Install Jasmine:

1. To install Jasmine run:

**npm** i jasmine

1. Add a reporter for outputting Jasmine results to the terminal:

**npm** i jasmine-spec-reporter

1. Add type definitions for Jasmine with :

**npm** i --save-dev @types/jasmine

Add Testing Scripts:

* Find the scripts object in the package.json and add the following to run jasmine:

"jasmine"**:** "jasmine"

Set Up the File Structure:

1. In the root directory of the project, create a folder named spec.
2. In the spec folder, create a folder named support.
3. In the support folder, create a file named jasmine.json to hold the primary configurations for Jasmine.
4. In the src folder, add a folder named tests.
5. In tests add a file named indexSpec.ts to hold the tests for code in the index.js file.
6. In the tests folder, add another folder named helpers.
7. In helpers, add a file named reporter.ts. This will be the primary configuration for your spec reporter.

**File Structure**

├── node\_modules

├── spec

│ └── support

│ └── jasmine.json

├── src

│ ├── tests

│ │ ├── helpers

│ │ │ └── reporter.ts

│ │ └── indexSpec.ts

│ └── index.ts

├── package-lock.json

├── package.json

└── tsconfig.json

**Best Practices For File Naming**

When creating files for tests, a best practice is to name the .ts file the same as the .js file to be tested with Spec appended to the end. The more tests needed to be run, the more test files will need to be created. Be sure to follow this best practice to keep track of the test file that contains the tests for each .js file.

In reporter.ts, add the following code from the[**jasmine-spec-reporter TypeScript support**](https://github.com/bcaudan/jasmine-spec-reporter/tree/master/examples/typescript) documentation to configure the reporter to display Jasmine results to your terminal application. These are default settings and can be adjusted to meet your specific needs. The documentation on [**GitHub**](https://github.com/bcaudan/jasmine-spec-reporter) provides more options available.

**import** {DisplayProcessor, SpecReporter, StacktraceOption} **from** "jasmine-spec-reporter";

**import** SuiteInfo = jasmine.SuiteInfo;

**class** **CustomProcessor** **extends** **DisplayProcessor** {

**public** **displayJasmineStarted**(info: SuiteInfo, log: string): string {

**return** `${log}`;

}

}

jasmine.**getEnv**().**clearReporters**();

jasmine.**getEnv**().**addReporter**(**new** **SpecReporter**({

spec: {

displayStacktrace: StacktraceOption.NONE

},

customProcessors: [CustomProcessor],

}));

In the jasmine.json add the following configurations for a basic Jasmine configuration:

{

"spec\_dir": "dist/tests",

"spec\_files": [

"\*\*/\*[sS]pec.js"

],

"helpers": [

"helpers/\*\*/\*.js"

],

"stopSpecOnExpectationFailure": false,

"random": false

}

In the tsconfig.json file, add "spec" to the list of folders that we want to exclude.

"exclude": ["node\_modules", "./dist", "spec"]

### Write a Basic Test

We'll start with a simple test:

***index.ts*** ```typescript const myFunc = (num: number): number => { return num \* num; };

**export** **default** myFunc;

```

We can write a simple test for the function:

***indexSpec.ts*** ```typescript import myFunc from '../index';

**it**('expect myFunc(5) to equal 25', () => {

**expect**(**myFunc**(5)).**toEqual**(25);

});

```

To test this we'll need to first run the build script and then the test script:

**npm** run build

**npm** run jasmine

Or we can combine the two into one script in our package.json file:

"test": "npm run build && npm run jasmine"

### Troubleshooting

It is not uncommon for conflicts to arise between NPM packages as authors update or add functionality or as the packages that these packages depend on are updated. When you get an error, you can either look for the error to see if it has been reported and follow a solution offered, use older stable versions, or attempt to remedy the issue and submitting a solution.

#### Example:

Shortly after this course was released, an update to jasmine-spec-reporter caused the code shown in the demo to break. The issue was resolved within a few weeks, but while it was being fixed, students needed to use a workaround. You can read about the issue and the resolution here: [**https://github.com/bcaudan/jasmine-spec-reporter/issues/588**](https://github.com/bcaudan/jasmine-spec-reporter/issues/588)

At the time of writing, the current version of jasmine-spec-reporter in the [**npm package is v7.0.0**](https://www.npmjs.com/package/jasmine-spec-reporter). However, the example codes use [**jasmine-spec-reporter@6.0.0**](https://www.npmjs.com/package/jasmine-spec-reporter/v/6.0.0) version. There is [**a breaking change**](https://github.com/bcaudan/jasmine-spec-reporter/blob/master/CHANGELOG.md) in these two versions. Please check package.json file to ensure you are using v6.0.0 version. Or, you can modify the starter codes to work with v7.0.0 as shown in [**this Knowledge post**](https://knowledge.udacity.com/questions/582324).