

background-color: rgba(45, 55, 72, var(--bg-opacity))

}

.dark\:bg-gray-900 {

--bg-opacity: 1;

background-color: #1a202c;

background-color: rgba(26, 32, 44, var(--bg-opacity))

}

.dark\:border-gray-700 {

--border-opacity: 1;

border-color: #4a5568;

border-color: rgba(74, 85, 104, var(--border-opacity))

}

.dark\:text-white {

--text-opacity: 1;

color: #fff;

color: rgba(255, 255, 255, var(--text-opacity))

}

.dark\:text-gray-400 {

--text-opacity: 1;

color: #cbd5e0;

color: rgba(203, 213, 224, var(--text-opacity))

}

}

</style>

<style>

body {

font-family: 'Nunito', sans-serif;

}

</style>

</head>

<body class="antialiased">

<table>

<tbody>

@foreach($lists as $list)

<tr>

<td>

@if($list['.tag'] == 'folder')

<a href="/?folder={{ $list['path\_lower'] }}">{{ $list['name'] }}</a>

@else

{{ $list['name'] }}

@endif

</td>

</tr>

@endforeach

</tbody>

</table>

</body>

</html>