Create React App End-to-End Tests

Usage

These tests ensure various functionality contracts are upheld across dependency upgrades.

To get started locally, run `npx jest test/ --watchAll`.

It's suggested that you filter down tests to avoid re-running everything. The most common tests will be the webpack messages.
To only run the webpack messages, type `p` followed by `webpack-message` and press `[enter]`.

How do these work?

`fixtures/`

Each `fixture/` gets spun up in a temporary directory and has its dependencies installed with Yarn PnP (for speed).
To opt-out of PnP, create a `.disable-pnp` file in the specific fixture directory.

A global (`testSetup`) is created which has a few interesting properties:

- `testSetup.testDirectory`: the directory containing the test application
- `testSetup.scripts`: an object allowing you to invoke `react-scripts` commands and friends

All tests for each `fixture/` are then ran.

`testSetup.scripts`

`start`

This will run the `start` command, it can be ran asynchronously or blocking if `{ smoke: true }` is used.
If ran asynchronously, it will return the `port` and a `done` function to clean up the process. If ran blocking, it will return the `stdout` and `stderr` of the process.

`build`

This will run the `build` command and return the `stdout` and `stderr` of the process.

`test`

This will run the `test` command and return the `stdout` and `stderr` of the process.

`serve`

This will run serve the application. It will return the `port` and a `done` function to clean up the process.