DREAM

Developing with DREAM Platform

Nexedi 2015-03-26

FoF.NMP.2012-6
Collaborative Project
Grant Agreement: 314364

SEVENTH FRAMEWORK PROGRAMME

nexedi

Agenda

1. Install DREAM Runner
2. Edit DREAM platform source code
3. Run DREAM platform unit tests
4. Accessing DREAM GUI
5. Introduction to JSON format
6. Configuring Graph Editor
7. Configuring Application configuration, adding a spreadsheet tab and a new gadget
8. Configuring Knowledge extraction plugins
9. Cloud Execution of ACO

Install DREAM Runner

Easy way: use vifib hosting

1. Create an account on https://www.slapos.org/

Install DREAM Runner

Easy way: use vifib.com hosting

1. Create an account on https://www.slapos.org/
2. Request New Service

Install DREAM Runner

DREAM

Easy way: use vifib.com hosting

1. Create an account on https://www.slapos.org/
2. Request New Service
3. Choose “SlapOS web runner”

SEVENTH FRAMEWORK PROGRAMME

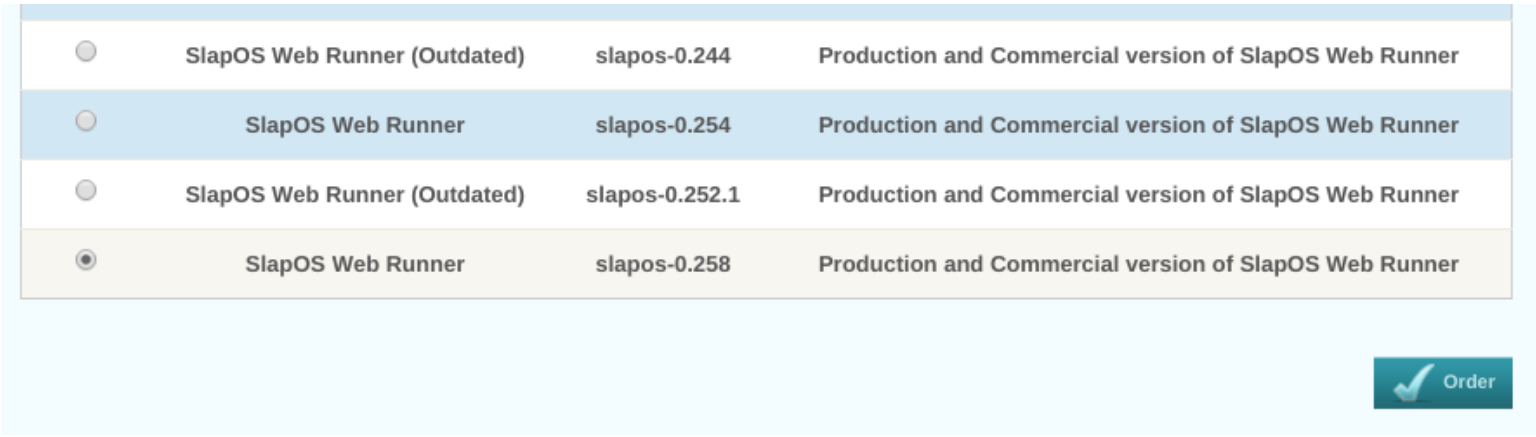
nexedi

Install DREAM Runner

DREAM

Easy way: use vifib.com hosting

1. Create an account on https://www.slapos.org/
2. Request New Service
3. Choose “SlapOS web runner”
4. Choose latest version



SEVENTH FRAMEWORK PROGRAMME

nexedi

Install DREAM Runner

Easy way: use vifib.com hosting

1. Create an account on https://www.slapos.org/
2. Request New Service
3. Choose “SlapOS web runner”
4. Choose latest version
5. Set Service Title and XML Parameter

Install DREAM Runner

Easy way: use vifib.com hosting

1. Create an account on https://www.slapos.org/
2. Request New Service
3. Choose “SlapOS web runner”
4. Choose latest version
5. Set Service Title and XML Parameter
   - slapos-software: the software to use, available in slapos git repository, see http://git.erp5.org/gitweb/slapos.git/tree/HEAD:/software?js=1
   - slapos-reference: The git branch from slapos repository
   - auto-deploy: Auto deploy the software
   - user-authorized-keys: Your ssh public key, if you want to use ssh
   - custom-frontend-backend-url: This will generate an IPv4 frontend for you if you cannot access it through IPv6.

