2021_INFO2_PWCS_NodeJS_TP2

Learning Outcomes

At the end of this 4 sessions laboratory, you will be able to :

- Create a web app with NodeJS and Express
- Create and use a micro-service architecture
- Create a Restful API with NodeJS

The main goal

You will create a web application on which identified users (players) will be able to play various games. This will be called the *game server*. The scores of the players and its identity will be served by a RESTful webservice (and stored on an online database to be shared by various game servers). The available games will be served by a RESTfull webservice too. The server hosting those two micro-services will be called the *micro-service server.*

Games are simple ones : “guess the number”, “give a palindrome”, “how many time can you click in 5 seconds...”

Use cases

1. A user authenticate itself on the game server through a form. If it is his first visit, a new user form is provided.
2. An identified player pick up a game in a list. The rank of the player for a game is displayed on the line of the game.
3. At the end of the game, the player’s score is updated and the full ranking for this game is displayed.

Technical details

1. Players info (login, scores for all games) and Games (their names) should be served by RESTful webservices. Since those micro-services could be accessed by various *game servers*, the *micro-service server* ought to be on a distinct machine. Here, we will settle for the same computer but the database should be inline. You can use inline instances of MongoDb (Mogolab) for instance.
2. You will need to create sessions on the *game server.* To implement it, you may use cookies or express sessions. The security of the application is not at stake, here.

Bonus

To go further, you may implement other functionalities on top the micro services :

1. A super-user (admin) is able to ban a player through a form (on the game server)l.
2. A user authenticates himself with github Oauth.