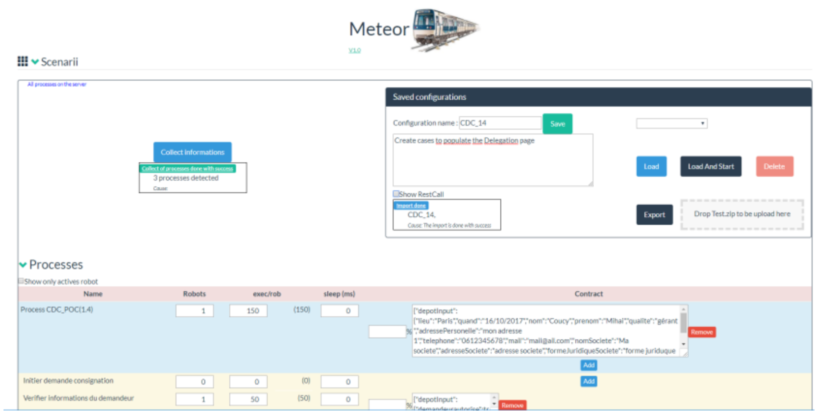


Collect of processes done with success
3 processes detected
Cause:

Configuration name : CDC_14

Create cases to populate the Delegation page

Show RestCall

Import done
CDC_14,
Cause: The import is done with success

Processes

Show only actives robot

Name	Robots	exec/rob		sleep (ms)	Contract
Process CDC_POC(1.4)	1	150	(150)	0	{"depotInput": {"lieu":"Paris","quand":"16/10/2017","nom":"Coucy","prenom":"Mihai","qualite":"gérant","adressePersonelle":"mon adresse 1","telephone":"0612345678","mail":"mail@ail.com","nomSociete":"Ma societe","adresseSociete":"adresse societe","formeJuridiqueSociete":"forme juriduque
Initier demande consignation	0	0	(0)	0	
Verifier informations du demandeur	1	50	(50)	0	{"depotInput": {"demandeurautorise":tr

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?

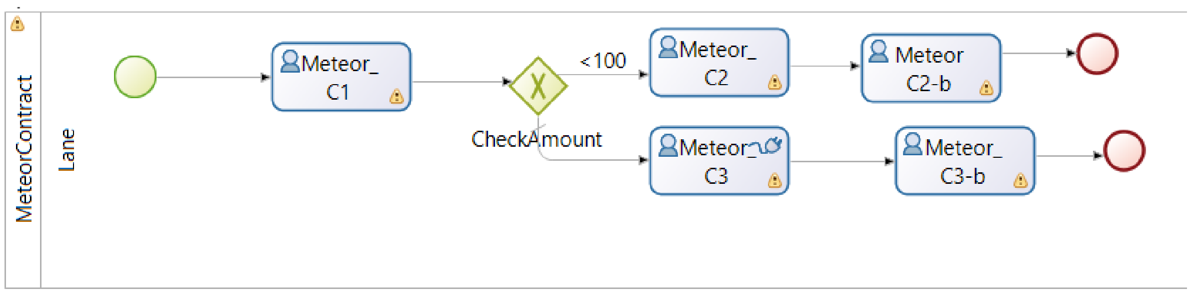
Build a scenario with Meteor, simulate 200 users executing a task every 5 minutes, and then run the test. During the execution, you can follow the average time per execution, to detect if the running time is stable, or if suddenly, after 4200 cases, the platform slow down.

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?

