LARAVEL DEVELOPMENT STEPS

# Create Laravel Project

Depends on the environment.

# Create models with everything

Models, migrations, controllers and factories can all be created in single line below:



* Better if I put all models in a single folder called models

# Design database in migrations

* If database is complicated, try mind mapping it on paper first.

## ALWAYS USE “unsigendBigInteger” IN RELATIONSHIPS

I should always use unsigned big integers (UnsignedBigInteger) on relationships, because the auto increment id in all tables are unsigned and big integer. Therefore, if I use foreign() to for example cascade on delete or something, the foreign is not going to match and error happens.

## SOFT DELETE

* I once read an article that said we must not use soft delete on user model. However, it is very good to use soft delete in as many as models we can.

To use soft deletes, we add it to the model:



And we add it to the migration:



## CASCADE ON DELETE | CASCADE ON RESTORE

We must use cascade on delete (or restore if it is an option). To do it more eloquently, based on my experience we better use a package:





The “replies” is the relationship. For example, post has many “replies”.

This package is making this very EASY!

Now we use delete(), withTrashed(), restore() like before and the rest is done for us automatically.

# Code the relationships in Models

* Remember: Let’s say Post belongs to User. We know that Post is the child and User is the parent. then in Post model we say belongs to.
* And: Let’s say Post has many Replies. This time we know that Post is parent and replies are children, then in Reply model we say belongs to, and in Post model we say has many.

## USE MODEL CLASSES INSTEAD OF NAMES:



# Design Factories

I can always use bitfumes factory tutorial that is inside “Laravel spa forum pusher app” tutorial, until I can do it on my own. Its pretty complete and easy though.

# Code the Routes

* All routes inside “routes/api.php” will get “/api” prefix automatically

## USE “apiResource”

In order to create the routes for api, I use apiResource option:

For example:

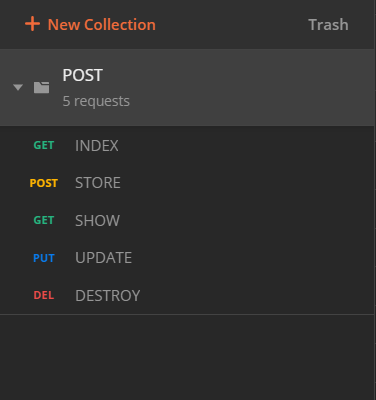


# Code the Controllers & Test with Postman

* Double shift and type “response.php” to get every statues text available!

## OPEN POST MAN TO TEST RESULTS

For every model I can create a new collection in postman, for example, Post. Inside post collection I can have the required api calls.



## START CODING IN CONTROLLER

In order to ignore mass assignment, I use this in my models:



### INDEX

My current template for index is like this:



### STORE



### SHOW



### UPDATE



### DESTROY



## DESIGN RESOURCE TEMPLATE

* Designing template resource for every model is optional for me

### Create a resource

To create a resource, we run this command:



### Use resource in controller

* To use a resource in controller the syntax is based on if it’s on index or show.
* The example is already shown in top controller section.

### Design resource template

Its optional, for example:



* Also, we can use an additional function called with to add more information.

## DESIGN REQUEST VALIDATION RULES