**Creating Testing Framework**

Feature Testing/Browser Testing

Server Testing

NPM Packages

* Chai
  + Provides an advanced assert library
  + const {assert} = require(‘chai’);
    - assert.equal(response.status, 200);
* SuperTest
  + Provides the functionality required to test APIs. Can make get/post requests.
  + const request = require('supertest');
    - const response = await request(app).
    - get('/');
* JSDOM
  + Allows you to create the parseTextFromHTML helper function which will take your response and all you to find the text that is to be rendered within the HTML.
  + const {jsdom} = require(‘jsdom’);

1. Test status codes - [httpstatuses.com](https://httpstatuses.com/)

describe('root page', () => {

describe('GET request', () => {

it('returns a 200 status', async () => {

const response = await request(app).

get('/');

assert.equal(response.status, 200);

});

});

});

1. Response content
   1. Tests that exercise the ‘happy path’ – the expected use cases of the application.
   2. Tests that exercise the ‘sad path’ – unexpected or invalid use of the application.
   3. Many servers return dynamic HTML content based on the user, the URL accessed, header values, and more. We use TDD to ensure the server responds correctly for each case.