

# Laravel Backend Coding Simulation

## Trax Milage Tracking Application

---

### Overview

In this coding simulation you are being asked to develop the backend for 'Trax' - a milage tracking application developed using Laravel, Vue.js and Vuetify.js.

The front-end of the application has already been developed and consists of the following features:

- User registration and authentication (Note: No back-end work is required for user management. This work is already complete.)
- Ability for a logged in user to add a car to Trax
- Ability for a logged in user to remove a car from Trax
- Ability for a logged in user to view a list of all their cars
- Ability for a logged in user to add a trip to Trax specifying which car was used.
- Ability for a logged in user to a view a list of all their trips

The frontend of the application currently makes requests to a set of mock endpoints implemented in-line in `/routes/api.php`, with each mock endpoint sending back statically defined JSON objects. It is your task to develop a functional API for Trax.

You are being ask to:

- Design the Data-Model for Trax and create corresponding Database Migrations.
- Implement the Eloquent Model Layer
- Design a RESTful API to be consumed by the front-end using the mock end-points as a guide for the expected data format.
- Implement the Controller(s) for the API and leverage them in the routes.
- Update the frontend's `/resources/assets/js/traxAPI.js` file so that the frontend makes uses of your new API.

Ideally, you should showcase your Laravel experience by using as many Laravel specific features as possible. For example: Scopes, Resources, Policies, Unit Tests, etc.

---

## Getting Started

Perform the following steps to get started with the coding simulation.

- Install Docker Desktop from <https://www.docker.com/products/docker-desktop>
- Navigate to the root directory of the git repo and create a branch for your work.

```
> cd <trax repo root directory>
> git branch <your-last-name>
> git checkout <your-last-name>
> git push --set-upstream origin <your-last-name>
```

- Setup your working environment by executing the following commands

```
> cd <trax repo directory>
> git submodule update --init --recursive
> cp laradock-env laradock/.env
> cp createdb.sql laradock/mysql/docker-entrypoint-initdb.d/.
> cd laradock
> docker-compose build --no-cache nginx workspace mysql
> docker-compose up -d nginx mysql workspace
> docker-compose exec workspace composer install
> docker-compose exec workspace npm install
> docker-compose exec workspace php artisan migrate
```

At this point you can open <http://localhost:8888/> and start using the mock-API backed application. As a first step, you should click 'Register' in the upper right to create an account and enter the application.

Should you make changes to any of the JS files, such as `/resources/assets/js/traxAPI.js`, you can run the following in order to compile your changes

```
> cd <trax repo directory>/laradock
> docker-compose exec workspace npm run dev
```

Or to watch for any JS changes you can run

```
> cd <trax repo directory>/laradock
> docker-compose exec workspace npm run watch
```

**GOOD LUCK!**