

Asynchronous requests

In some applications, there may be asynchronous requests that occur in the background. For example, consider an application where a request performs a long server operation, and during the operation, the browser polls the server for progress.

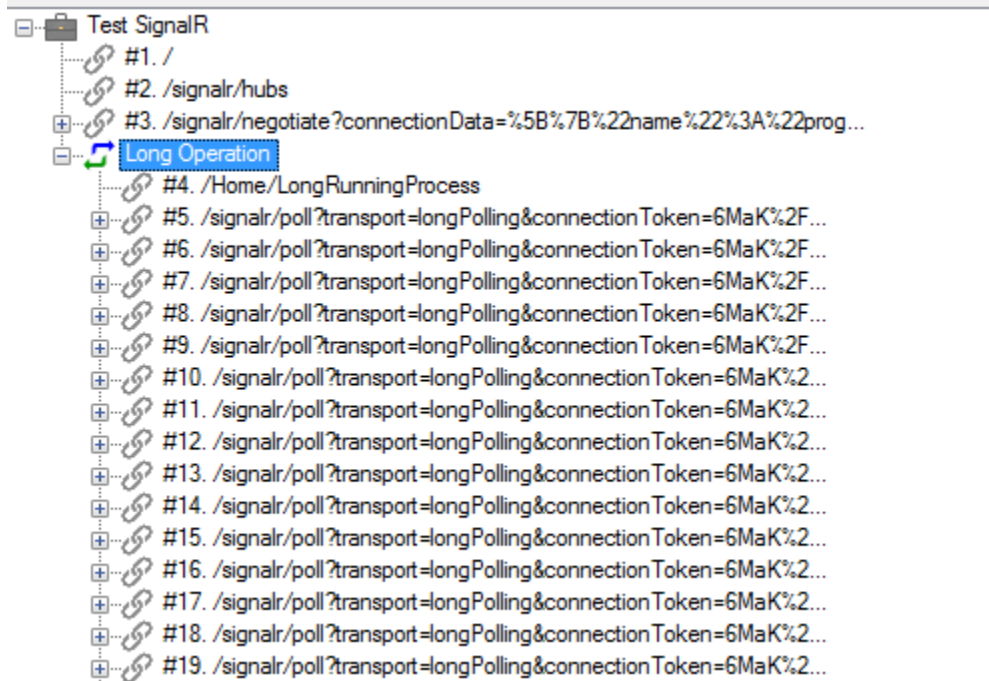
To load test this scenario, StresStimulus needs to execute the work request and the series of polling requests at the same time (asynchronously).

Let's go through an example of setting up such a test and explaining its playback.

Recording & configuring the test case

After recording a test case, the test case tree looks as follows.

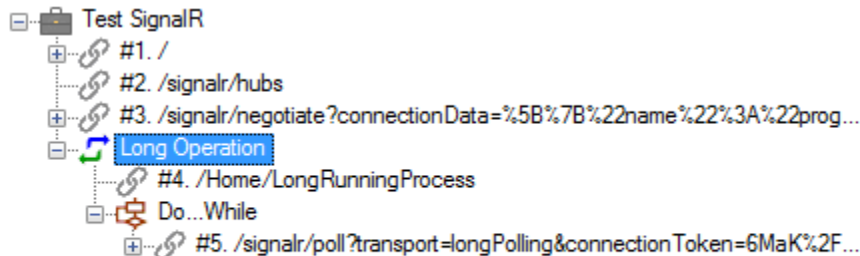
- There is a long work request /Home/LongRunningProcess
- There are many polling requests /signalr/poll



Test case objects

Since the server must be polled until the operation is completed, the polling request must be executed in a **Do...While** loop until the server responds back with a "Done" response.

The while loop should be created around a single polling request, and all the other requests deleted.



Asynchronous requests

The Do...While loop (containing request #5) will not begin until the response for /Home/LongRunningProcess (request #4) comes back. Therefore the requests in the Long Operation transaction (#4 and #5) must set their concurrency mode to **Asynchronous**.

Object type	Session
Timeout (s)	60
Caching Rules	Not Cached
Ignore warnings/errors?	No
Disabled	No
Concurrency mode	Default
Async Delay (s)	Default
	Parallel
	Sequential
	Asynchronous

When a request is asynchronous, StresStimulus will not wait for its response in order to continue test case execution. Therefore, immediately after request #4 is sent, the polling do...while loop will begin.

Polling must be done at a certain predefined interval, meaning there needs to be a delay between sending the polling request. Since it is an asynchronous request, set the Async Delay property to the desired polling interval.

Unlike think times or delay object, the async delay property delay is included in calculation of parent transaction times.