

**Game Test API**

Installing:

1. Unzip the files from the **GameTest** archive
2. In your local environment just create a database with the name “**gametest**”
3. Run **composer install.** Here, I have 2 specific packages:
   1. nwidart/laravel-modules
   2. cviebrock/eloquent-sluggable
4. If everything It’s ok on the 2nd step just run **composer dump-autoload** command
5. Edit the file .env with your local database server credentials
6. On this step, run **php artisan migrate**
7. If everything It’s ok the project must be be up and running

The project is also on a Git repository: https://github.com/keops007/gameTest.git

Database:

The database has 3 tables:

* **game\_competition** – store the competitions informations
* **game\_player** – the players for the competitions
* **game\_competition\_player\_log** – a log with all recived points

Sorce code:

The project it is created on Laravel 8 with PHP 7.4 and with a MySql database.   
I’ve installed a package, nwidart/laravel-modules, that helps me to organize the project. With this package you cand have an independent “ecosystem” for every module of the project.

So, the core of this project is located in “**Module/Game**”. Here, we have three models in the “**Entities**” folder, one controller in the “**Http/Controllers/Api**”, one repository in “**Repositories**” and three form requests in “**Http/Requests**”.

Also, here, I have the route file (**api.php**) in the “**Routes**” folder. In this file we have all the routes for the API endpints.

For some specific validations I’ve created 3 rules in “**App\Rules**”.

API’s:

I have built 5 API endpoints:

1. /api/competitions/list – method GET  
   List all competitions from the API
2. /api/competitions/create – method POST  
   Create a new competition
   1. Fields: **name and players\_number**
   2. Headers: **Accept: application/json**
3. /api/competitions/create\_player – method POST  
   Create a new player in a competition
   1. Fields: **name and competition\_id (form the 1. API resource)**
   2. Headers: **Accept: application/json**
4. /api/competitions/player/add\_points – method POST  
   Add new points for a player
   1. Fields: **player\_id, competition\_id and points**
   2. Headers: **Accept: application/json**
5. /api/competitions/list/{**competition\_id** } – method GET  
   Get all players from a competition, ranked by points number

All API’s responses ar in JSON format.

Testing

In the archive You’ll find a configuration for PostMan where you can replace my base url API with yours.   
GameTest.postman\_collection.json

