**Unit Testing Event Listeners on Form Input Fields with Karma**

[Karma](https://karma-runner.github.io/4.0/intro/how-it-works.html) is a JavaScript test runner where you can run your unit tests on headless chrome or real browsers. Because it runs on the real browser, it supports the browser web APIs and makes it easy to test front-end JavaScript. With Karma, the environment where unit tests run is truer than, say, using [jsdom](https://github.com/jsdom/jsdom) as it uses the real browser. Krama is often associated with Angular, but it is a really nice test runner for regular front-end JS code as well. It works well with CD/CI pipelines as it spawns a headless chrome for test execution environment.

Karma works well with TypeScript. Setting up is not too difficult. It simply uses a webpack to transpile TS into JS and runs the test. Once it is set up, it works exactly the same as JS. You can check my example of [karma.config.js for TypeScript](https://github.com/mydatahack/web-accessibility-form-typescript-spa/blob/master/karma.conf.js). Dependencies to install can be found [here](https://github.com/mydatahack/web-accessibility-form-typescript-spa/blob/master/package.json).

The focus on this post is to write unit tests for the event listeners attached to select, radio and checkbox input. Because Karma runs on the actual browser, we do not need to learn special APIs (like you need to do with Jest & Enzyme for React) to do browser actions. We can just use the vanilla JavaScript to programmatically update input fields and dispatch actions.

Let’s get started.

1. **Select Input Event Listener Tests**

We have a select element event listener that adds different classes.

<script src="https://gist.github.com/mydatahack/05e5f9716f97899ed106b0ecba972f9a.js"></script>

Let’s unit test this. The most important thing about select element is that the value must exist in the child option elements. If we put non-existing value, the value will not change.

<script src="https://gist.github.com/mydatahack/a8fbe7d6a9298dc55ebe3f541a2107d4.js"></script>

1. **Radio & Checkbox Input Event Listener Tests**

Here is the simple event listener class to listen to the change on radio & checkbox input and update aria-checked attribute on click.

<script src="https://gist.github.com/mydatahack/3b2604072989af4d57a513dbc6773afe.js"></script>

We can unit test the public method where it attaches event listeners to elements. The testing is fairly simple, updating value and dispatching action.

<script src="https://gist.github.com/mydatahack/2846c93f0cdabd643e00c6c93d392b50.js"></script>

You can see more comprehensive Karma unit test example [here](https://github.com/mydatahack/web-accessibility-form-typescript-spa/tree/master/src/ts/components/test). Karma uses [Jasmine](https://jasmine.github.io/) as a testing framework out of the box. To write unit tests, [cheatsheet](https://devhints.io/jasmine) is always your best friend!