

## **Dali Mock App - Setup**

A Laravel-based simulation system for managing and monitoring virtual “agents” across a geographic area. The system simulates a driver navigating through checkpoints, intersections, traffic lights, and highway points, each with status-based behavior.

### **Features**

- Manage agents with types: `checkpoint`, `intersection`, `traffic\_light`, `highway\_point`
- Simulate status-based behavior: green, red, yellow
- REST API to fetch agents and simulate movement
- Notification system that returns contextual messages
- Demo-ready: seeded with 50 geo-positioned agents

### **Requirements**

Make sure these are installed on your machine:

- PHP 8.4+: <https://www.php.net/>
- Composer: <https://getcomposer.org/>
- Laravel 12: <https://laravel.com/docs/12.x>
- SQLite3: <https://www.sqlite.org/download.html>
- Node.js (v22.14.0+): <https://nodejs.org/> + npm: <https://www.npmjs.com/>

### **Environment Variables**

In `.env`, set the following:

***GOOGLE\_MAPS\_API\_KEY=your\_google\_maps\_api\_key\_here***

***DB\_CONNECTION=sqlite***

***DB\_DATABASE=/absolute/path/to/your/project/database/database.sqlite***

### **Installation**

**Option 2: Use the ZIP**

- 1. Unzip the `dalisava.zip` file***
- 2. Move into the folder:***

**Option 2: Clone the Repo**

- 1. git clone https://github.com/elchroy/dalisava.git***
- 2. cd dalisava***

**Common Steps (After Either Method)**

***composer install***

***cp .env.example .env***

***php artisan key:generate***

Edit `.env` to match your setup:

**`DB_CONNECTION=sqlite`**

**`DB_DATABASE=absolute/path/to/dalisava/database/database.sqlite`**

### Run Migrations and Seeders

**`php artisan migrate --seed`**

### Install Frontend Dependencies

**`npm install`**

**`npm run dev`**



### Run the Application

**`php artisan serve`**

**Visit: `http://127.0.0.1:8000`**

**Dashboard: `http://127.0.0.1:8000/dashboard`**

**Login: `test@example.com`**

**Password: `password`**



### Core Concepts

#### Agent Structure

Each agent has:




- **``name``**: Display label (e.g., Main Checkpoint)
- **``type``**: ``checkpoint``, ``intersection``, ``traffic_light``, or ``highway_point``
- **``latitude``**, **``longitude``**: Map location
- **``state``**: ``G`` (green), ``R`` (red), ``Y`` (yellow)
- **``code``**: Unique key like ``Agent1``, ``Agent50``

#### Seeder

- **`php artisan db:seed`**

## Notifications

Based on the agent state, the following messages are resolved:

State	Message
Green 	Normal
Red 	Alert
Yellow 	Caution

## API Endpoints

### Get All Agents

*GET /api/agents*

### Simulate Driver Location

*POST /driver/location*

```
{  
  "next_agent_code": "Agent15"  
}
```

## Simulate Real Traffic Conditions

*php artisan app:simulate-traffic*

## Resources

- **Laravel:** <https://laravel.com/docs>
- **Vue:** <https://vuejs.org/>
- **Composer:** <https://getcomposer.org/>
- **Node.js:** <https://nodejs.org/>
- **SQLite:** <https://www.sqlite.org/>
- **NPM:** <https://www.npmjs.com/>