



HOW NEXMO INTEGRATED TESTRTC INTO THEIR TEST AUTOMATION FOR THE NEXMO VOICE API

NEXMO:

One of the top CPaaS vendors out there providing cloud communication APIs to developers, enabling enterprises to add communication capabilities into their products and applications

PRODUCT:

testRTC Testing (Regression Testing)

OBJECTIVE:

Provide regression testing for Nexmo Voice API

BENEFITS:

- End to end testing for WebRTC
- Ability to integrate with PSTN
- Integrate with an existing testing framework
- Supporting multiple users on the same project
- Visualizing and debugging issues found

Nexmo found in testRTC a solution to solve its end-to-end media testing challenges for their Nexmo Voice API product, connecting PSTN to WebRTC and vice versa.

nexmo® | The **Vonage®** API Platform

Nexmo is one of the top CPaaS vendors out there providing cloud communication APIs to developers, enabling enterprises to add communication capabilities into their products and applications.

One of Nexmo's capabilities involves connecting voice calls between regular phone numbers (PSTN) to browsers (using WebRTC) and vice versa. This capability is part of the Nexmo Voice API.

TESTING @ NEXMO:

Catering to so many customers with ongoing deployments to production means that Nexmo needs to take testing seriously. One of the things Nexmo did early on was introduce automated testing, using the pytest framework. Part of this automated testing includes a set of regression tests - a huge amount of tests that provide very high test coverage. Regression tests get executed whenever the Nexmo team has a new version to release, but these tests can also be launched "on demand" by any engineer, they can also be triggered by the Jenkins CI pipeline upon a merge to a particular branch.

At Nexmo, development teams are in charge of the quality of their code, so there is no separate QA team.

In many cases, launching these regression tests first creates a new environment, where the Nexmo infrastructure is launched dynamically on cloud servers. This enables developers to run multiple test sessions in parallel, each in front of their own sandboxed environment, running a different version of the service.

When WebRTC was added to Nexmo Voice API, there was a need to extend the testing environment to include support for browsers and for WebRTC technology.



When it comes to debugging, when something has gone wrong, testRTC is the first place we'd go look. There's a lot of information there

Jamie Chapman,
Voice API Engineer at Nexmo



ON SELECTING TESTRTC:

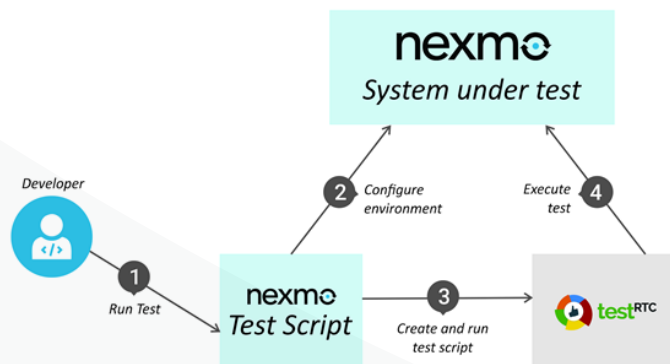
Nexmo needed WebRTC end-to-end tests as part of their regression test suite for the Nexmo Voice API platform. These end-to-end tests were around two main scenarios:

1. Dialing a call from PSTN and answering it inside a browser using WebRTC
2. Calling a PSTN number directly from a browser using WebRTC

In both cases, their client side SDKs get loaded by a web page and tested as part of the scenario.

Nexmo ended up using testRTC as their tool of choice because it got the job done and it was possible to integrate it into their existing testing framework:

- The python script used to define and execute a test scenario used testRTC's API to dynamically create a test and run it on the testRTC platform
- Environment variables specific to the dynamically created test environment got injected into the test
- testRTC's test result was then returned back to the python script to be recorded as part of the test execution result



This approach allowed Nexmo to integrate testRTC right into their current testing environment and test scripts.

CATERING FOR TEAMS:

The Voice API engineering team is a large one. All these users have access to testRTC and they are able to launch regression tests that end up running testRTC scripts as well as using the testRTC dashboard to debug issues that are found.

The ability to have multiple users, each with their own credentials, running tests on demand when needed enabled increased productivity without dealing with coordination issues across the team members. The test results themselves get hosted on a single repository, accessible to the whole team, so all developers can easily share faulty test results with the team.

DEBUGGING WEBRTC ISSUES:

Nexmo has got regression testing for WebRTC off the ground by using testRTC. It does so by integrating with the testRTC APIs, scheduling and launching tests on demand from Nexmo's own test environment. The tests today are geared towards providing end-to-end validation of media and connectivity between the PSTN network and WebRTC. Validation that testRTC takes care of by default.

Any result	Any test	Any date	Search history by name/machine		Q	
Manual	Name	Machines	Iterations	Probes	Time	Reason
✓	test_media_is_exchanged_between_pstn_and_ip	flaky-pipe	0	1	Aug 18 2018 - 23:04	
✓	test_media_is_exchanged_between_pstn_and_ip	better-worm	0	1	Aug 18 2018 - 23:03	
✓	test_media_is_exchanged_between_ip_and_pstn	amused-daughter	0	1	Aug 18 2018 - 23:03	
✓	test_media_is_exchanged_between_pstn_and_ip	taboo-kitty	0	1	Aug 17 2018 - 23:04	
✓	test_media_is_exchanged_between_pstn_and_ip	low-flame	0	1	Aug 17 2018 - 23:03	
✓	test_media_is_exchanged_between_ip_and_pstn	easy-collar	0	1	Aug 17 2018 - 23:03	
✓	test_media_is_exchanged_between_pstn_and_ip	abundant-profit	0	1	Aug 17 2018 - 19:36	
✓	test_media_is_exchanged_between_pstn_and_ip	separate-offer	0	1	Aug 17 2018 - 19:36	
✓	test_media_is_exchanged_between_ip_and_pstn	thundering-slave	0	1	Aug 17 2018 - 19:36	
✓	test_vapi_rest_calls_put_transfer_ip_to_simple_conversation	dead-instrument	0	1	Aug 17 2018 - 18:39	
✓	test_vapi_rest_calls_put_transfer_ip_to_simple_conversation	cute-parcel	0	1	Aug 17 2018 - 18:06	

When things break, developers check the results collected by testRTC. As Jamie Chapman, Voice API engineer at Nexmo said: "When it comes to debugging, when something has gone wrong, testRTC is the first place we'd go look. There's a lot of information there".

testRTC takes screenshots during the test run, as well as upon failure. It collects browser logs and [webrtc-internals](#) dump files, visualizing it all and making it available for debugging purposes. This makes testRTC a valuable tool in the development process at Nexmo.

ON THE HORIZON:

Nexmo is currently making use of the basic scripting capabilities of testRTC. It has invested in the API integration, but there is more that can be done.

Nexmo are planning to increase their use of testRTC in several ways in the near future:

- Use testRTC's [media thresholding](#) capabilities as well as its [network configuration](#)
- Running active monitoring on the production network and service, covering multiple geographic locations