Weather Service

Note: All code challenges should be hosted on a publicly accessible git repository (i.e. GitHub, BitBucket, etc).

The business wants a web service developed that returns current weather information for a mobile client user based on their current zip code.

Constraints

- **Framework:** Laravel 5.5 or Lumen 5.5
- **HTTP Library:** Guzzle
- **Dependency Manager:** Composer
- **Hydration:** Symfony Serializer or JMS Serializer
- **Test Framework:** PHPUnit
- **Coding Style Guide:** PSR-2

Design Goals

- Keep controllers skinny.
- Implement an adapter/wrapper for the client class responsible for getting the weather data.
- Implement the cache as a decorator for the weather client.
- Bind services to an interface (not an implementation) in the service container.

Functional Requirements

- Consume weather data from https://openweathermap.org/.
- Provide an HTTP **`GET /wind/{zipCode}`** method that takes a zipcode as a required path parameter and returns a wind resource.
- Validates input data.
- Response format should be JSON.
- Cache the resource for 15 minutes to avoid expensive calls to the OpenWeatherMap API.
- Provide a CLI command that will bust the cache if needed.
- Response fields should include:
  - Wind Speed
  - Wind Direction

Unit Testing Requirements

- Use mock responses from the OpenWeatherMap API.
- Use mocks when interacting with the cache layer.

How To Run

1. Clone the repository.
2. Install dependencies:

```
$ composer install
```

3. Run the built-in web server:

```
$ php artisan serve
```

4. The wind resource should now be accessible by running a curl command:

```
$ curl -x http://localhost:8000/api/v1/wind/89101
```