

Welcome to your culture day

In order to evaluate your skills, and for you to experience our working environment, we have prepared a programming task which we would like you to perform. Your goal will be to write a small web page with a JavaScript part that reads a JSON feed, accumulates the data, and outputs the calculated results.






You can use any JavaScript-based technology you want. For a quick React setup, you could e.g. use create-react-app to get you started (<https://create-react-app.dev/>).

Please don't hesitate to ask Ralph or Sebastian if you have questions.

Goal

Your goal is to create a web page with a component that calculates and displays scores of sports matches in a sports league in real time. Multiple sports matches are always running on the same match day. Display the team pairings for the current day, receive match events, and use them to display and update live scores. When new matches start, the scores should reset and start again with the new team pairings.

Use the following mock-up as a reference for how the live score display should look.

MEX		0:0		CRI
JAM		0:0		USA
ESP		0:1		ITA
ENG		0:1		TUR

If you have the time, try to style the output closely to the mock-up, but as long as the output is clearly understandable, your priority should be the correct handling of the data. Assets for the team flags are available in the trial workday folder. Optionally, you can implement animations for the score updates, or add an option to display full names instead of abbreviations, or graphically indicate when matches are running.

Additional Requirements

If you are doing this coding exercise from home with more time there are additional technical requirements:

- Use Typescript
- Implement unit tests

JSON feed

In order to complete this task, you will need to poll a JSON data feed, which you can access at <http://vgcommonstaging.aitcloud.de/livescore/>

This feed represents a continuous stream of events that happen in multiple sport matches running at the same time. The feed takes no parameters.

You need to call this feed repeatedly every 2 seconds because the content will change continuously.

JSON Feed structure

This is an example of the feed output:

```
{
  phase: 'match',
  matches: [
    {
      match_id: 987654,
      tournament_id: 123,
      round: 4,
      home_team_id: 1,
      away_team_id: 2
    },
    ...
  ],
  teams: [
    {
      team_id: 1,
      team_name: 'Mexico',
      team_name_short: 'MEX'
    },
    ...
  ],
  events: [
    {
      event_id: 123456,
      event_type: 'goal',
      event_time: 5,
      match_id: 987654,
      score_amount: 1,
    }
  ]
}
```

```
        score_team: 'home'
      },
      ...
    ]
  }
}
```

The feed returns a JSON object with multiple properties. These are explained in the following paragraphs:

Property 'phase'

The `phase` property contains a string indicating whether matches are running. Possible values include 'pre_match', 'match', 'post_match' and some others, which you can ignore.

Property 'matches'

The `matches` property contains an array of match objects. For the purpose of this exercise, the feed contains all matches that are currently running – the number of matches in a round is constant.

A match object contains the id of the match and of the participating teams ('home' team and 'away' team).

Property 'teams'

The `teams` property contains an array of team objects. For the purpose of this exercise, the feed always contains all teams.

A team object contains the id of the team and its name.

Property 'events'

The `events` property contains an array of event objects. There are multiple types of events, such as 'match_start', 'match_halftime', 'match_end', 'goal', etc.

The 'score_team' and 'score_amount' properties are only available for the 'goal' event. The 'event_time' is a number in seconds relative to the start of a match.

Events are only filled during the match. If the matches are not currently running (i.e. not in the 'match' phase), the events array is empty.

Good luck!

P.S.: You probably won't have enough time to fully complete this exercise. 100% completion is not a requirement – we just want to test your skills and how you approach problems and challenges.