```
~/Projects/SovTech
➜ npm start -- career
```

Fullstack Engineer Test (Chuck)

Focus	Fullstack
Last Edited	@Oct 13, 2020 6:58 AM
Ready	✓
Tech Stack	NodeJS React

At SovTech our Frontend teams are multidisciplinary, and we 💙 Fullstack JS. It saves us time, works incredibly well, and opens up many many doors. We've put this test together to afford Fullstack engineers at any level out there, the freedom to prove their skills 😎

Estimated completion time (skill dependant): **~ 6 - 8 hours**

SovTech.com | Custom Software Development Sorted | USA
SovTech.com is a leading custom software development company. New York and San Francisco. Mobile and Web Apps, Software Development Teams and Blockchain.
https://www.sovtech.com/

⚙️ What is the technical challenge?

We're looking for professionals who are very familiar with the stack we're using.

As this is a Fullstack engineer test, we've broken down the it down into two distinct parts, server and client:

Server:

1. You should develop an Apollo GraphQL API
2. Your GraphQL API should wrap the ChuckNorris.io API
3. Your GraphQL API should have a Query type that resolves all Categories (https://api.chucknorris.io/jokes/categories)
4. Your GraphQL API should have a Query type that resolves a random joke given a Category as an argument (https://api.chucknorris.io/jokes/random?category={category})

Client:

1. You should develop a Single Page App (SPA)
2. Your SPA should consume the above GraphQL API
3. Your SPA should have a home page with a list of categories
4. When a category is clicked, the category detail (a random joke) should be displayed appropriately

👮 What are the requirements?

For this application, you should, at the least use:

- TypeScript
- React (LTS)

- Apollo Client (react-apollo)
- Apollo Server
- NodeJS (LTS)
- Centralised state management (Redux/Context API) any state management framework/tool can be used
- Your centralised store must be immutable and make use of the Action/Reducer pattern
- You can use a boilerplate of your choice
- Your solution should be checked into a public Github repo. Additionally the README should outline all necessary steps required to bootstrap the code
- Your solution can be implemented in CodeSandbox or similar
- Once complete, a link to your submission along with any other resources should be emailed to us

✅ How will you be scored?

This test we feel allows you to express your resourcefulness with many Fullstack elements. It is for this reason we have left it pretty open.

Aspects to note:

- A lot of emphasis will be on your familiarity with modern JavaScript (React), NodeJS and GraphQL development, tooling and techniques
- There is no required design from our side - however, aesthetic and experience is important and should be considered accordingly
- Most best-practices and standards should be portrayed in your file organization,
  methods, naming conventions etc.
- Your ability to execute the required task

💅 Are there any other considerations?

Again, use this test as an opportunity to express yourself. Here are some of the things we use and like - only to keep in mind, of course 😄

- We like Git
- We like all the new React things (Hooks, Context, Suspense etc.)
- We believe in quality code (think tests)
- We always consider responsiveness
- We like CSS-in-JS (think Styled Components)
- We always go above and beyond
- We like all the new GraphQL things (think Prisma, Hasura etc.)
- We value security, strongly
- We like Serverless
- We like Typescript

🔒 Closing

Thank you for taking the time to complete this test. We look forward to your submission. If you have any feedback, please feel free to share it with us. As we are continuously improving our hiring process.