

TESTING!

Best Practices

- Testing isolated small pieces of code - Most should be unit tests
- Async tests - Promise (microtask), Timer, XHR (HttpClient)
 - Tests can complete too early. Use “done();”
 - Async await
 - done(); can't handle errors. Use async await.
 - You get an opportunity to write an asynchronous code in synchronous fashion.
 - fakeAsync
 - looks async but is synchronous
 - tick() simulates the asynchronous passage of time.

Future

- AOT Test - use aot mode in unit test, much faster.

JEST Test Runner

- Benefits:
 - Easy to setup, VERY fast
 - `npm install jest-preset-angular`
 - VERY nice output :)
 - Snapshot testing - lets you automate creating assertions.
 - writes out a good snapshot to GIT
 - `expect(state).toMatchShapshot()`.
- Cons:
 - Trust, but verify snapshots.
 - No way to log that the snapshots were verified
 - VSCode plugin snapshottools

TESTING!

Best Practices

- Testing isolated small pieces of code - Most should be unit tests
- Async tests - Promise (microtask), Timer, XHR (HttpClient)
 - Tests can complete too early. Use “done();”
 - Async await
 - done(); can’t handle errors. Use **async await**.
 - You get an opportunity to write an asynchronous code in synchronous fashion.
- fakeAsync
 - looks async but is synchronous
 - tick() simulates the asynchronous passage of time.

Future

- AOT Test - use aot mode in unit test, much faster.