

# POLLEN8

## Technical Exercise

### Beer finder

The intent of this exercise is to build a SPA that allows users to browse information on their favourite BrewDog beers. You are free to pull in whatever dependencies you feel are relevant to achieve the specified features. Feel free to use Google as much as you want, but make sure to complete this exercise by yourself. The solution you present will be discussed in detail in a follow-up interview.

### Application requirements

The goal is to build a simple SPA that displays BrewDogs beer offering and allows users to search and refine this list to find the beer for them.

### **Landing Page**

This is the list view for the beers. It will need to:

- Present a list or grid of 25 beers
  - List/grid items should show image, name, tagline, first\_brewed and abv
- Support pagination, lazy loading or a load more button
- Search input allowing users to search by name
- Ability to sort the list/grid by first\_brewed date and abv

### **Single page**

This view shows the information for a single beer. It will need to:

- Present a view of a single beer
  - Should show image, name, description and a table of ingredients

### **Additional requirements**

We would like to see some examples of how you would write tests for your SPA. You don't need to have tests written for the entire app, a single component fully tested to show a solid understanding of how to write good quality tests will suffice.

### API

BrewDog provides a free to use API that allows access to a full list of their beer recipes. The API and documentation can be found at <https://punkapi.com/documentation/v2>. No authentication or API key is required, however there are some limitations on the number of requests. This should be taken into consideration when implementing the search functionality.

### Tech stack

To create this SPA it is mandatory that you use a Javascript framework, the choice of framework is up to you. Here at Pollen8 we use Vue.js as our framework of choice, however we are looking for

exemplary use of any framework. The focus of this exercise is the result of writing the app, not configuring the environment for it, so we recommend you use a pre-configured environment.

You should use a version control repository such as Github or Bitbucket to submit your solution.

### UX/UI

Although the primary function of this technical test is to assess your javascript skills, some points will be awarded for layout and design principles applied. Feel free to use a UI/Component library to pretty up the design. One thing we will be looking closely at is that the app works across all devices sizes following responsive layout best practices.

### Evaluation

Your project will be reviewed by one or more of our senior developers, who will try to understand the way you approach development and use this tech stack. It's important for us that you understand the technology you're using and can justify the decisions you made while producing the solution. Whilst this is a simple exercise, you should approach it as if this is a feature you'd be producing within a team, making sure you follow best practices, the code is SOLID, DRY and easy to maintain.

If we determine you've accomplished the exercise in a successful manner, it will be followed up by an interview where we will discuss in detail the approach you took to tackle some of the proposed problems and other technical aspects.

Good luck and thank you for your interest in Pollen8.