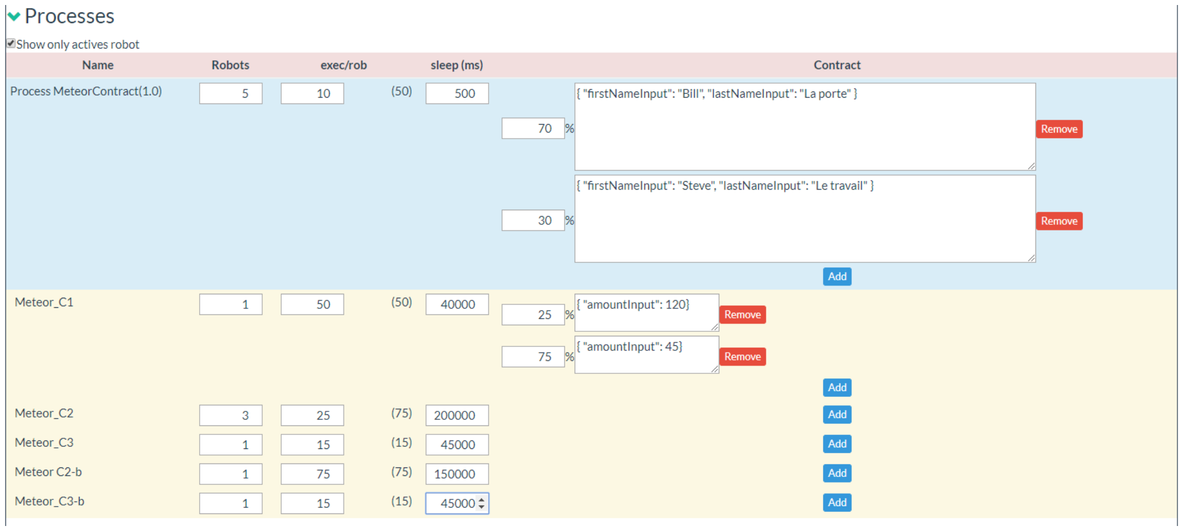
Meteor is based on a real process of a customer



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



Processes

☑ Show only actives robot

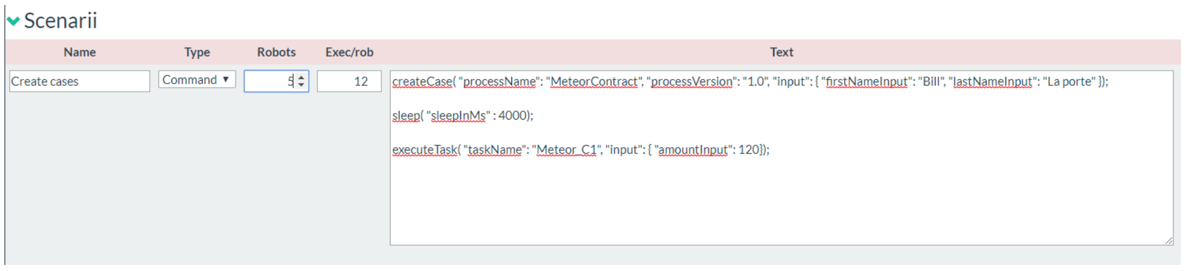
Name	Robots	exec/rob		sleep (ms)		Contract	
Process MeteorContract(1.0)	5	10	(50)	500	70 %	{ "firstNameInput": "Bill", "lastNameInput": "La porte" }	Remove
					30 %	{ "firstNameInput": "Steve", "lastNameInput": "Le travail" }	Remove
						Add	
Meteor_C1	1	50	(50)	40000	25 %	{ "amountInput": 120}	Remove
					75 %	{ "amountInput": 45}	Remove
						Add	
Meteor_C2	3	25	(75)	200000		Add	
Meteor_C3	1	15	(15)	45000		Add	
Meteor C2-b	1	75	(75)	150000		Add	
Meteor_C3-b	1	15	(15)	45000		Add	

