One Test for Multiple Salesforce® Environments

Profile

When many of your 9,000 customers routinely give you five-star ratings, you know you’re doing something right. That’s business as usual for Conga (getconga.com), which provides the number-one paid application on the Salesforce AppExchange (among 3,000+ apps). Conga’s products help organizations simplify and automate key processes involving document generation, process automation, data management, eSignature and contract management. For example, with the company’s Composer app, users can use templates to easily generate nicely formatted reports, which at runtime, display valuable Salesforce data that informs smart business decisions.

Knowing that a single app download could quickly be deployed to hundreds or thousands of employees, Conga is obsessed with digital quality. “It’s one of the biggest things there is. We’re really proud of our high NPS, thanks to our top-quality products and amazing support team. We definitely want to hold onto that and take care of our customers,” says Melissa Dixon, QA Automation Engineer at Conga.

In addition to designing Conga’s automation framework, Dixon is working on some custom automation and testing a new electronic signature app.
Supporting Classic and Lightning

Salesforce has more than three million subscribers and delivers three releases per year for two distinct user interfaces (UIs) — Classic and Lightning Experience. While Conga can control the features and functions of its products, it can’t control modifications to Salesforce. From a testing perspective, any little change in Salesforce could mean changes to Conga’s tests for all the same actions on both UIs. Ahead of each release, Conga’s small but mighty test team needed to ensure that the company’s products delivered a stellar user experience (UX) on both the Classic and Lightning versions — something the team was doing manually.

Testers could get creative to streamline their manual test processes, such as by tweaking the code behind Salesforce buttons to make them correspond to tests, but they were still doing double work and needed an automated testing solution. “We knew how helpful that would be, especially since the company was growing really fast,” notes Dixon. “We weren’t running on such a skeleton crew anymore, and could afford to slow down and dedicate some resources to speed up in the future to accommodate more people and potentially more products.”

Evaluating Eggplant

Conga’s team wanted a solution that could provide a solid development environment on a Mac or a PC. Dixon was familiar with Eggplant, and the test team spent a few months in trial mode with Eggplant Functional, evaluating it particularly for cross-browser testing capabilities.

For Dixon, Eggplant had two big advantages. “The image collections gave us the ability to have one test that worked in multiple environments, which was huge for testing Salesforce Classic and Lightning because they are two completely different interfaces, each with its own distinct DOM, visual elements, and text. Also, the ability to create a really accessible framework for testers and potentially anyone on a team — so we didn’t need a group of people just coding with Selenium all day.” She adds, “My vision is for anyone to be able to add a new test by just adding a line to the CSV that’s driving the table script. So people have more time to do things other than test automation.”

More Time, Better Coverage

Even with limited time to dedicate to automation, Dixon and her team got up and running right away with Eggplant and are making good progress. Automating with Eggplant frees testers from some repetitive, manual testing tasks, and helps them do a lot more cross-browser verification to vastly increase their test coverage.
According to Dixon, being able to test from the user perspective is a great way to combine testing science with creativity. “It’s so important to test the whole product, and testing it through the UI is the best way to do that,” she says. “As opposed to a Selenium script with 28-character-long IDs, by running and looking at an Eggplant script, you can tell what the user was doing, what the business logic was, and what the UX was intended to be. It’s really powerful when QA is the voice of the customer, and the more you test from that perspective, the more you’re likely to find quality issues, and deliver a better quality product.”

As Conga’s testers become more familiar with Eggplant, they’re learning about some of its more sophisticated aspects, which sparks ideas about different ways to approach testing for more flexibility in the future.

“So far, I haven’t found anything that Eggplant can’t do.”
MELISSA DIXON QA AUTOMATION ENGINEER AT CONGA