|  |  |
| --- | --- |
|  | Meteor page is used for the developer to automate tests, to populate a process, to advance a case to a specific task, to verify that you cover all the process  **When to use it?**  In development, to advance a case to your task.  In a Load test  To populate a process for a demonstration  To automate the test execution |

Build the test scenario

Meteor run on the customer process, not on a theory process.

Diverse ways exist to build scenario:

From the analysis. Meteor check on the process all existing process and human task. For each process / each task, you can setup one or multiple robot (to simulate multiple human working on a human task for example). Then you can set up multiple data set, in order to cover all the use case:

* “in 35%, set the AmountInput to 3454
* in 40%, set the AmountInput to 8930
* in 25%, set the AmountInput to 9530

From a Command scenario. Give the operation as a list of command like “Create Case / Execute Task”. Run this command scenario multiple time, in parallel by multiple robots

From a Groovy Scenario. Use the Groovy to build a very complex scenario. The GroovyLib test is included, to help you to define the scenario. The API is available too.

Development

In development, you want to develop and test all your form. You change your process, and you want to check again the task “Deliver Card”. But to arrive to this task, you must execute before 6 another task.

Build a simple scenario to create a case and execute automatically the 6 tasks. After each deployment, click on “RUN”.

Load test

In a Validation platform, you have your different process. Connector will do the work to fetch data. How to verify this platform will be enough to manage 8000 cases a day? and can manage 12000 cases a day if needed? Where will be the bottleneck? A connector? The engine itself?

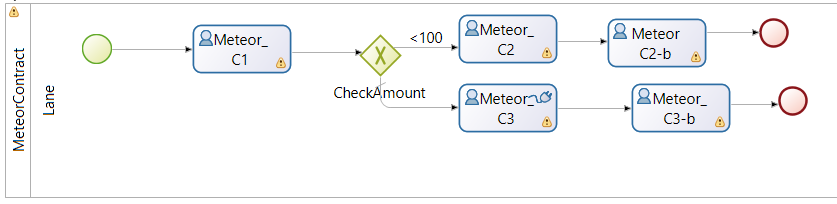
Build a scenario with Meteor, simulate 200 users executing a task every 5 minutes, and then run the test. During the execution, you can follow the average time per execution, to detect if the running time is stable, or if suddenly, after 4200 cases, the platform slow down.

On going test

You want to run every day some tests, to verify if the result is the same and if the performance is stable day after day. Build multiple scenario, and load them on the platform. Every day, via a REST API call, execute the test, and get a detailed status. Is the status the one expected in term of execution and time?

How it’s works?

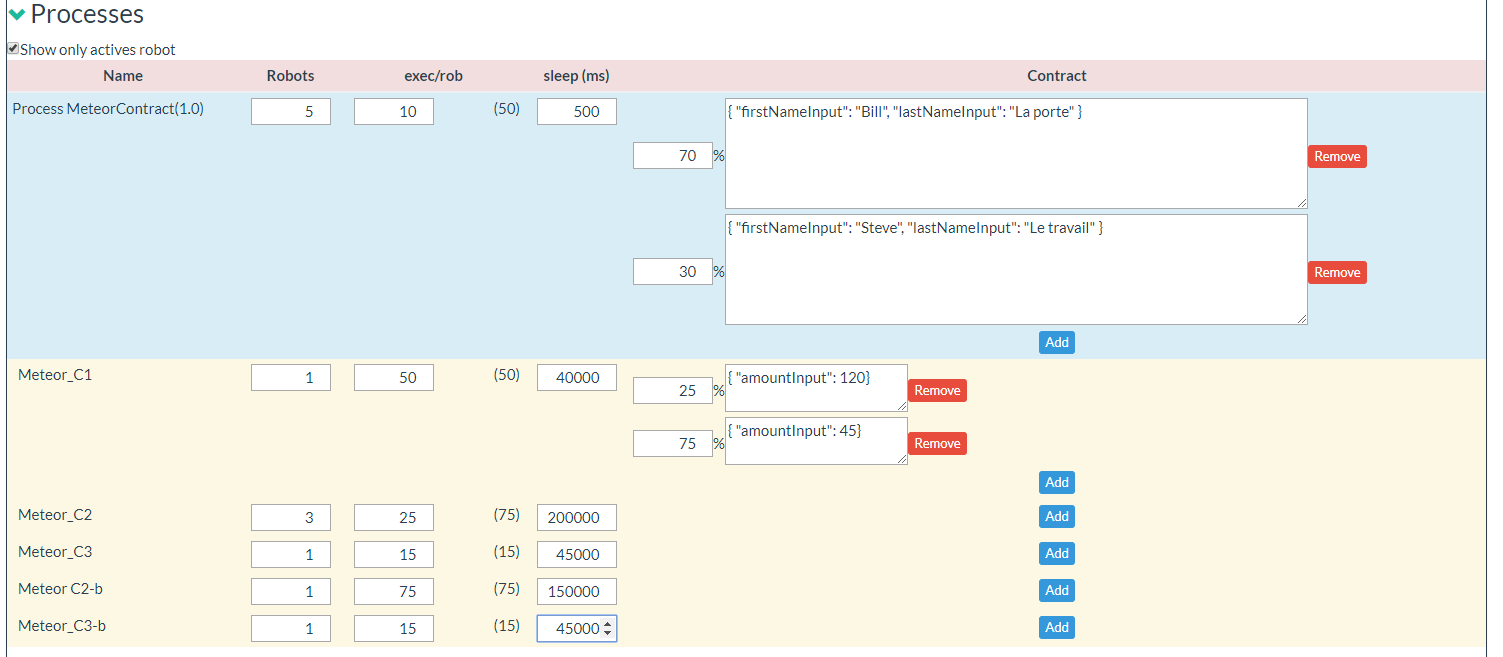
Meteor is based on a real process of a customer



