

## Lab 6: Your last lab!

### COMP 3015: Web Application Development

#### Your mission:

Build a small RESTful API from scratch. No starter kit provided. It should conform to level 2 on the [Richardson Maturity Model](#), which is what is typically referred to as RESTful in industry.

- Generate and configure your Laravel application
  - Run: **composer create-project laravel/laravel <proj name>**
  - Set the needed database values in your **.env** file (this file is automatically generated when you create the project). Specifically, you will need to set the **DB\_CONNECTION=mysql**, the database name, credentials, host and port information.
- Run **php artisan install:api** which configures your Laravel application to be nicely set up for this out-of-the-box. One of the things this creates is the **routes/api.php** file. These routes are all prefixed with “api/”.
- Run **php artisan make:model Article --all**
- Find the generated article table migration at: in **database/migrations**
  - Note that the file name will have a timestamp in it
  - Add title and url columns, keep the timestamps and id (which becomes the PK)
    - See: [creating columns](#)
- Run the migrations using: **php artisan migrate**
  - This will create a few tables including an articles table which we’ll be working with.
- Add routes under the **routes/api.php** file. When making requests to your server remember that these will be prefixed with “api/” by Laravel.

- Within the **ArticleController**, implement the **index**, **destroy**, **update** and **store** functions. You can delete the other functions that were created for you – they're not needed as we aren't rendering any view files.
  - **index** should return all articles in the database as JSON.
  - **store** should save an article in the database with a provided title and url.
  - **update** should respond to a HTTP PUT request and update a given article, based on a provided ID (the ID should be a route parameter eg. HTTP PUT `http://localhost:8000/api/articles/2`).
  - **delete** should respond to an HTTP DELETE request and remove the given article from the database. Similar to **update**, the ID should be a route parameter.
  - All of the **index**, **store**, **update** and **delete** functions should return JSON responses to the client. Laravel has the `Illuminate\Http\JsonResponse` class for this.
  - Don't worry about data validation at this stage. The purpose of this lab is to just get an introduction to the basics of web APIs.
  - For getting data out of the Request object, see: [Accessing the request](#)
  - See Eloquent docs for saving records to the database: [Eloquent: Getting Started](#)
  - The generated create and edit functions can be deleted as those are for server side rendered applications (such as our assignment 1, 2), but now we're developing a web API that simply returns JSON to the client.
- Ensure you run the server using Laravel: **php artisan serve**
- Test your web API using [Postman](#) or [Insomnia](#), or a similar HTTP client

**Submission:**

- Commit and push your changes to GitHub as usual.
- Do not submit the **/vendor** folder or your **.env** file.
- Due week 12

**Evaluation:**

- Functionality / 10
  - 4 controller function implementations, 2 marks each
  - 1 mark for adding the required routes to the **routes/api.php** file
  - 1 mark for the articles table migration