Integration with NeoLoad

- Overview
- Main features
  - NeoLoad quick overview and core concepts
  - Mapping of concepts
- Setup
- Example
- Tips
- Learn more

Overview

NeoLoad is an enterprise tool for performance and load testing, covering both API and end-to-end application testing.

Besides support for web-based applications, it also supports a broader set of applications and protocols, including SAP, Oracle Forms, Adobe Flex, etc.

Users may define SLA's based on performance metrics, giving them insights about performance that can be crucial for business.

Performance tests can be triggered manually (e.g. from within NeoLoad GUI or NeoLoad web) or from a CI/CD pipeline.

Main features

This integration provides the ability to track performance & load testing results in Xray, thus making it the single source of truth for tracking different testing efforts.

It provides:

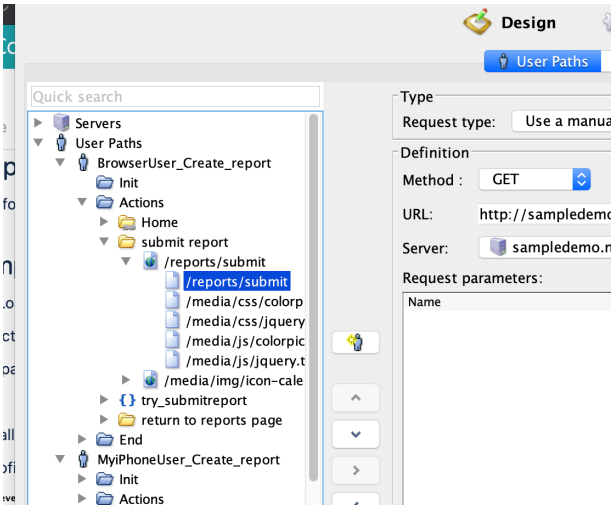
- integration with both Xray on Jira server/DC or cloud
- chart with an overview of the testing session
- global statistics of the test
- link to NeoLoad web, for more in-depth results about the testing session
- automatic provisioning of the test scenario
- overall test result and per SLA result
- from NeoLoad side, the ability to:
  - identify the project, its version and the revision of the SUTI
  - link scenario to an existing "requirement" in Jira/Xray
  - add labels to Test entity that is used to abstract the scenario
  - identify the target environment (i.e. Test Environment)
  - link results back to an existing Test Plan in Xray

NeoLoad quick overview and core concepts

In NeoLoad, all happens in the context of a working project, which may be created/modified using NeoLoad GUI.

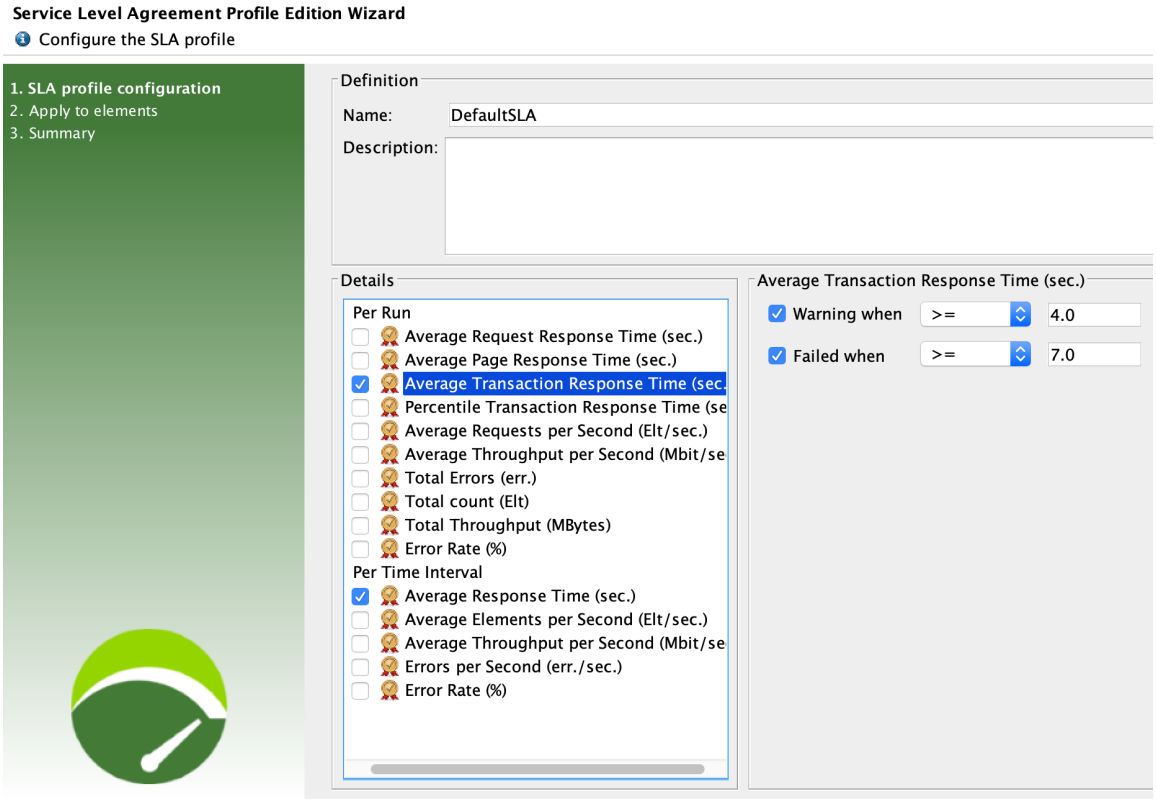
A project contains one or more (test) scenarios. Each scenario has a set of user paths (i.e. flows) to be exercised along with the running conditions (e.g. duration, load policy, VUs).