By the process scenario

Meteor can detect this process, and all the human tasks. A scenario can be build, using percentage to use all the different path. The idea is to test a real use case.

So, we decided to create 500 cases, simulated by 5 different users. Each user is supposed to create a case every 500 seconds (which is realistic if users are external software). We used two different data set

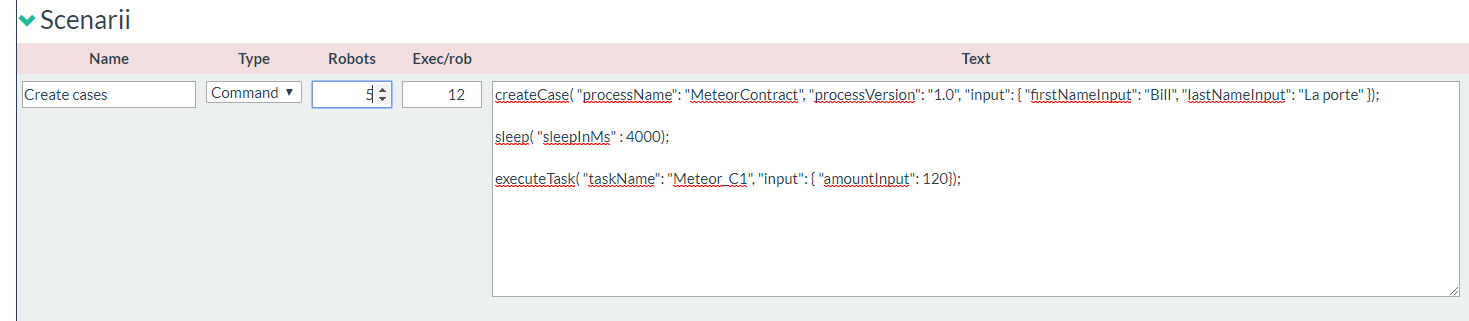


On the task 1, a user task, 1 users will woks and manage the 50 cases. These users wait 40 s between each execution. In 25 %, he will set the value 120. In 70%, it will set the value 45.

Same for the different tasks of the process.

