



Davide de Paolis

Posted on 20 de out. de 2020



2



1

Do you do Snapshot Testing?

#discuss #testing #bestpractice codequality

And more specifically do you write Snapshot tests for your Infrastructure as Code (like AWS CDK stacks)?

The context

Snapshot testing is a type of “output comparison” or “golden master” testing.

Basically, you freeze a specific state/view/characteristic of your application and compare it overtime to make sure it did not change. **Snapshot tests do not test the behavior**, the fact that with a given input the output is always the same.

They just make sure that the current state/representation is the same **as the one you have stored**. If it is not, you are asked if you want to overwrite the saved snapshot and replace it with the current result.

Over the years I **can count on the fingers of one hand**, the times when I actually used Snapshot tests:

- when we were refactoring the basic engine of one of our games and we wanted to be really really sure we weren't changing anything of how that complex UI was looking. Once the refactoring was done, we dropped the snapshot testing because changes to the UI were part of every sprint, so it did not make sense.
- when we just started with React, and we were really excited to take advantage of the Snapshot testing functionality for our React components.
- when we started using AWS CDK to build our Serverless Infrastructure. In this case, the snapshot is not that “Visual”, like a picture of the UI, rather a comparison of the JSON output of Cloudformation.

The problem:

As soon as you change something, not even in the stack itself but in the lambda code (since the hash of the uploaded asset will be different), the test would fail. Therefore, you have to rerun the test, forcing/confirming and you have to run the test locally, confirm the update of the snapshot, and push.

The difference with Unit (or integration tests etc) is that with those, usually, you have to update the tests only if something changes **either in the inputs or in the expected outputs**.

That is why we have tests in the first place, **to make sure that refactorings of the code itself won't affect behavior from the outside**.

I understand that if you add a new resource to your API, or change a policy to write to your bucket, you are modifying the architecture of your application, I understand that it requires an additional action from the developer to intentionally edit the snapshots. and I know that the snapshots will be version controlled, so you can quickly compare all the final deployed stacks. But I still don't see much benefit in it.

It could be handy for some small things like (very) small react components returning confirmation or error messages (instead of using regex to check for the match, match the snapshot)

<https://kentcdodds.com/blog/effective-snapshot-testing>

But for the entire stack or the entire screenshot of an application, snapshot updates would be required for ANY single change, so it will simply become part of the routine:

- change code
- run test
- get the warning snapshot test failed



Davide de Paolis

Follow

Trouble maker and Problem solver 🛠️
Loves simplicity, hates bullshit 🗑️
Productivity obsessed, avid learner 📖
Sport and outdoor freak 🏃⚡ Metalhead
👨👩 Father of 2 👶👶 Opinions are my own

LOCATION
Hamburg

WORK
Technical Lead at Goodgamestudios

JOINED
28 de jan. de 2019

More from Davide de Paolis

Revive your JSON (examples with TDD and composition)
#javascript #beginners #typescript #bestpractice

Solve TypeScript 'Not all code paths return a value' by properly using ReturnType[]
CodeReview Chronicles
#beginners #javascript #typescript #bestpractice

Code Reviews are bottlenecks. What is the point of Code Reviews?
#bestpractice #codereviews #techleadership #beginners

- update snapshot
- commit without much thinking. Therefore making the test **useless**.

The Question

I do not have yet a strong opinion about such tests, and I think it is nice to have them especially for some specific *critical happy paths* in your UI (like the login form, or shop page or for small CDK Constructs (that maybe you share with other teams), but it really adds a burden of work, and often gives **a false sense of security**, and when it comes to test the CDK I much prefer to have a smaller test checking for the existence of a specific Resource / Role rather than Snapshot test the entire thing)

Do you use them, how far do you go with them?

How extensively do you snapshot test your application?

How long have you been doing it (in some project I worked on, it was introduced, but then dropped just after a few weeks, for the reasons I mentioned above)


Reference

- [Improving Communication and Confidence with Snapshot tests](#)
- [Recognize the need for visual testing](#)
- [Jest Image Snapshot Docs](#)
- [How to test AWS CDK](#)
- [Benefits and Drawbacks of Snapshot testing](#)

Photo by [britt gaiser](#) on [Unsplash](#)

Top comments (3) ↕

Subscribe

 Add to the discussion


 **Per Lundholm** • 21 de out. de 20

I use it and I call it approval testing. I make up the input for the test and approve it if I like it. Ideal for GUI while I wouldn't do it for logic where it is better to decide the expected outcome first and let that drive the solution, aka TDD.

The thing with GUI is that there are so many curved balls that get thrown at you so you need to look through the GUI quickly. I use Backstopjs that runs my app, take snapshots for different screen sizes and reports the difference from earlier approved snaps. It is a visual report that highlights the difference. It is a quick thing to do.


It has helped me when I upgraded the framework I'm using. However, there's still the problem with the asynchronous rendering browser does. When is it done?

2 likes Reply

 **Davide de Paolis** 🇮🇹 • 21 de out. de 20

yep this is mostly how we used it for our UI, we had a tool take snapshot and highlighting the differences. was quite cool. but only when we were making changes to the code (refactoring the framework) and no visual changes were supposed to happen. as soon as we started working on the UI, adding buttons, removing graphics etc, it was a bit of a pain to maintain all the screenshots

1 like Reply


 **Kristijan Pajtashev** • 20 de out. de 20


I am using it and we had quite a bit part of the application covered. But works only for small pieces or when the application is in the maintenance stage. When talking about react, it can be used a bit more, but it would def need shallow rendering. Otherwise changing some small thing in reusable component and your PR goes from 50 lines to 500 lines.

2 likes Reply

[Code of Conduct](#) · [Report abuse](#)

Read next

 **Playwright's UI Mode - watch mode and time travel debugging**
Debbie O'Brien · May 5

 **Will Mojo🔥 Replace Python? 🤔**
SAIFULLAHIN · May 5



20 Tools for Building Stunning Websites without coding

Anmol Baranwal · May 6



Well-designed presentation

Tara Doridy · Apr 24

DEV Community — A constructive and inclusive social network for software developers. With you every step of your journey.

[Home](#) · [Listings](#) · [Podcasts](#) · [Videos](#) · [Tags](#) · [FAQ](#) · [Forem Shop](#) · [Sponsors](#) · [About](#) · [Contact](#) · [Guides](#) · [Software comparisons](#)

[Code of Conduct](#) · [Privacy Policy](#) · [Terms of use](#)

Built on [Forem](#) — the [open source](#) software that powers [DEV](#) and other inclusive communities.

Made with love and [Ruby on Rails](#). DEV Community © 2016 - 2023.