

Laravel Exercises Cheat-Sheet



To create a new migration run

php artisan make:migration create_tableName_table

If you use the correct syntax for creating a new migration like mentioned above the artisan will make a blueprint for you with the schema structure the Id and timestamps, Also it will set up the drop table in the down function if you want to revert your change.

To migrate your new migration:

php artisan migrate

This will migrate all new migrations you have added to the database.

To revert your last (set) migration:

php artisan migrate:rollback

This will revert all the last migration(s) you have migrated

To create a new model

php artisan make:model modelName

This will setup a empty new model instance

To create a new controller

php artisan make:controller nameController

This will create an empty controller instance

To write a query with Eloquent

A few examples:

ModelName::all(); // Will return all rows present in database

ModelName::all()->with('relation'); // Will rows with relation attached (this is defined in the model)

ModelName::where('column','=', 'value'); // Will return all rows with the value in defined column

ModelName::all()->orderByDesc('id'); // Will return all models in descending order (newest first)

Find more at <https://laravel.com/docs/9.x/eloquent>

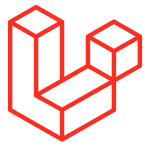
To save a model

```
DB::transaction(function() use ($request){
    $car = new Car(); //Creates a new model
    $car->brand = $request->get('brand'); //Set value to model
    $car->type = $request->get('type');
    $car->color = $request->get('color');
    $car->save(); //Saves the model to data
});
```

Using a transaction is optional. But it's good practice to use one. In the controller you can call for the creation of a new model, where you then can set the properties to it. to save this model to the database you call the save() function

Tip: To update a model replace the new Car() with an eloquent query resulting in the first() result of a car

Laravel Exercises Cheat-Sheet



To create a new route

First navigate to the routes/web.php

```
Route::get('/cars',[Carcontroller::class, 'display'])
    ->name('cars.display');
```

Here you can define your new route, it has to contain a location /cars here for example. And you connect the controller to that request, while specifying the controller you also specify which function in that controller. and you give the route a name for the redirect function in inertia.

Inertia Rendering

```
class CarController extends Controller
{
    //Retrieves all cars in database and renders the Dashboard with cars as props
    public function display(){
        $cars = Car::all();
        return Inertia::render('Car/Dashboard',[
            'cars' => $cars
        ]);
    }
}
```

If you want to render a vue file and supply it with properties you return an Inertia render in your controller. The file rendered is stored in resources/js/pages/car/dashboard.vue (in my case) and the supplied properties 'cars' are being queried with a Eloquent query.