

# ContentWise Pokémon Challenge

We ask you to build a single-page application that consists of a wizard that will guide a user in creating a Pokémon team.

For the purpose of this challenge, you will use the Pokémon API (<https://pokeapi.co>). You can choose either or both of the following API:

- <https://pokeapi.co/docs/v2>
- <https://pokeapi.co/docs/graphql>

## Functional Requirements

The wizard should be structured into 3 steps:

1. **Pokémon trainer details.** In this step, the user inputs the details of the trainer they want to be and their team, consisting of:
  - Player name
  - Team name
  - Favorite Pokémon type

The Pokémon type should be selected from a list of types. The user should be able to search them and select one using the type API e.g.

`https://pokeapi.co/api/v2/type/{id or name}`

2. **Pokémon team selection.** In this step, the user chooses the Pokémons to compose their team. The user should be able to see a list of available Pokémon from the Pokémon API e.g.

`https://pokeapi.co/api/v2/pokemon/{id or name}`

`https://pokeapi.co/api/v2/type/{id or name}`

The Pokémons list should present the Pokémons belonging to the favorite type (selected by the user in step 1) at the top, followed by all the other Pokémons. The list should show for each Pokémon:

- the name
- an image preview
- the type(s)

Each item in the list should expose this behavior:

- on click, a card/infobox/drawer containing at least two images and all Pokémon major information (at least name, types, species, list of moves, list of forms) should open
- a way to select the Pokémons to compose the team from the list

The selected Pokémon composing the team should be easily identifiable while the user selects them. The maximum number of Pokémon composing the team is 7.

3. **Opponent team selection.** How to design and implement this wizard's step is up to you. In this step, the user will be proposed by the opponent's team, automatically generated by the system.

The opponent's team size is 4 and should be composed only of Pokémon whose generation does not match the generation of any Pokémon in the user's team (chosen at step 2).

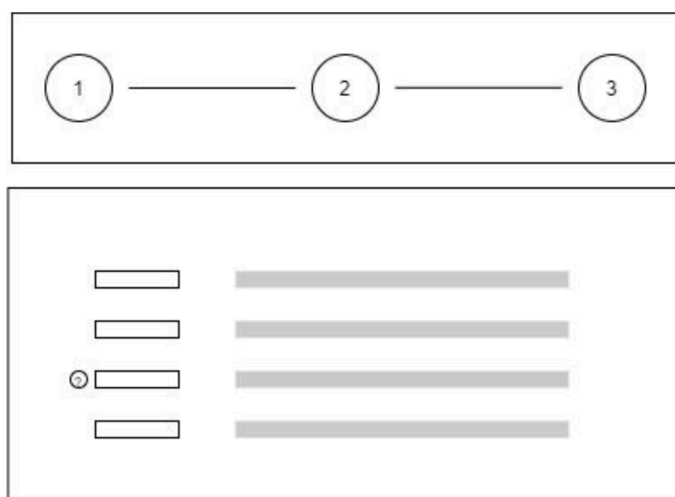
The user can confirm the proposed team or ask the system to generate a new one.

The opponent's team can be shown with any component of your choice (cards, columns, list, tables ...) and should contains at least the following information:

- name
- types
- species
- stats

The user should be able to navigate through the wizard by clicking on appropriate "back" and "next" buttons.

The application should show a small overview of the wizard as in the wireframe below:



The UI should notify the user if they haven't inserted required or invalid information before letting them proceed to the next step.

In the last step, the "Next" button should simulate a fake request (eg. waiting for a bunch of seconds) and give the user successful feedback.

## General Requirements

You should strive for a well designed, maintainable and testable codebase, don't spend too much time in styling and making things pretty, but rather invest in good architecture and logic.

It's not compulsory to implement all the details of this challenge (but it would be very appreciated if you did), but you could prioritize and implement what you think is more important for the overall functioning of this simple app.

Design appropriate testing strategies for the application (you can decide whatever framework and testing strategy you want) and implement some tests that seem more reasonable and meaningful according to you.

You can use the framework and libraries of your preference (we suggest you use React and TypeScript).

Provide a Dockerfile that allows deploying the application.

Write a readme file that explains how to set up a dev environment for the application, how to build it and to package it for deployment.

We are aware that some specifications and requirements may appear vague, and it is up to you to make reasonable assumptions and motivate your choices accordingly. There are many possible ways to implement this challenge, but we are more interested in hearing all your insights about how you would design this simple application.

## Contacts

If you find the text of this challenge not clear enough please don't hesitate to contact [riccardo.biondi@contentwise.com](mailto:riccardo.biondi@contentwise.com)