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



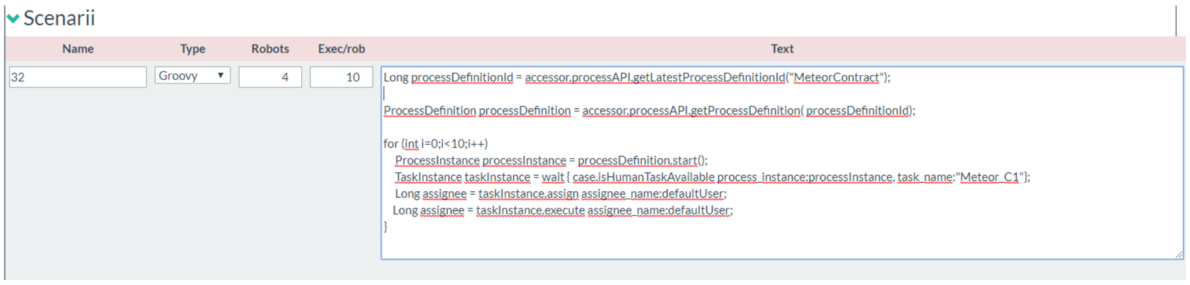
Scenarii

Name	Type	Robots	Exec/rob	Text
Create cases	Command ▾	5	12	createCase("processName": "MeteorContract", "processVersion": "1.0", "input": { "firstNameInput": "Bill", "lastNameInput": "La porte" }); sleep("sleepInMs" : 4000); executeTask("taskName": "Meteor_C1", "input": { "amountInput": 120});

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.


Scenarii

Name	Type	Robots	Exec/rob	Text
32	Groovy	4	10	see below

```
Long processDefinitionId = accessor.processAPI.getLatestProcessDefinitionId("MeteorContract");

ProcessDefinition processDefinition = accessor.processAPI.getProcessDefinition( processDefinitionId);

for (int i=0;i<10;i++)
   ProcessInstance processInstance = processDefinition.start();
   TaskInstance taskInstance = wait { case.isHumanTaskAvailable process_instance:processInstance, task_name:"Meteor_C1"};
   Long assignee = taskInstance.assign assignee_name:defaultUser;
  Long assignee = taskInstance.execute assignee_name:defaultUser;
}
```

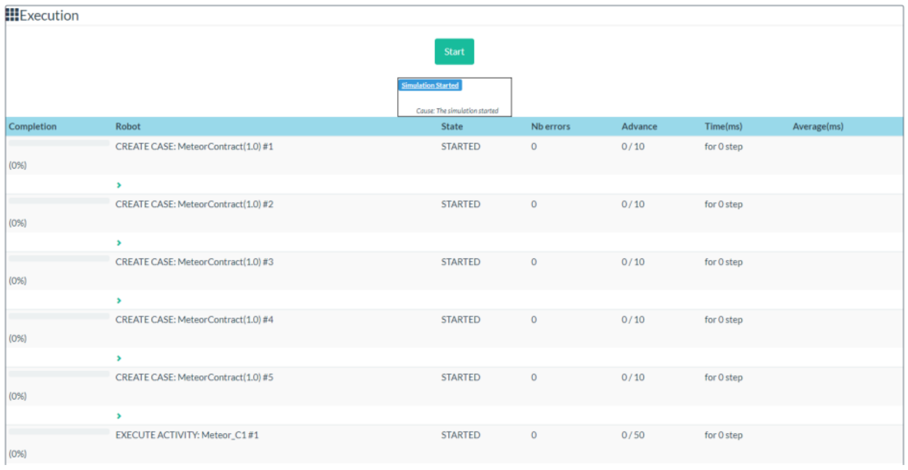
Save it, export it, import it

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:

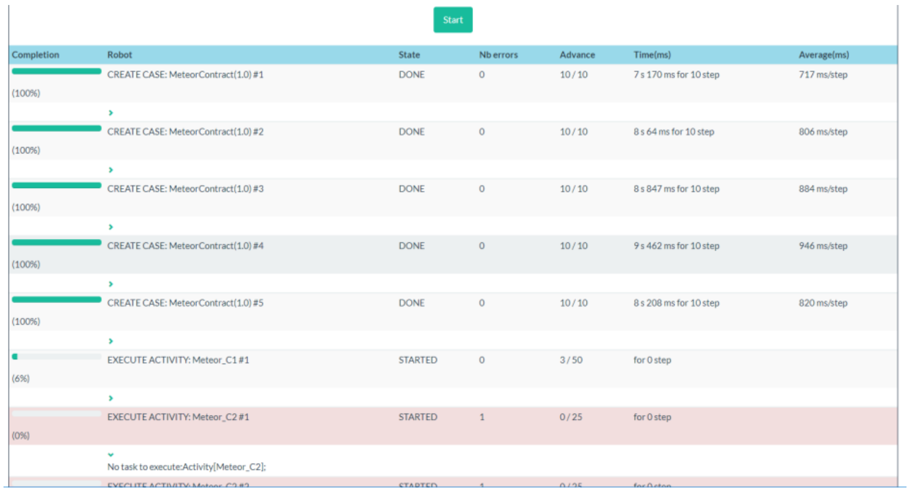


Simulation Started

Cause: The simulation started

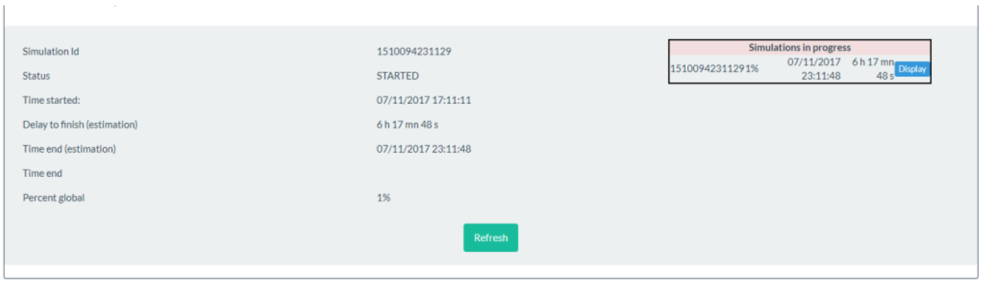
Completion	Robot	State	Nb errors	Advance	Time(ms)	Average(ms)
(0%)	CREATE CASE: MeteorContract(1.0) #1	STARTED	0	0 / 10	for 0 step	
(0%)	CREATE CASE: MeteorContract(1.0) #2	STARTED	0	0 / 10	for 0 step	
(0%)	CREATE CASE: MeteorContract(1.0) #3	STARTED	0	0 / 10	for 0 step	
(0%)	CREATE CASE: MeteorContract(1.0) #4	STARTED	0	0 / 10	for 0 step	
(0%)	CREATE CASE: MeteorContract(1.0) #5	STARTED	0	0 / 10	for 0 step	
(0%)	EXECUTE ACTIVITY: Meteor_C1 #1	STARTED	0	0 / 50	for 0 step	

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



Completion	Robot	State	Nb errors	Advance	Time(ms)	Average(ms)
(100%)	CREATE CASE: MeteorContract(1.0) #1	DONE	0	10 / 10	7 s 170 ms for 10 step	717 ms/step
(100%)	CREATE CASE: MeteorContract(1.0) #2	DONE	0	10 / 10	8 s 64 ms for 10 step	806 ms/step
(100%)	CREATE CASE: MeteorContract(1.0) #3	DONE	0	10 / 10	8 s 847 ms for 10 step	884 ms/step
(100%)	CREATE CASE: MeteorContract(1.0) #4	DONE	0	10 / 10	9 s 462 ms for 10 step	946 ms/step
(100%)	CREATE CASE: MeteorContract(1.0) #5	DONE	0	10 / 10	8 s 208 ms for 10 step	820 ms/step
(6%)	EXECUTE ACTIVITY: Meteor_C1 #1	STARTED	0	3 / 50	for 0 step	
(0%)	EXECUTE ACTIVITY: Meteor_C2 #1	STARTED	1	0 / 25	for 0 step	
	No task to execute:Activity[Meteor_C2];					
	EXECUTE ACTIVITY: Meteor_C2 #2	STARTED	1	0 / 25	for 0 step	

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



Simulation Id	1510094231129
Status	STARTED
Time started:	07/11/2017 17:11:11
Delay to finish (estimation)	6 h 17 mn 48 s
Time end (estimation)	07/11/2017 23:11:48
Time end	
Percent global	1%

Simulations in progress

1510094231129	1%	07/11/2017 23:11:48	6 h 17 mn 48 s	Display

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

Bonitasoft

www.bonitasoft.com