Install DREAM Runner

DREAM

Easy way: use vifib.com hosting

1. Create an account on https://www.slapos.org/
2. Request New Service
3. Choose “SlapOS web runner”
4. Choose latest version
5. Set Service Title and XML Parameter
6. Wait for a while and connection parameter will appear

SEVENTH FRAMEWORK PROGRAMME

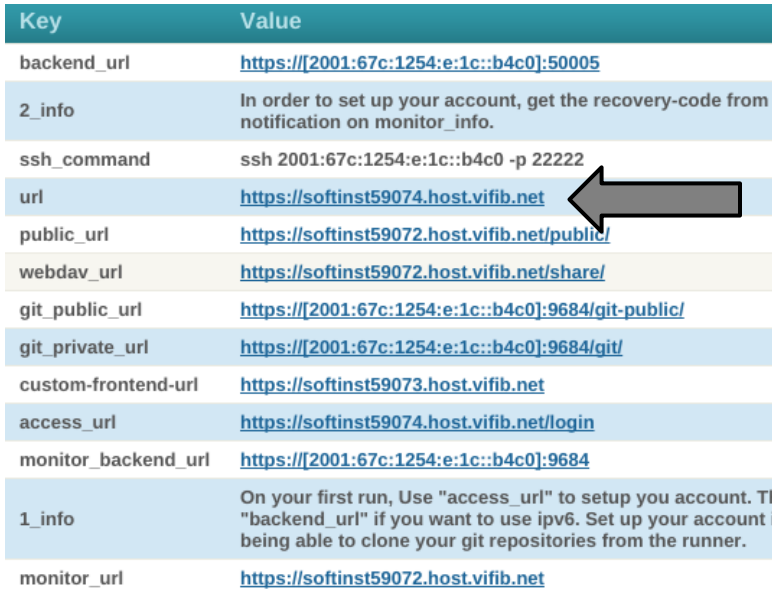
nexedi

Install DREAM Runner

DREAM

Easy way: use vifib.com hosting

1. Create an account on https://www.slapos.org/
2. Request New Service
3. Choose “SlapOS web runner”
4. Choose latest version
5. Set Service Title and XML Parameter
6. Wait for a while and connection parameter will appear
7. Click on ***URL*** parameter



Key	Value
backend_url	https://[2001:67c:1254:e:1c::b4c0]:50005
2_info	In order to set up your account, get the recovery-code from notification on monitor_info.
ssh_command	ssh 2001:67c:1254:e:1c::b4c0 -p 22222
url	https://softinst59074.host.vifib.net
public_url	https://softinst59072.host.vifib.net/public/
webdav_url	https://softinst59072.host.vifib.net/share/
git_public_url	https://[2001:67c:1254:e:1c::b4c0]:9684/git-public/
git_private_url	https://[2001:67c:1254:e:1c::b4c0]:9684/git/
custom-frontend-url	https://softinst59073.host.vifib.net
access_url	https://softinst59074.host.vifib.net/login
monitor_backend_url	https://[2001:67c:1254:e:1c::b4c0]:9684
1_info	On your first run, Use "access_url" to setup you account. Th "backend_url" if you want to use ipv6. Set up your account being able to clone your git repositories from the runner.
monitor_url	https://softinst59072.host.vifib.net

SEVENTH FRAMEWORK PROGRAMME

nexedi

Install DREAM Runner

Easy way: use vifib.com hosting

1. Create an account on https://www.slapos.org/
2. Request New Service
3. Choose “SlapOS web runner”
4. Choose latest version
5. Set Service Title and XML Parameter
6. Wait for a while and connection parameter will appear
7. Click on URL parameter
8. ***Pay invoices every month***

Install DREAM Runner

… or Install SlapOS in your IT infrastructure.