Use the Command Scenario

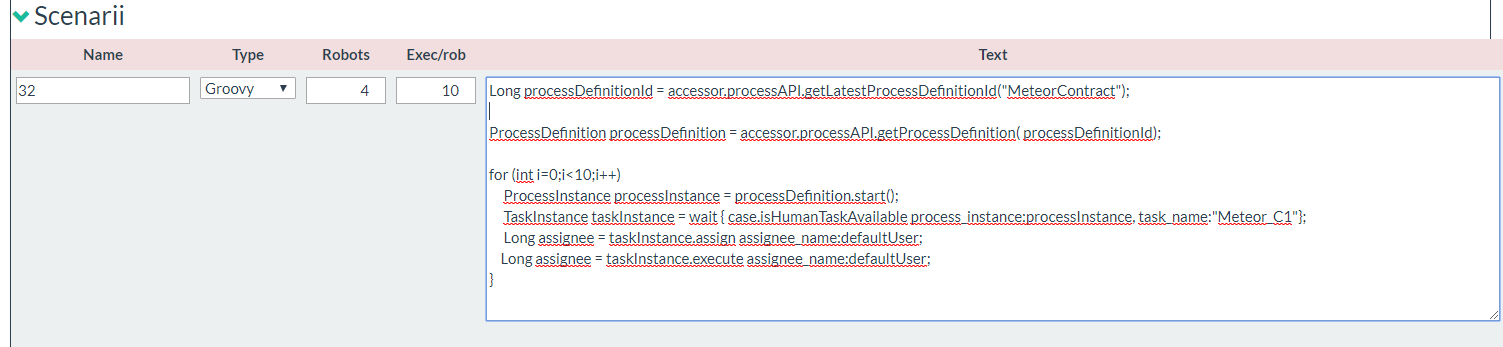
A command scenario is a list of command. It can be run in paralleled, multiple time



Use a Groovy scenario

The scenario is developed in Groovy. The special object “accessor” is used to access ProcesssAPI, IdentityAPI or any another element.

The GroovyScenario 2.0 is included, and new order are available like “wait” to wait a task name.



