# Cornershop Frontend Test

## Before you begin

You will need to create a repository using the base project that we provided and invite as collaborators: @daslaf @fyanezv @redthree @raerpo @erasmo-marin

**PLEASE CONSIDER THAT WE EXPECT YOUR SOLUTION TO BE PRODUCTION-READY. IN OTHER WORDS, THAT MILLIONS OF USERS WOULD BE THRILLED TO USE YOUR PRODUCT.**

## The Test

You need to create a simple counter application that can do the following:

- Add a named counter to a list of counters
- Increment any of the counters
- Decrement any of the counters
- Delete a counter
- Show a sum of all the counter values
- Implement sorting counters by title and by amount
- It must persist data back to the server (using the API we're providing you)
- A text based searchbar
- A section where we can filter counters by the following rules:
  - Less than a given value that we can change (let's say in an input)
  - Greater than a given value that we can change (let's say in an input as well) - We must be able to toggle the filters in the UI

Some other notes:

- The design, layout and UX is all up to you. We´re looking for good concern for the UI/UX of the app.
- You must use React.
  - Using CLI tools to bootstrap your project (like Create React App) is recommended so you don't waste time setting up a build system.
  - If you decide to use a precompiler of any kind (js/css/etc..) we need to be able to run it with `npm run build`.
- Good management of state throught out the app using built-in solutions or third party dependencies (`redux`, `mobx`, `ngrx`, `vuex` or whatever).

- You can change anything you want (server stuff included) as long as the list above is completed.
- This isn't a backend test, don't make it require any databases.
- Your project must be self-contained, hence we don't want to run any `npm install -g whatever` commands. **NO GLOBAL DEPENDENCIES**

Extra points:

- Tests are good.
- Good practices in version control.

We have provided:

- Compiled Directory: of `/static/`.
- `/static/index.html` that will be served at `localhost:3000` when the server is running.
- `/static/app.js` and `/static/app.css` will be used automatically by `/static/index.html`.

If you need other publicly available files, other than `index.html`, `app.js`, `app.css` you will have to modify the server code in

## Install and start the server

```
$ npm install
$ npm start
$ npm run build #[optional] use for any precompilers you choose
```

## API endpoints / examples

The following endpoints are expecting a `Content-Type: application/json`

```
GET /api/v1/counters
# []

POST {title: "bob"} /api/v1/counter
# [
#   {id: "asdf", title: "bob", count: 0}
# ]

POST {title: "steve"} /api/v1/counter
# [
#   {id: "asdf", title: "bob", count: 0},
#   {id: "qwer", title: "steve", count: 0}
# ]
```

```
POST {id: "asdf"} /api/v1/counter/inc
# [
#   {id: "asdf", title: "bob", count: 1},
#   {id: "qwer", title: "steve", count: 0}
# ]

POST {id: "qwer"} /api/v1/counter/dec
# [
#   {id: "asdf", title: "bob", count: 1},
#   {id: "qwer", title: "steve", count: -1}
# ]

DELETE {id: "qwer"} /api/v1/counter
# [
#   {id: "asdf", title: "bob", count: 1}
# ]

GET /api/v1/counters
# [
#   {id: "asdf", title: "bob", count: 1},
# ]
```

***NOTE:* Each request returns the current state of all counters.**

This must be our face after reviewing your submission (hopefully, for good).

Pika Pika