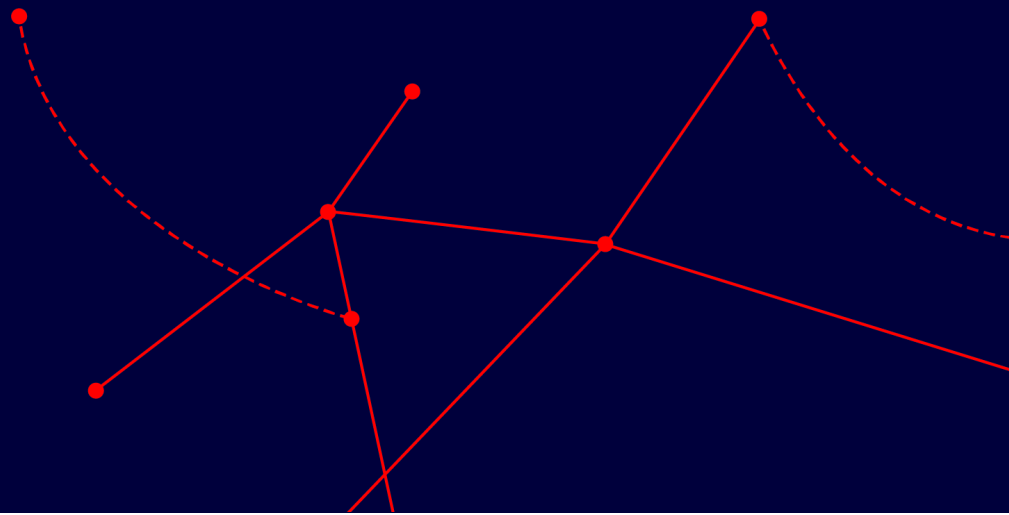


# Frontend Engineer Assignment



## 1. Resources

- You should've received the following files together with this document
  - `/assets/` - a folder containing 3 background images
  - `/mock/carousel.png`
  - `/mock/matchstates.png`

## 2. ASSIGNMENT 1 – HTML/CSS

- Implement a single static web page containing one “match card” based on design template `mock/matchstates.png`. Choose one state, for example, pre-match.
- Web page must be fully responsive. Note the 3 breakpoints defined in design template at 870px, 450px and 320px.
- Use OpenSans font available at <http://www.google.com/fonts/specimen/Open+Sans>

### 2.1. Assignment Format Requirements

- Provide a folder `/html-test/` containing:
  - One HTML file: `index.html`
  - One CSS file: `style.css`
  - Any additional assets in folder `/assets/`

## 3. ASSIGNMENT 2 – Match Carousel React Component

- Implement a `React.Component` class according to the following parameters:
  - Classname: `MatchCarousel`
  - Configuration parameters and behaviour:
    - `sportId` – Integer – if set display only matches for particular sport. Defaults to none.
    - `max` – Integer – maximum no. of matches displayed in carousel. Defaults to 10.
  - Component usage examples:
    - `<MatchCarousel max={15} />`
      - Will display 15 matches from any of the sports available. Keeping the order of sports and matches exactly as provided in the API.
    - `<MatchCarousel sportId={2} />`
      - Will display 10 (default) basketball matches.
- Each carousel item (aka “match card”) must follow the design template used in Assignment 1. Note the 3 distinct match states.
- Note that some of the attributes described in API documentation might not be available for all of the sports.
- UIX behavior:
  - Carousel must cycle through available items automatically with 3 seconds delay per item.
  - Carousel must provide a standard navigation element rendered as dots below the match card items. Clicking a dot must navigate to particular card. Clicking a dot must also reset the auto-play timer but must not stop the auto-play.

## 3.1. API Documentation

- [http://ls.betradar.com/ls/feeds/?/betradar/en/Europe:Berlin/gismo/event\\_fullfeed/0/5](http://ls.betradar.com/ls/feeds/?/betradar/en/Europe:Berlin/gismo/event_fullfeed/0/5)
  - `match.status._id`
    - 0 for not started
    - 100 for ended
    - consider anything else live
  - `match.status.name` – match status display name
  - `teams.home/away.name` – team display name
  - `teams.home/away.abbr` – team short name
  - `teams.home/away.uid` – teamid. Obtain the team crest/flag by using id with the following URL :
    - [http://ls.betradar.com/ls/crest/big/<team\\_id>.png](http://ls.betradar.com/ls/crest/big/<team_id>.png)
  - Tournament label at the top of the card:
    - `tournament.name + '-' + tournament.seasontypename`
    - `realcategory.name`
  - Time:
    - `match._dt.time`
    - `match._dt.date`

## 3.2. Assignment Format Requirements

- Use the <https://github.com/facebook/create-react-app> to create a new app named 'match-carousel'. Afterwards, create:
  - `src/components/Card.js` – The card component, which displays the data of single match
  - `src/components/Card.css` – Style for Card component.
  - `src/components/MatchCarousel.js` – Carousel component which fetches the data and displays the carousel made of Card components.
  - `src/components/MatchCarousel.css` – Style for MatchCarousel component.
  - `src/components/App.js` – Application root with two tabs.
    - One tab displays single MatchCarousel with max number of matches being 10.
    - Other tab displays two MatchCarousel components. One for sport 1 and one for sport 2.
  - `src/assets` – Folder with 3 background images provided
  - `notes.txt` – List all files/folders in the `src/` folder and dependencies in `package.json` file with short description. Similar to this list. If necessary, provide additional section "Known Issues" and describe corner cases, potential issues or cases that your implementation doesn't cover.
- Zip the contents of the `match-carousel` folder, without `node_modules`.
- How will we review:
  - System requirements: Nodejs 8+, npm 5+
  - Unzip the file
  - Go into the folder
  - Run `npm install`
  - Run `npm start`
  - Open <http://localhost:3000> in a browser

#### 4. ASSIGNMENT 3 – Optimization

Consider a scenario where more than one plugin instance is rendered on a single web site and describe how would you optimize or implement the client side data layer. Main points are reducing number of XHR requests and optimizing match carousel initialization speed.

##### 4.1. Assignment Format Requirements

- Provide a single plain text file `/notes-optimization.txt`