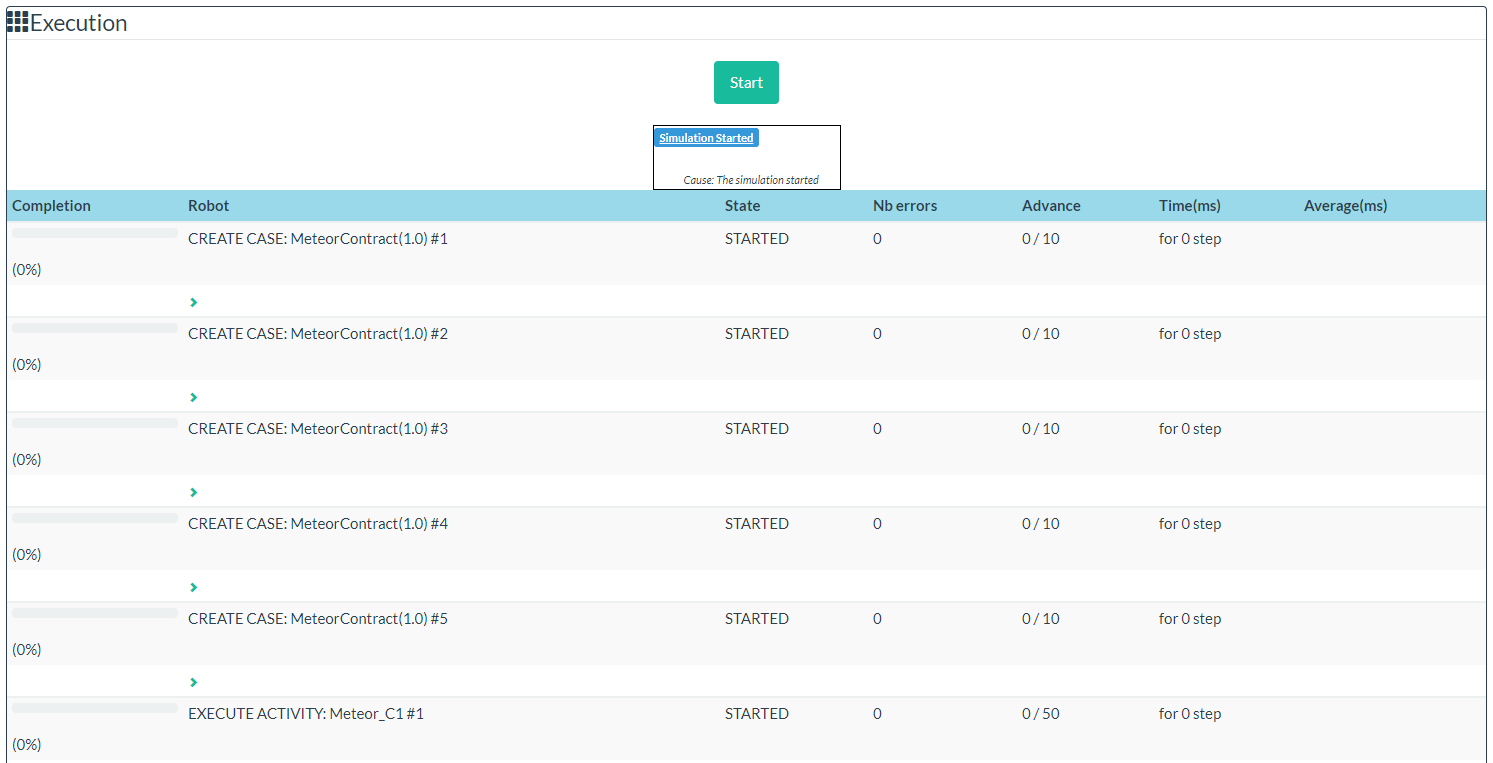
Save it, export it, import it

Note that you can save the test, and export it to be able to replay it, or top replay it on another platform

Run it!

A run is immediate when the RUN button is clicked, but the run can be started by a REST API too. Each Scenario can be start by a REST API.

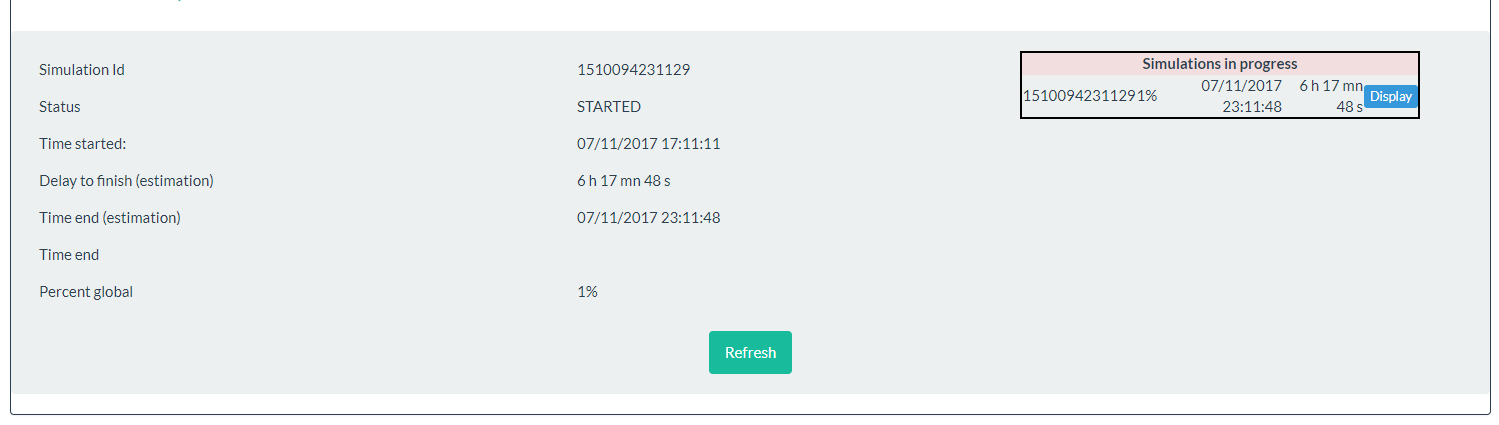
Let’s start the scenario:



Then a real status on line is given, and an advancement for each robot is given



The monitoring is used to give the time to the end of the execution. All different test can be accessible in the list



HEADQUARTERS

PARIS, FRANCE

76 boulevard de la République

92100 Boulogne-Billancourt

EMEA, ASIA & LATIN AMERICA

GRENOBLE, FRANCE

32, rue Gustave Eiffel  
38000 Grenoble

NORTH AMERICA

SAN FRANCISCO, USA

44 Tehama Street  
San Francisco, CA 94105

NEW YORK, USA

33 Nassau Avenue

Brooklyn NY 11222