A user path corresponds to the definition of a browsing sequence in the target application; it contains, for example, the HTTP requests to be performed.



Optionally, users can define one or more SLA (Service Level Agreement) profiles as a means to evaluate if the application meets scaling/performance needs.

SLA profiles are defined as a combination of one or more SLAs; in turn, a given individual SLA is based on a metric and some target/reference value.

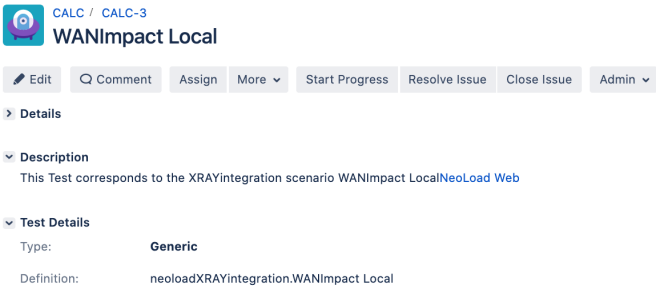


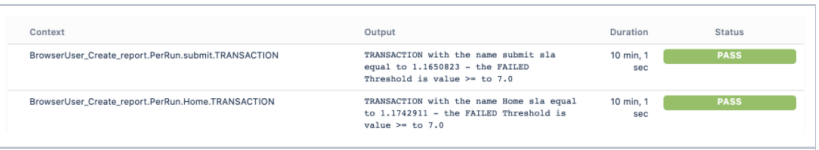
SLA profiles can be applied at the whole user path level or at individual elements within the user path flow.

From a testing perspective, each time that you apply an SLA to a certain context (e.g. user path, request) you are defining a goal to that context.

This way, at the end of your test session you may evaluate if those goals were met or not (individually and as a whole).

