

JAVASCRIPT DEVELOPER EXERCISE INSTRUCTIONS

Overview

You will be consuming the open source [jokes API](#) to fetch and display a random joke. Your work can be delivered via a publicly available [CodePen](#), [JSFiddle](#), [CodeSandbox](#), or [GitHub](#) repo. You should be limiting your time taking this code test to 2 hours.

What we are looking for:

- ▶ Approaches to pulling data from the API
- ▶ Approach to rendering the results and different states
- ▶ Well-crafted, accessible HTML
- ▶ Re-usable CSS and JavaScript
- ▶ Approach to using libraries/utilities as an assistive tool
- ▶ Responsiveness (smartphone portrait/320px and up)
- ▶ Adding your own touch through animations and transitions
- ▶ Attention to detail

What we are not looking for:

- ▶ Browser testing (we'll be reviewing in Chrome only)
- ▶ Completing some of everything with nothing displaying or rendering
- ▶ Tooling (you can use any build setup, starter, template, or task runner you're comfortable with)

Instructions

- ▶ Login or create a free account at [CodePen](#), [JSFiddle](#), [CodeSandbox](#), or [GitHub](#)
- ▶ Create your project or repo using one of the services above
- ▶ You must use Typescript and Material UI to complete this assessment.

- ▶ You can use any open source library or package that is publicly available
- ▶ You can use any flavor of CSS you want (vanilla CSS, Sass, Less, CSS In JS, etc.)
- ▶ Take the code test following the functionality/acceptance criteria listed below
- ▶ Email the link to the public CodePen, JSFiddle, CodeSandbox, or GitHub repo back to your Mercury point of contact when 2 hours have passed
- ▶ Do not edit your code test after the 2 hours have passed and you have sent the link to Mercury

Functionality/Acceptance Criteria

- ▶ The purpose of the screen is to load and show a random joke from the Jokes API
 - API docs can be found at https://github.com/15Dkatz/official_joke_api
 - Reference the attached mockup for layout and styling that you should be matching
 - The screen should be responsive, fitting a mobile device screen up to a desktop screen (note that the mockup only shows desktop layout and styling)
- ▶ On initial screen load, show the setup (first sentence) of a random joke
 - Show a loading state while data is being retrieved from the API
 - Show error state if the API is down
- ▶ Display a "Show Punchline" button
 - Show the punchline (second sentence) of the joke
 - The "Show Punchline" button should now read "Hide Punchline"
 - Clicking "Hide Punchline" should hide the punchline of the joke
- ▶ In the header of the screen, display a "Get A New Random Joke" button
 - Clicking this button should retrieve and show a new random joke
 - The "Hide Punchline" should reset to "Show Punchline" and the punchline should be hidden
 - Loading and error states should still show as described above
- ▶ In the header of the screen, display a link to the API documentation with the text "View API docs"
 - Clicking the link should open the URL in a new tab/window