



CPL/Featurespace Front-end Coding Task.

Please write a web application in any JavaScript framework or Elm that allows a user to query and show details for a given UK postcode.

No backend implementation is required, and no persistent state – this is entirely a client application.

The app should use the freely available (no API key needed) JSON REST API at <http://postcodes.io>

Feel free to use any libraries you feel suitable, EXCEPT for any published that directly relate to postcodes.io – we're interested in how you build an application to directly interact with a REST API.

Details:

- The main page should consist of a form with a text input field for the user to enter a postcode.
- After submitting the form, the application should show:
 - The country and region for the submitted postcode: API path `/postcodes/{POSTCODE}`
 - A list of the nearest postcodes, and their countries and regions: API path `/postcodes/{POSTCODE}/nearest`
- The application should either work by entering a postcode in the form, or by browsing directly to `<app_hostname>/<postcode>`
- The application URL should update to `<app_hostname>/<postcode>` after submitting the form.
- Possible things to consider, time-depending:
 - Unit tests
 - Postcode validation
 - Error handling (eg API failure / timeout)
 - Suggestions of postcodes (autocomplete method as you type)

Test postcode to use: **CB4 0GF** (Featurespace Cambridge office)

Please pay attention to how you structure your code - we will be looking at how you chose to solve the problem, how clean and well-structured your code is, and how easily it could be extended and adapted.

This should take no more than 60 minutes. Please note how long it took you, any assumptions you made, and also if you'd had more time which things you might improve.

Either upload the code to an online collaboration site like <http://jsfiddle.net> and send us the link in advance, or bring your code on a memory stick. It must work, and we will ask you to run and talk us through your code in the interview, and perhaps make some changes.

Please write at least 2 unit tests - don't go overboard, but show us an example or two of some edge cases or things that could go wrong.

You may use any publicly available libraries you feel suitable, but you must bring all dependencies with you (or use something like NPM). You may assume you will have internet access.