E2E testing

* E2E testing (End to End testing)
  + Testing your entire application
  + For your front end to work it means that your backend has to also be up and working
    - If my API doesn’t work then clicking a button that is supposed to get information from the backend will not work even if all the JS is correct
      * API won’t work if services don’t
        + Services don’t work if DAOs don’t

DAOs don’t work if that data don’t

* + You can manually do E2E testing
    - You as a human clicking buttons and checking results
      * Cons
        + It can take a while
        + It uses time that is better spent writing features
        + Can be error prone (input your information incorrectly)
  + You can automate your E2E tests
    - You can write a program/script that can perform the actions a user could perform on the web pages
      * Cons
        + It is skill that takes time to learn to automate e2e tests
        + It can take a while to set up
      * Pros
        + Much faster to execute than a human
        + Less error prone
        + Can run these test at any time
        + Cheaper in the long run
* How to perform automated e2e tests
  + Selenium
    - Browser automation software
    - Allows us to program user actions for a web page
    - Webdriver is the main piece of software that allows selenium to perform automation
      * You need a different web driver for each browser type
      * It is the main object to perform automation tasks in selenium
        + Abstracted away in protractor
  + Protractor (Designed for testing Angular webpages in particular)
    - Works for any webpage
    - Library that uses selenium to automate user actions

Node.js

* Node.js
  + Node is a runtime environment of JavaScript on the server side
  + This allows you to create backend applications/APIs written in JS
* Structure of a node application
  + Every node application has a main.js file (technically can be named anything)
    - It is the entry point of the application (like a main method in Java)
  + Package.json
    - This is essentially node’s version of a pom.xml
    - It keeps track of dependencies of your application
    - It also has meta information like the name and author of the application