3x4 Assessment Documentation

Prepared for: Stewart Reid

Prepared by: Josh Harington

07 June 2021

EXECUTIVE SUMMARY

Objective

Build a web based portal to display a list of users with the ability to create and update them. Provide the ability to swap out the data source without any changes in the code.

Framework

The framework that was used is Laravel 5.8, due to the restriction of using PHP 7.2. Had the PHP restriction been 7.3+ I would have used the latest Laravel 8 version. VueJS was used for the frontend.

Environment requirements

To setup the app to run in your own environment you will need:
- PHP 7.2
- NPM 6.14.11
- Composer

Download the project

To download the project from Github, enter the following into your terminal:
NB: Make sure the terminal is pointing to your desired host directory, ie: Documents/Github is the folder that contains all the GitHub projects that I work on.
- git clone https://github.com/joshharington/multi-data-source-assessment.git
- cd multi-data-source-assessment

Pre-Install

Before you can setup the code and run it, you will need to have a database already setup and configured. You can create a MySQL database on your local machine and specify the connection details in the .env file inside the root project folder (/multi-data-source-assessment). There are also other ENV keys that are needed

The keys you will need to update are:
- DB_DATABASE
- DB_USERNAME
- DB_PASSWORD

The keys you will need to add are:
- STORE_TYPE
  This will be either JSON or DB
- DATASTORE_JSON_FILENAME
  This is the name of the JSON file in **storage/app** to be used as the alternate data store

Install & run the project

To install and the project, enter the following into your terminal:

NB: Make sure the terminal is pointing to your desired directory, ie: make sure you run these commands from the multi-data-source-assessment directory.

- composer update
- npm install
- php artisan key:generate
- php artisan vendor:publish —all
- php artisan migrate
- php artisan db:seed
- npm run dev

NB: If you used the sql dump to create the database, it already has the tables and data, so you don't need to run the *php artisan migrate* and *php artisan db:seed* commands

Once the above commands have all completed, you will be ready to run and access the application. To run the app, enter the following into your terminal:

php artisan serve

You should now be able to access the application in your browser at: http://localhost:8000

API Setup

The system is built API-first, so the frontend us consuming the API. This is built this way, so that if we wanted any other frontend to be implemented, there are no changes required for the logic.

For authenticating users on the API, you are required to update the Passport keys. In your terminal (from the route of the project) run:

php artisan passport:install

That will generate a password grant client (It generates 2 keys, a Personal Access Grant Client and a Password Grant Client) that you can find in the **oauth_clients** table in the database. Take the Password Grant Client secret that was generated, and replace the current value in resources/js/components/auth/LoginComponent.vue in the axios.post method call where there is a **client_secret** key.

NB: You may need to set the client_id as well.

Once you have changed the **client_secret** in the VueJS component, you will need to run **npm run dev** so that the changes reflect.

Login details

When seeding the database (php artisan db:seed) it created a demo user account that you can log in with:
Username: **josh1@live.co.za**
Password: **12345678a!**

Convert data from truth

Although not part of the requirements, I also included a command that can convert the data from the DB to JSON or from JSON to the DB. This does overwrite the data source that you are converting to, so be careful when running it. The command is:
php artisan data:convert-from {from}

You can specify what you want in place of the {from} key above. If you enter DB, it will take the data in the database and overwrite the JSON datastore file with the data from the database. If you enter JSON, it will take the data in the JSON file, truncate the users table and insert the users into the database. The default password it uses for any of the users is **12345678a!**

Common Problems

Common issues that might arise are:

- database connection, make sure that the database connection has the correct info in the **.env** file.
- File permissions, you may need to change the file permissions of the storage and bootstrap directories. The correct permissions for these folders / files are: 755 for the folders and 644 for the files. For local development, you can give the folders a permission of 777 recursively - but NEVER do this on a live server. This will give anyone read, write and execute abilities in the directory, and is a major security vulnerability. You can find more info on this here: https://stackoverflow.com/questions/30639174/how-to-set-up-file-permissions-for-laravel