

Parkside Coding Challenge

Frontend 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 one of the following frontend frameworks: Angular, React or VueJS. Other than that, we don't impose any restrictions on how you implement the example - but of course, code format, structure and quality do play an important role. Since the application is not too complex, certain features like state management, tests 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 API.

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 will ask you to walk us through the example and explain your code and decisions.

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 you to create a simple frontend web application to provide the robots with the stage they were always dreaming about.

Features

- Allow the user to start a new competition
- Allow the user to create two teams and give them names
- Team members are filled in automatically using the following criteria:
 - Each team is assigned 5 robots
 - Robots are fetched from the API provided below
 - Robots which are out of order can't dance
 - The total experience of each team can't exceed 50 points
 - The total experience is the sum of the experience property of each team member
- Allow the user to start the dancing competition using the following criteria:
 - Each robot dances against another robot of the opponent team in a dance-off
 - For each dance-off, the system picks a winner randomly
 - In total, 5 dance-offs take place
 - The dance-off results should be sent to the backend using the API below
- Allow the user to see a leaderboard
 - There is an API endpoint to retrieve all dance-offs
 - The leaderboard should **ONLY** be accessible if at least one competition has been finished. It should show a list of all robots ranked by most dance-off wins.

API

Please use our API to implement the example:

<https://challenge.parkside-interactive.com/api>

<https://challenge.parkside-interactive.com/docs>

Schema

GET /robots

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

GET /robots/:id

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

POST /danceoffs

```
[
  {
    "opponents": [1,3],
    "winner": 3,
  },
  {
    "opponents": [5,23],
    "winner": 5,
  },
  ...
]
```

GET /danceoffs or /danceoffs/populated

```
[
  {
    "id": 1,
    "winner": 2,
    "loser": 4,
    "dancedAt": "2019-11-29T12:06:28.000Z",
  },
  {
    "id": 2,
    "winner": 3,
    "loser": 5,
    "dancedAt": "2019-11-29T12:06:28.000Z"
  },
  ...
]
```

Disclaimer

Robots lovingly delivered by Robohash.org

