parkside

Parkside Coding Challenge

Backend Developer

We create interactive experiences that people love to use. That is what all our developers take to heart and why we ensure our code is of the highest possible quality. We believe you can be a part of this and would therefore like to see your skills and your talent in action!

The process

We'd like to ask you to implement our coding challenge with a programming language mentioned in the job offer and backend framework of your choice. We don't impose any restrictions on how you implement the example - but of course, format, structure and quality do play an important role. Since the application is not too complex, certain features like containerization, authorization or hosting are considered optional.

If due to time constraints you need to leave certain aspects out, please add a section in the Readme explaining what you would do better given you have more time. If you are applying for a full-stack position please add a section to the Readme and explain how you would set up the frontend application.

Once you are done, host the code in a **private** Git repository of your choice (for example GitHub) and give us access - please invite codingchallenge@parkside-interactive.com.

Afterwards, let HR know through the already established communication channel. We will review your code and provide you with feedback. In the next step, we would ask you to walk us through the example and explain your code and decisions.

parkside

The challenge

Imagine a world where all robots have only one goal - to win the **Robo-Dance** competition! As you know, all robots love dancing and regularly battle each other in fabulous dancing competitions. We would like to ask you to create a simple REST API to provide the robots with the stage they were always dreaming about.

Design the API in a way that a frontend application could consume it.

Features

- Provide an endpoint to retrieve all **/robots**
  - The robots should come from a persisted data source
  - Provide an option to retrieve an individual robot as well
- Robots need to be able to share their **/danceoffs** results
  - Robots battle each other in teams of 5. Each member of every team battles one member of the opposing team in a dance-off - this should lead to 5 battles in total. A battle between two robots has one winner.
  - Think of and implement reasonable validation
  - Optionally, this endpoint can be protected
- Allow retrieving of all previous **/danceoffs** to see who is the real dance champion
  - Ideally, provide an order so a leaderboard can be shown

parkside

Schema

Robot

```
{
    "id": 1,
    "name": "Funky Joe",
    "powermove": "Spinning Turtle",
    "experience": 5,
    "outOfOrder": false,
    "avatar": "https://robohash.org/funky-joe.png"
}
```

Disclaimer

Robots lovingly delivered by Robohash.org