

## SUMMARY OF YOUTUBE COURSE

### Laravel API Crash Course for Beginners By Code with Dary

#### RATINGS: /5

- REST: Representational State Transfer, is an architectural style used for communication between a server and client, it used HTTP,
- REST data will be send in either XML or JSON
- Disadvantages of REST: It is not stateful, You cannot push data through REST, You always need to send some kind of context to the server
- HTTP Verbs: GET, POST, PUT/PATCH, DELETE
- Stick with Plurals or Singular, just keep it consistent
- Status Codes:
  - 2xx – Request Successful
  - 3xx – About redirection
  - 4xx – Have been an error
  - 5xx – Deals with the server
- He suggests sticking to “snake cases” when writing out JSON APIs
- Use “Collection” in PostMan to keep API endpoints organized
- “laravel new project\_name”: to create a Laravel Project
- “php artisan --version”: to get current Laravel version
- “.env”: to setup Database connection parameters
- “php artisan tinker”: very nice for populating the database
- To make api request with Laravel, you always need to add “/api” before the api route
- You might need to set the header of you request to “Accept: application/json”
- Different ways to authenticate a user using middleware like “passport”
- For this course he used “passport credentials” to authenticate users
- “php artisan make:model -a -r dbTableName”: creates a model, (“-a”) database migration, controller, (“-r”) resource
- It’s not best practice for our controller to be singular (\*like who cares\*)
- “app > Http > Controllers > ”: path to controllers
- “database > migrations/seeders/factories > ”: path to access migrations
- “app > Models > ”: path to access DB Models created
- Before you run “php artisan migrate” : make sure the configs in “migration” and “Model” for that particular table is correct, like table fields, which are fillable
- “Factory”: used to generate faker data for our DB Table, to call our factory we need a DB Seeder setup, which calls the Model class
- “php artisan migrate --seed”: to seed the DB
- Route setup: \*Route::get('/route', [Controller::class, method\_for\_that\_route\_in\_the\_controller])\*

- “php artisan make:resource resource\_name”: used to make a Resource, We create a resource and then use it in our Controller
- In REST APIs IDs must be of type “string” (so he says)
- “php artisan route:list”: List of available routes
- “php artisan make:request request\_name”; Helps to customize our own request, specify which fields are uniques, required, it can be done from the DB but the error messages here are more helpful
- “php artisan make:migration MigrationFileName”: Create your own migration file
- Using a pivot table for Many to Many relationship