Mapping of concepts

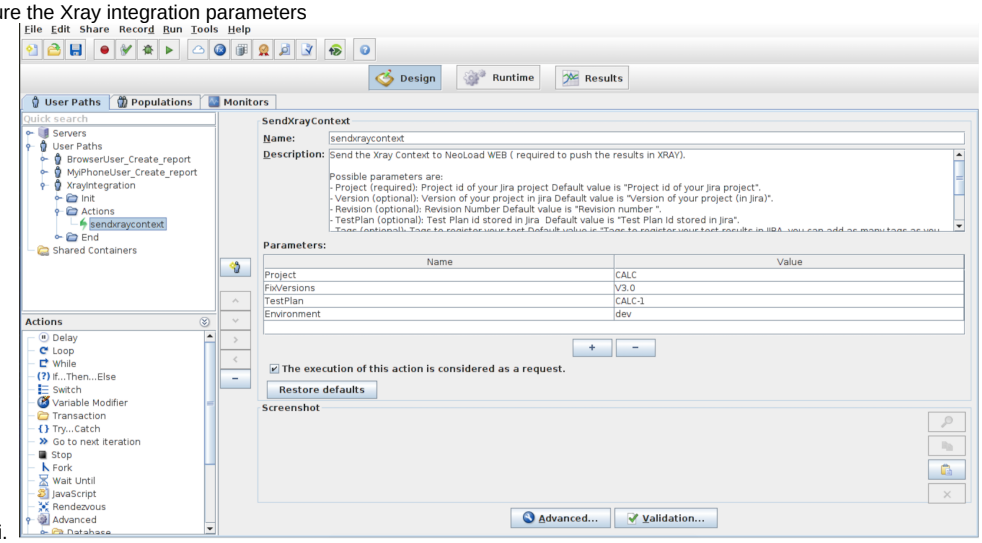
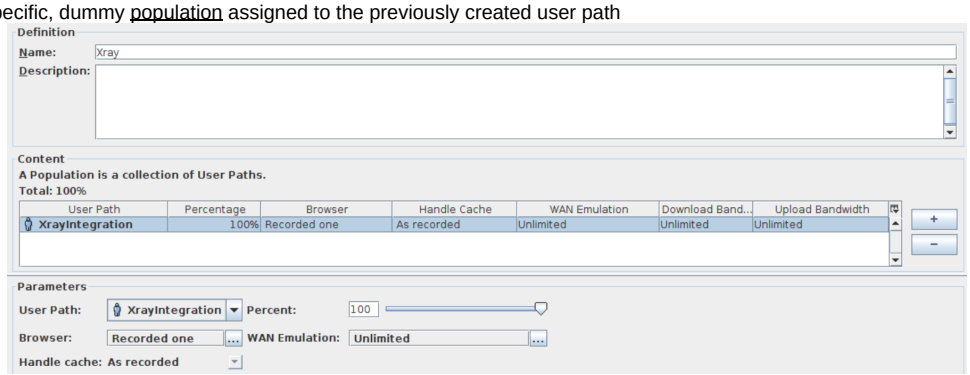
NeoLoad	Xray	Observations/examples
(test) scenario	Test (generic/unstructured) • it may be linked to one, or more, requirements (e.g performance requirement) Note: this Test is usually auto-provisioned the first time whenever importing test results and reused afterwards	
(scenario) test results	Test Run within a Test Execution • the Test Execution corresponds to the whole performance testing session • a Test Run contains the details for a specific scenario	

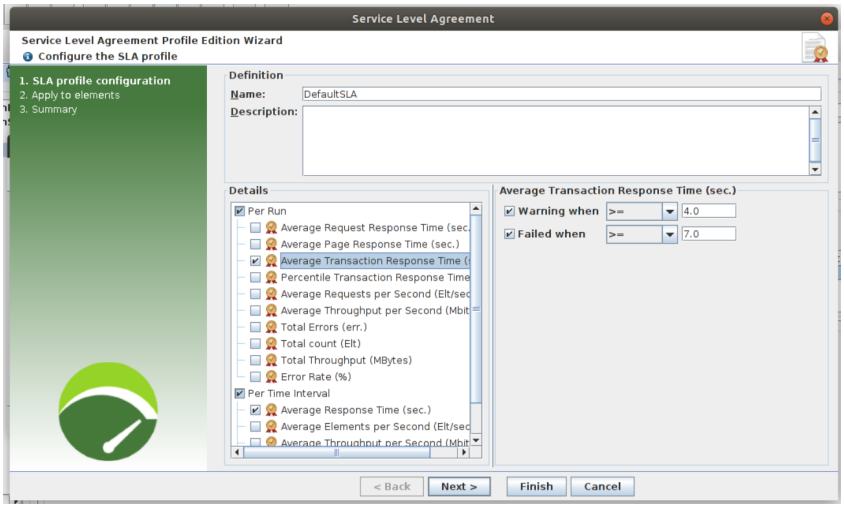
SLA's	multiple entries in the Test Run details	
status of test results	status of the Test Run	based on the status of each SLA

