DevOps22 - CI/testing Assignment 1

Per Arn, Philip Salqvist

Agenda

Introduction

- Why testing?
- The downsides of not testing.
- Web applications require more rigorous testing.

What to use where

Express based backend

API endpoint testing

Supertest

React based frontend

Integration testing

React Testing Library

NodeJS

Unit testing

Jest, Mocha, Jasmine

What to use where

- Unit test
- Integration test
- API end-point testing

Unit testing

- Jest
  - Developed by Facebook and used by Spotify, Twitter and Instagram to only name a few.
  - Runs standalone.
  - Efficient using parallel testing.
- Jasmine
  - Mature framework.
  - Easy to use for TDD.
  - Downside - complex configuration.
- Mocha
  - Good for flexibility.
  - Runs tests serially.
  - Downside - depends on the
    use of other libraries for assertions.

Integration testing

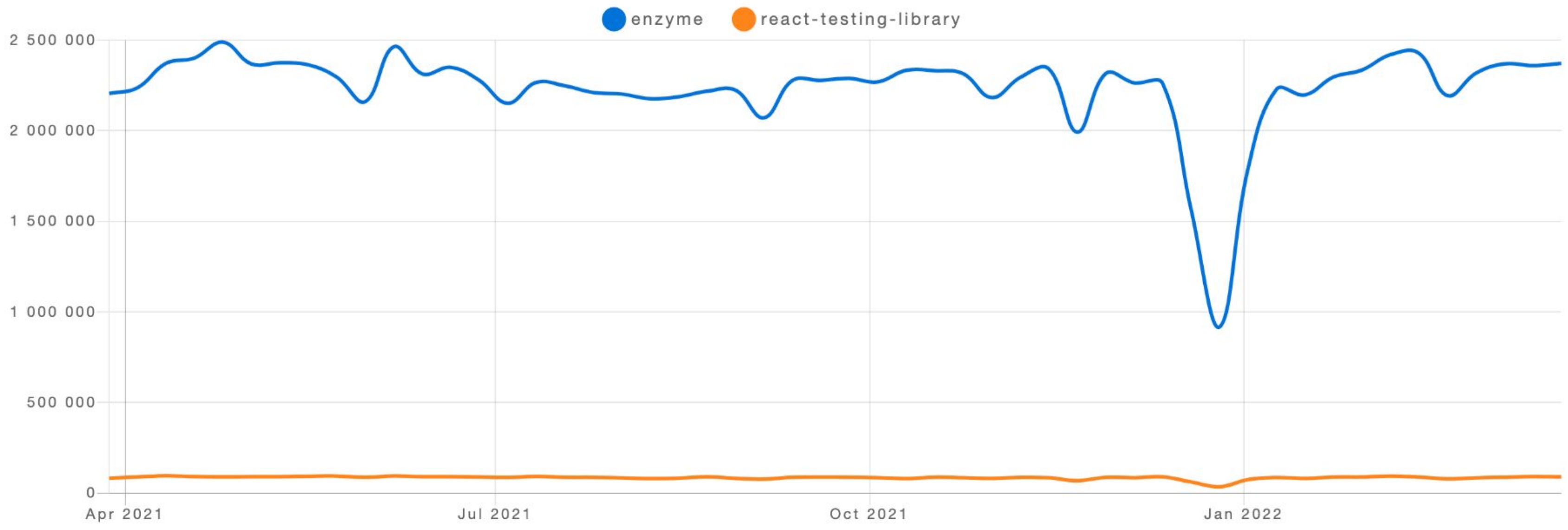
With integration testing we simulate the DOM tree, fire events on specific elements in the DOM and assert that we get the desired outcome.

- Shallow rendering
  - Each component rendered, is simulated in the DOM tree, without its corresponding children.
  - Each component needs a seperate test - not very efficient.
- Full rendering
  - A component, or entire screen, can be rendered in DOM simulation, and interactions with child component can be asserted.

Integration testing

- React Testing Library
  - Supports full rendering
  - Easy to query nodes in the dom, and intuitive to fire events and assert the outcome
- Enzyme
  - Was developed specifically for React, and is now standard in the community
  - Although, indications show it is badly maintained and could become obsolete

Integration testing

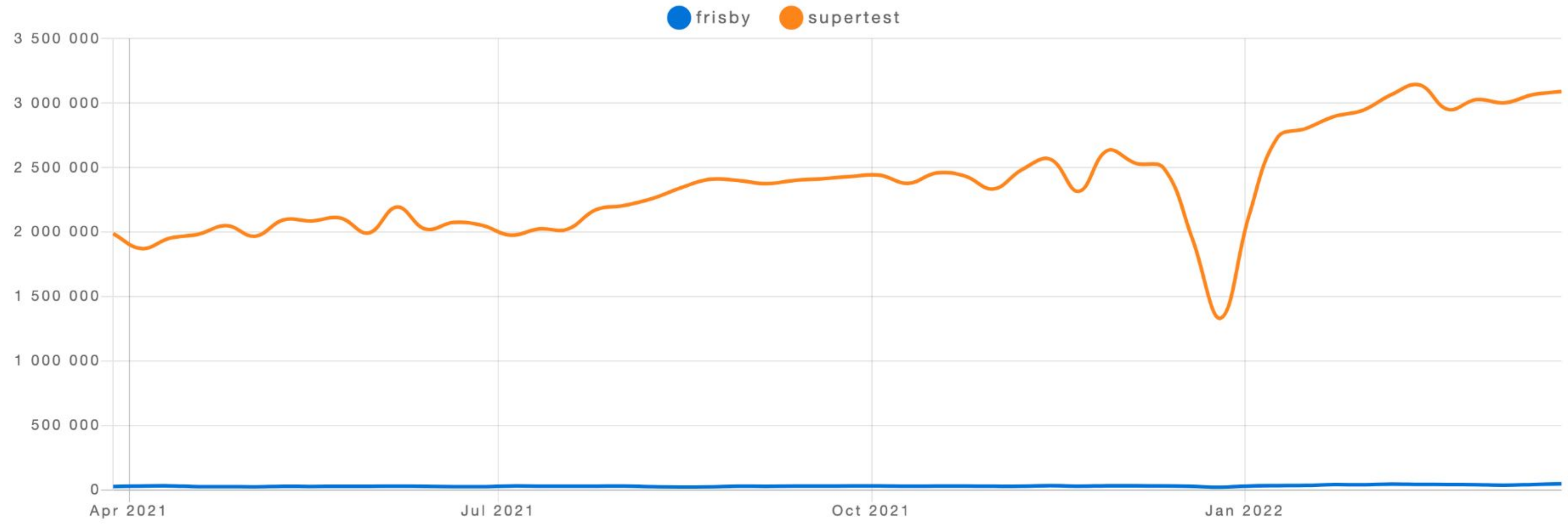


API end-point testing

We can use a framework to enable out of the box functionality to test different http-requests against our end-points

- Frisby
  - Built on top of Jasmine
  - Only supports CRUD-operations
- Supertest
  - Supports non CRUD-operations
  - Widely used

API end-point testing



Trade offs

- Different frameworks provide different features.
- Using the same library for both FE/BE?
- Using libraries that aren’t continuously being updated.

Conclusions

- Testing is important.
- Testing is a bit different in web applications compared to what most of us are used to.
- There are many different testing frameworks.
- Use those proven to be good and used by many.
- React Testing Library - RTL for front end
- SuperTest - for back end

Thank you very much for your attention!