ECO Surv PHP Laravel Exercise

These exercises are designed to see how well you understand Laravel. It is also an opportunity to see how far you can get before you feel you cannot continue further. Please do not worry about perfection; we are looking for persistence and your ability to complete the requirements.

Please read through the test fully before you start. There are some useful hints along the way which might save you some time.

Please try to finish one part before you start on another. Please ensure that you test your work.

The exercise consists of three sections:

1. Creation of an API which requires you to load data from an external service.
2. Creation of a data model which asks you to use more complex functionality of Laravel to store data and relationships provided in the first part.
3. GraphQL section which looks at your experience with working with different API types and how well you wrote your code in the first two parts; we will commend you for having a go!

For this exercise you will need:

- PHP 8.0
- Laravel 8
- mySQL/MariaDB/REDIS

Part 1 – External Data API – (45mins-1hr)

This part of the exercise requires you to construct the models, migrations, controller, and handler to return data provided by the dogs-ceo API. We ask that you use PHP to return the data from the API. Please note we will not accept a web-based FE approach using ajax to get the data.

Task Requirements

To complete this section, we want you to create an API in Laravel which provides access to the data provided by the dog ceo API at https://dog.ceo . We want you to undertake this activity so you can demonstrate an understanding of how to retrieve data from an external API provider.

You do not need to create an account nor authenticate to consume the API but please be aware that this API is rate limited.

The API must be able to:

1. Return all breeds.
    - Please use the url: **/breed**
2. Return a specific breed
    - Please use the url: **/breed/<breed id>**
3. Return a random breed.
    - Please use the url: **/breed/random**
4. Return an image by breed
    - Please use the url: **/breed/<breed id>/image**

Part 2 – Laravel Models, Relationships and Saving Data – (45mins – 1hr)

This part of the exercise is designed to see if you understand the varied data relationships Laravel supports, how to save data into them and how to work with caching using REDIS. This will also test your understanding of RESTful API's and the HTTP methods.

Task Requirements

1. Save the data provided by the dog.ceo API in a database model when you retrieve it from the external API (30mins)
    - Save all breeds into a database table – you may want to call this model **breed**
    - Save all breeds into a REDIS cache
    - Handle data updates/changes from the external API
    - Handle data deletes from the external API

2. Create a User model
    - The user model must have an **email(string 255)**, **name(string 32)** and **location(string 32)** property

3. Create a Park model
    - The park model must have a **name(string 32)** property

4. Create three polymorph tables called '**userable**', '**parkable**' and **'breedable'**

5. Create an API handler which will link a **park** to a **user** using the polymorph relationship
    - Please use the url: **/user/<user id>/associate** and a POST method which will contain the details for the type being added

6. Create an API handler which will link a **breed** to a **user**
    - Please use the url: **/user/<user_id>/associate** and a POST method which will contain the details for the type being added.

7. Create an API handler which will link a **breed** to a **park** using the polymorph relationship
    - Please use the url: **/park/<park id>/breed** and a POST method which will contain the details for the type being added
    - We may want to link **breed** models to other model types in future so please use the **breedable** table to achieve this. You might find some hints at the end of the test to save you some time in achieving the polymorphic relationships.

8. Create update your API handler which will return a given breed and add the data for all the users and parks which are attached to that breed

Notes

- Many users can attend many parks
- A user may have many breeds of dog
- Some parks only allow some breeds

Part 3 – GraphQL – (30mins)

This part of the exercise will demonstrate how well you have written your base models and if you have put the functionality of the exercise into your controllers or models. It will also highlight how you reuse code and functionality. Please return the data from REDIS if it exists, otherwise you will need to return the data from the API.

Please implement the graphql queries to return all the base models you have created so far. The following queries need to be solved (the property names may be different depending on how you named them).

```
query users {
      name, email, location
}

query breeds {
      image,
}

query parks {
      name
}
```

Feedback and some useful information

Please provide a README.md with documentation. If you struggle, want to justify something, or just have feedback about the exercise, please feel free to say whatever you like, and we will take this into consideration. We want you to succeed and we appreciate your time spent on this exercise.

ECO Surv is a great place to work, and the challenges we have are data related, so whilst this test may seem simple on the face of it there are some interesting challenges to solve.

We do not expect you to do any kind of validation or be side-tracked into ensuring your code is perfect from a production standpoint. Where you encounter a potential problem, please just add comments to your code.

e.g.

```
// this might not work
```

or

```
// I am not handling this properly
```

The estimated time is the time it should take you if you are fluent with Laravel and have a vast amount of experience. If it takes you longer, we encourage you to keep going!

A big hint: there is a part in the exercise which once completed, you should be able to copy and paste and change a little, to other parts of the exercise. This should save you a significant amount of time.

Best of luck! – we are looking forward to seeing how far you get! Have fun with it and please take your time.