When a Test Fails

In this lesson, we'll learn about what to do when a test fails in Snapshot Testing.

WE'LL COVER THE FOLLOWING

The snapshots generated are the source of truth for deciding if a test is valid or not. That's the way regressions are checked, and in the end, it depends on your criteria.

Example

For example, go to the `ContactBox.vue` component and change the `fullName` computed prop to be separated by a comma:

ContactBox.vue

```
fullName() {
  return `${this.name}, ${this.surname}`;
}
```

If you run the tests again, some of them will fail since the rendering result is different than before. You'll get an error like this:

```
The received value does not match stored snapshot 1.

 - Snapshot
 + Received

   <div
     class=""
   >
```

```
    <p>
-     John Doe
+     John, Doe
    </p>
  </div>
```

From that point on, as it usually is while testing, you must decide if that's an intentional change or if it's a regression. You can press *'u'* in order to update the snapshots:

```
Snapshot Summary
 › 1 snapshot test failed in 1 test suite. Inspect your code changes or press `u` to
update them.

Test Suites: 1 failed, 1 total
Tests:       1 failed, 2 passed, 3 total
Snapshots:   1 failed, 1 passed, 2 total
Time:        0.99s, estimated 3s
Ran all test suites related to changed files.
```

It would be convenient when applying TDD to use the watch mode `npm run test -- --watch`, since Jest gives you some options to update snapshots:

- Press **'u'** to update all snapshots.
- Press **'i'** to update snapshots interactively, one by one.

Conclusion

Snapshot testing **saves you a lot of time**. This example was basic, but imagine testing a more complex component with many different rendering states.

Sure, you can assert on the specific things, but that's much more cumbersome than asserting how the component is rendered depending on the state, since most of the times if you change the code you must change the assertions on the tests, while with snapshot testing you don't need to.

Additionally, you can **find regressions** that you didn't take into account. Maybe something you didn't consider in your tests has changed the rendering of the component, but the snapshots will alert you about it.

Here are some **caveats** that you should remember:

- Snapshot testing doesn't replace specific assertions. While it can work most times, both ways of testing are totally combinable.
- Don't update snapshots too easily. If you see that a test fails because it doesn't match a snapshot, take a deep look at it before updating it too quickly. I've been there as well.

Let's test a running project of what we have done so far in the next lesson.