Setup

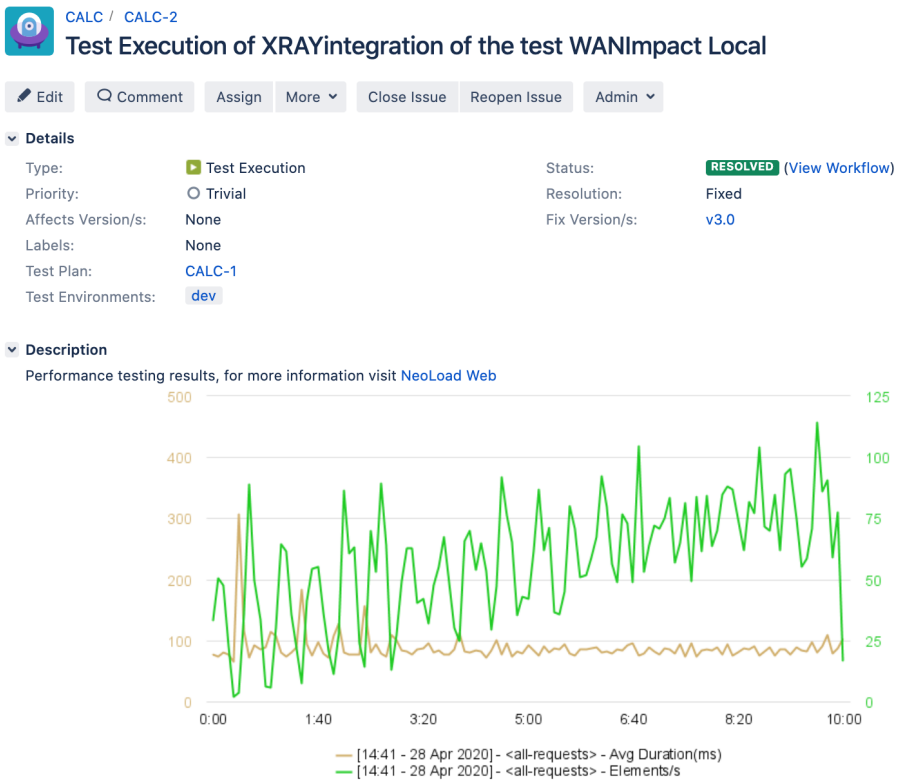
Please follow the instructions detailed by NeoLoad's team here.

Example

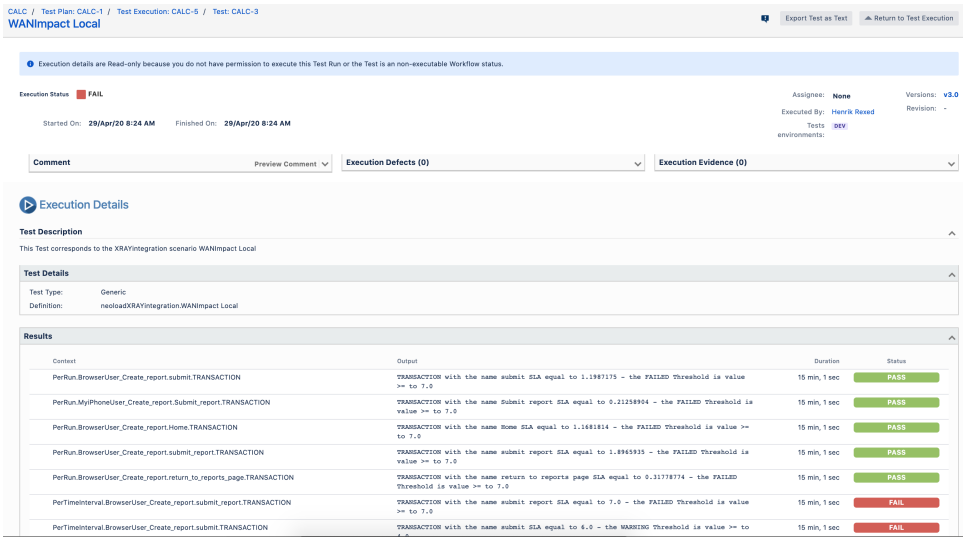
1. Setup
    a. use a specific, dummy user path for configuration the integration with Xray by including the "SendXrayContext" action in the "Actions" section
    b. configure the Xray integration parameters
        i. 
    c. use a specific, dummy population assigned to the previously created user path
        i. 
2. Create your test scenario (e.g. user paths) in NeoLoad
    a. Whenever importing the results, a Test issue will usually be auto-provisioned in Xray; the name of the scenario will be mapped to the Test' summary and will be used as the unique identifier for the Test (as seen on its Definition field) so it can be reused on subsequent result imports.
        i. Note: It is also possible to override the auto-provision behavior and enforce results to be reported against an existing Test issue that would need be created beforehand; in this case, the Test issue key has to be specified as one of values in the "Tags" parameter of the SendXrayContent action.
3. Define and apply some SLAs

a. 

