

< Project:
Feed Reader Testing



Project Details



- ✓ 1. Project Overview
- ✓ 2. Project Details
- ★ 3. Project: Feed Reader Testing

How will I complete this project?

Review the Feed Reader Testing [Project Rubric](#)

1. Take the JavaScript Testing [course](#)
2. Download the [required project assets](#).
3. Review the functionality of the application within your browser.
4. Explore the application's HTML ([./index.html](#)), CSS ([./css/style.css](#)) and JavaScript ([./js/app.js](#)) to gain an understanding of how it works.
5. Explore the Jasmine spec file in [./jasmine/spec/feedreader.js](#) and review the [Jasmine documentation](#).
6. Edit the `allFeeds` variable in [./js/app.js](#) to make the provided test fail and see how Jasmine visualizes this failure in your application.
7. Return the `allFeeds` variable to a passing state.
8. Write a test that loops through each feed in the `allFeeds` object and ensures it has a URL defined and that the URL is not empty.
9. Write a test that loops through each feed in the `allFeeds` object and ensures it has a name defined and that the name is not empty.
10. Write a new test suite named `"The menu"`.
11. Write a test that ensures the menu element is hidden by default. You'll have to analyze the HTML and the CSS to determine how we're performing the hiding/showing of the menu element.
12. Write a test that ensures the menu changes visibility when the menu icon is clicked. This test should have two expectations: does the menu display when clicked and does it hide when clicked again.

Mentorship
Get support and stay on track

2



Project:
Feed Reader Testing



Project Details



1. Project Overview



2. Project Details



3. Project: Feed Reader Testing

completes its work, there is at least a single `.entry` element within the `.feed` container.

15. Write a test suite named `"New Feed Selection"`.
16. Write a test that ensures when a new feed is loaded by the `loadFeed` function that the content actually changes.
17. No test should be dependent on the results of another.
18. Callbacks should be used to ensure that feeds are loaded before they are tested.
19. Implement error handling for undefined variables and out-of-bound array access.
20. When complete - all of your tests should pass.
21. Write a README file detailing all steps required to successfully run the application. If you have added additional tests (for Udacious Test Coverage), provide documentation for what these future features are and what the tests are checking for.

Useful Links

- [Udacity Front End Nanodegree JavaScript Style Guide](#)
- [FeedReader Testing Project Rubric](#)

NEXT

Mentorship

Get support and stay on track

2