

ADVANCED PHP TECHNICAL TASK

In this test we aim to assess your coding skills and see how you would design the solution for a new feature that is part of a larger project. This will also give you a preview of the type of tasks you will handle as a **PHP Developer** within our team, developing the game server **REST API** that communicates with the mobile game clients.

We challenge you to build the core of a competition system needed for the game, where a generic competition is defined by a unique name and a limited number of players that can sign in. For each player that enters a competition a simple scoring system is saved - only increments of 1 point will be allowed per API request.

Also, to keep things simple, no authentication or user session will be needed, we can consider players just entered a username and joined the selected competition.

So we will need the following features (endpoints) for the game API:

1. create a new competition

a) 2 parameters for competition name & player limit will be enough

2. add a new player in a competition

- a) *player name is enough and will be considered unique within the competition*
- b) *new players are not accepted once the competition player limit was reached*

3. increment a player's score (+ 1 point) in a competition

a) *player & competition should have unique identifiers*

4. return the competition name & players ranking

- a) *players should be ordered by total score - descending*
- b) *nice to have: paginated players ranking, in case the list would be too long for the game client*

Some technical details to keep in mind:

- use **Laravel** framework - there are plenty of resources online if you are not familiar with it and you can build your local project however you want
- all data should be stored in a MySQL (or similar relational) database
- the game client expects the API to return JSON response
- you can use any tools / libraries you need
- front-end is not needed, but API endpoints should be described somehow for easier testing (you can add comments with the parameters needed for each endpoint somewhere in the code or along with your notes)
- send us an archive with the project files - except "vendor" folder
- project installation steps should be provided (in the README file or along with your notes)

We are strong believers in ownership and team collaboration so we expect functional code that is easy to read and can be extended later on.

Don't worry, we will never use the code written for this test in any of our games, we already implemented these features in various iterations, but we are curious how you would implement them.

We are also interested in your feedback regarding this test, any notes you have for future improvements of your solution (we understand time is limited), so don't forget to pass them along with your solution - this will be the starting point for our technical discussion.

Good Luck!

greenhorse
G A M E S