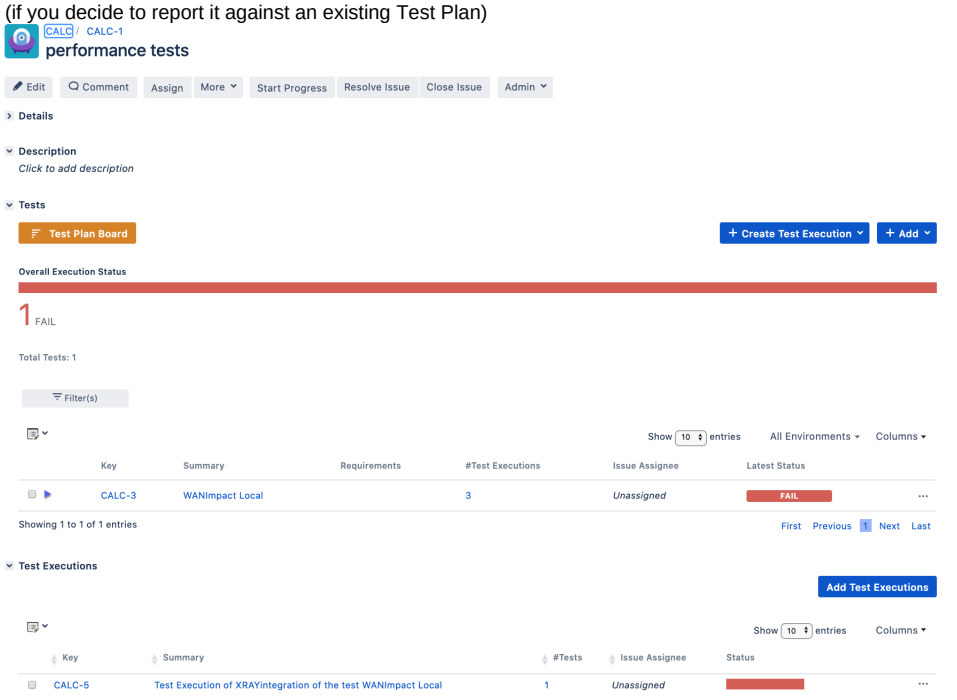
4. Define the runtime parameters for your scenario (e.g. duration, load policy, VUs)
5. Execute it
6. See results in Xray
    a. Test Execution



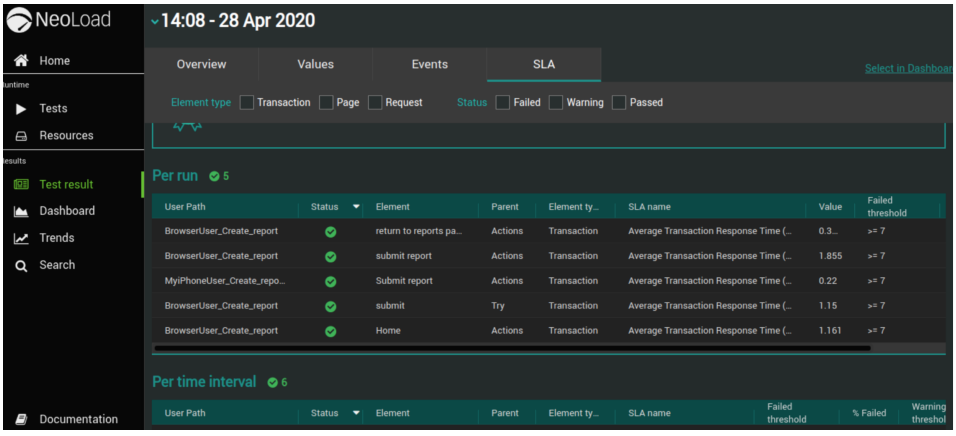
        i.
    b. Test Run

i. 

c. Test Plan (if you decide to report it against an existing Test Plan)

i. 

7. Check in-depth details in NeoLoad Web
   a. From within the description field of the Test Execution issue, you have a link to the test results in NeoLoad Web where you can see finer details

i. 

Tips

- you may link your scenario to existent requirements in Jira and thus see the impact on the coverage calculated for them. You can link it as usual, by creating a "tests" link between the requirement and the Test issue or you can do it automatically whenever importing the results back to Xray. In the later, you need to add the issue key of the requirement as one value in the "Tags" parameter of the "SendXrayContext" action
- use a Test Plan for tracking performance-related results; report against an existent Test Plan by providing its key in the initial configuration of the "SendXrayContext" action

Learn more

For specific questions concerning NeoLoad, please use Neotsys support resources.

Please refer to the following resources for more information:

- NeoLoad web site
- NeoLoad integration with Xray
- NeoLoad documentation