

Goal:

Use a tool such as codesandbox.io to write a react component that implements filters based on the query parameter. Simulate an ajax call on page load to fetch some data using the Promise API. Please spend **no more than one hour** on this exercise. The idea is to test your knowledge of React.js, but also to demonstrate familiarity with the native javascript libraries, browser API and CSS.

Problem:

We want to build a page to render questions that can be filtered by topics such as what company and what position they refer to.

We need the following filters implemented as query parameters in the URL:

q = search term (eg. ?q=foo)

co = company filter (eg. ?q=foo&co=facebook,linkedin,google)

pos = position filter (eg. ?q=foo&co=facebook&pos=software+engineer,data+scientist)

We also need a 'reset' button that when clicked will remove the filters by clearing the query params from the url.

The data should be of the format:

```
[{
  title: "a hard interview question",
  tags: ["sql", "probability", "modelling"],
  companies: ["facebook", "linkedin", "netflix"],
  positions: ["software engineer", "Business Analyst", "Data Scientist"]
},
{
  title: "easy question to solve",
  tags: ["sql", "probability", "statistics"],
  companies: ["facebook", "linkedin", "amazon", "google"],
  positions: ["software engineer", "marketing Analyst", "Data Scientist", "ML Engineer"]
}]
```

A summary of what the page should do is as follows:

1. Load the page.
2. On load, get the data from the "backend" (in this case, just mock the response)
3. save the data into state
4. Parse the query parameters from the URL
5. Filter the data by the query parameters
6. Render the data and filters
7. Add a 'reset' button that removes the filters and modifies the URL to be /

Expected behavior

1. Clicking a filter should apply it and change the url accordingly, re-rendering the output.
 - a. Clicking subsequent filters should apply those as well.
2. Clicking reset should reload the page and change the url.

url	output
/	Show all question titles
/?q=foo	Show only question titles containing “foo” as a substring in the title
?q=foo&co=facebook	Show only questions containing “foo” in the title and where the company list includes facebook
?q=foo&pos=software+engineer	Show only questions containing “foo” in the title and where positions list includes “software engineer”
?q=foo&&co=facebook&pos=software+engineer	Show only questions containing “foo” in the title and where positions list includes “software engineer” and the company list includes “facebook”

Sample UI

questions

facebook google software engineer data scientist reset

a hard question

easy question to solve

Fig 1. Filters shown in grey, question titles shown in white

Bonus points

1. Add styling to the page using css
2. Reduced usage of third party libraries
3. Add active states, hover effects, animations and transitions for the questions and filters
4. Add a way to disable a filter by clicking it again

When completed, submit your link on Typeform.