Developer Commentary

# My Ramblings / Thought Process

First of all I read the requirements and definitely didn’t attempt to just build an API using RAW PHP. That’d be crazy right especially since it clearly states “in a MVC Framework”.

Anyway, That out of the way. I looked at the Zend Framework initially but that was a little confusing to setup and the documentation was bit spotty.

I attempted to find something I could chew into for an hour before deciding eh let’s just do it in Laravel I don’t really have the time to mess around with trying to get Zend working also apparently Zend is also laminas now and…bit weird not much time.

“How to set up a rest api in Laravel” – download code – Composer Install – npm install – npm run prod.

Copy .env.example .env

Smashing I have committed plagiarism now to make this do what I want it to do arguably this would have been simpler to just do from scratch but sod it. I have got a guideline in the code I need to clean up later now.

Php artisan make:model Users. Oh bloody auth content. Sod it! Let’s butcher it inline with what’s in the database.

It was at this point I noticed that countries had an id name of country\_id. It was at this point I entered a state of confusion as to why it wasn’t just ‘Id’ as that simply just makes more sense.

Countries should have an id. In the same way user should have an id. Country\_id would be a foreign key surely? That’s the standard I was always used to. Maybe you guys know something I don’t.

Onwards and Upwards!

(I took a massive break between now and writing the end of this)

User and Users one was dynamically generated the other I made myself to connect to the stuff in the database. Realistically this would all be accounted for ahead of time.

I started with creating an applications table. I thought about what would be required, I decided on: charity\_id, user\_id, stage, created\_at and updated\_at.

I created a table for this data. Smashing.

I then made an endpoint that would query for these records. Perfect!

I then progressed through the tasks with relative ease just assembling logic in a concise but quick manner as I was quickly running of out of time!

I done a little bit of testing once all of the API implementations were in place to ensure that all the happy day scenarios were functioning.

# Improvements

So The long and short is that I did not have as much time on this as I would have liked. There are probably some less optimal responses to some invalid inputs, but I didn’t see any in the small amount of testing I done.

There is also some code that could be optimised in some places.

I would have liked to default the ids in the databases to be labelled Id in line with the coding practices I’m used to.

I am also sure that there is a way to get Laravel working with these kinds of Id’s but again due to time constraints I didn’t really have the time to read into it too much.

# Ajax Implementation

Obviously, there isn’t an AJAX implementation…

But it is a relatively simple thing to do I believe.

Throw a cheeky:

Data-ajax attribute on a table.

Some JavaScript to listen for when the document (website) has finished loading. Some eager loading logic to retrieve the applications and push them into the table. Sorted. Obviously, this is not a replacement for an implementation, but I wanted to address how I would implement it anyway maybe that gives me a bit of a gold star.

# Api Stuff

## Posts

In posts I’ll include the data that I have posted that gave me a response.

### /api/applications/move

{

    "application\_id": "1",

    "user\_id": "5",

    "direction": 0

}

### /api/applications

{

    "charity\_id": "stick2",

    "user\_id": "stick2"

}

### /api/applications/create

{

    "charity\_id": 5,

    "user\_id": 5

}

## Gets

In gets I will show you the response I received.

### /api/applications

[

    {

        "application\_id": 1,

        "charity\_id": 5,

        "user\_id": 1,

        "stage": "Organisation Approval",

        "created\_at": "2021-05-01T00:00:00.000000Z",

        "updated\_at": "2021-05-18T00:00:00.000000Z"

    },

…

### /api/applications/checkstatus/1/1

{

    "message": "application can be moved forwards"

}

### /api/charity/approved/5

1

(This should probably be tweaked in order to give a bit of a better response but it serves its purpose.)

### /api/charity/location/5

UKR

(This should probably be tweaked in order to give a bit more of a formatted response UKR probably doesn’t mean much to many people… Ideally this would be an array of countryCode and Name but I’m running short on time but If I can get the countryCode from a database I’m sure I could also query for the countryName and blast them both into an array to return)