Laravel – Hello World

I've recently been learning Laravel. Here are some of the steps necessary to create a very simple hello world application

- This is a whole Laravel application
- It has a single controller
- It has a single route
- This will store the text "Hello World" as a variable $text, in the controller
- It will that on the screen via an HTML template

This application doesn't use databases, but uses several of the necessary components of an MVC application

Step 1: Creating an Application

The first step is to use composer to generate the overall application with:

```
composer create-project laravel/laravel hello_world
```

The next step is to change into the directory with:

```
cd hello_world
```

and then open the code base with your favourite editor/ide. I'm using VSCode:

```
code .
```

Step 2: Generating a Controller

Back at your command line, you want to generate a controller with:

```
php artisan make:controller HelloController
```

This will create a file named HelloController in app/http/Controllers:

```
ls -l app/Http/Controllers

total 16

-rw-r--r--  1 jmccourt staff  361 May 18 11:37 Controller.php
```

```
-rw-r--r--  1 jmccourt  staff  122 May 27 09:12 HelloController.php
```

Inside the controller you want to add the following code (some of which will already be there from the generator):

```php
public function index()

{

    return view('hello_index');

}
```

inside the "class HelloController extends Controller" block of code (it goes inside the {} braces).

Step 3: Adding a Route

Moving back to the code base, you want to edit the file routes/web.php and add this line:

```php
Route::get('/hello', 'App\Http\Controllers\HelloController@index');
```

This adds a single route, /hello, which is then handled by the Hello controller and the hello_index action.

You can test that this route is available by running:

```
php artisan route:list
```

which should show you something like this:

```
+--------+----------+----------+------+---------------------------------------------+------------+
| Domain | Method   | URI     | Name | Action                              | Middleware |
+--------+----------+----------+------+---------------------------------------------+------------+
|     | GET|HEAD | /      |    | Closure                        | web     |
|     | GET|HEAD | api/user |    | Closure                        | api     |
|     |     |     |    |                       | auth:api  |
|     | GET|HEAD | hello   |    | App\Http\Controllers\HelloController@index | web     |
+--------+----------+----------+------+---------------------------------------------+------------+
```

Note: It is critical that you use the fully name spaced path to the controller (i.e. starting with **App**). According to what I've read, you can use a use declaration at the top of the

routes file to import controllers so you don't have to have the name spacing everywhere but I was unable to ever make that work.

Step 4: Adding a Variable to the Controller

Inside the public function index method of the Hello controller, you want to add / change the following code:

```php
$text = "Hello World";

return view('hello_index', ['text' => $text]);
```

This defines one variable, $text, which should then be available in the view. The way that it becomes available to the view is by passing a hash into the return statement which has a key of 'text' which is mapped to the variable $text.

Step 5: Creating a View and Displaying a Variable

Views, in Laravel, live in the directory resources/views so you can create one with the command:

```
touch resources/views/hello_index.blade.php
```

And you then want to edit that file and add this code:

```
<h1></h1>
```

Step 6: Previewing the Results

The key command here is:

```
php artisan serve
```

which starts the application and displays the result on:

```
http://localhost:8000/hello
```

At which point you should see "Hello World" come onto the screen.

Conclusion

Following the instructions above will help you create a fully functional hello world app in Laravel, with a route, a controller and a view.

For more guides like this please see https://johnmccourt.con