Follow the tutorials from:

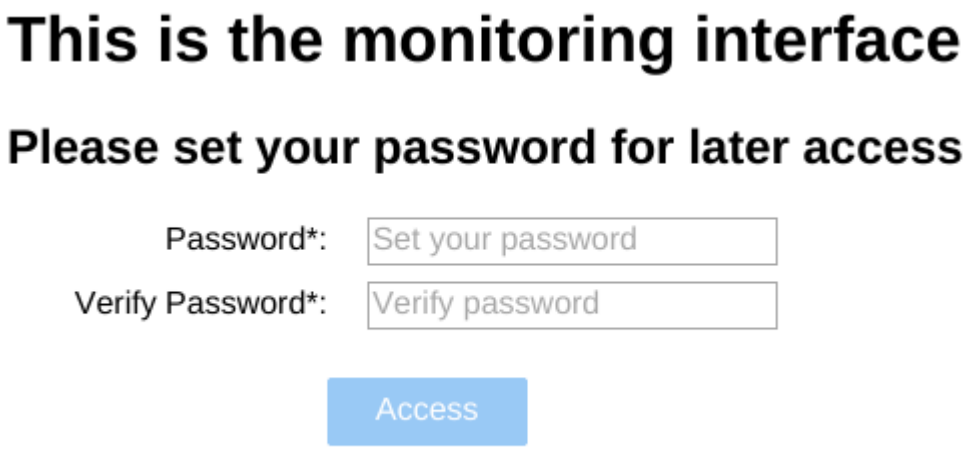
http://community.slapos.org/wiki/developer-Allocate.SlapOS.Master.Instance

http://community.slapos.org/wiki/developer-Installing.SlapOS.Package

Install DREAM Runner

Before start, configure monitoring from “Monitoring” link on the slapos page.

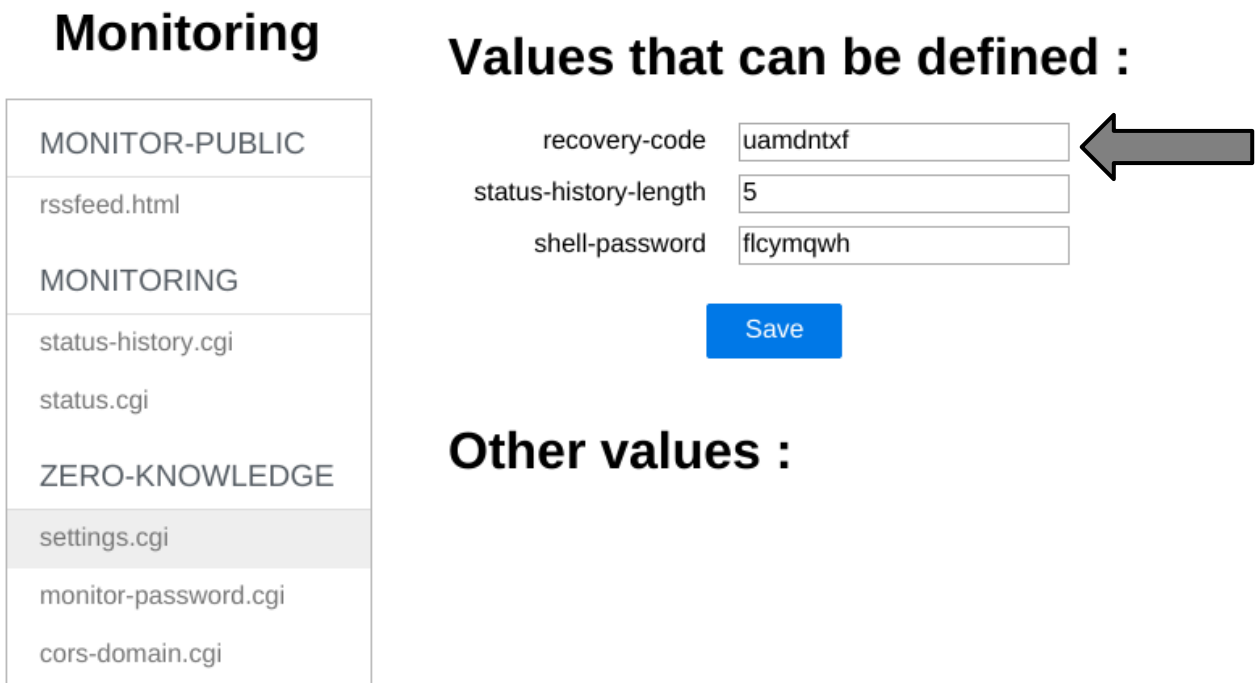
1. Set monitoring password



Install DREAM Runner

Before start, configure monitoring from “Monitoring” link on the slapos page.

1. Set monitoring password
2. Set recovery code



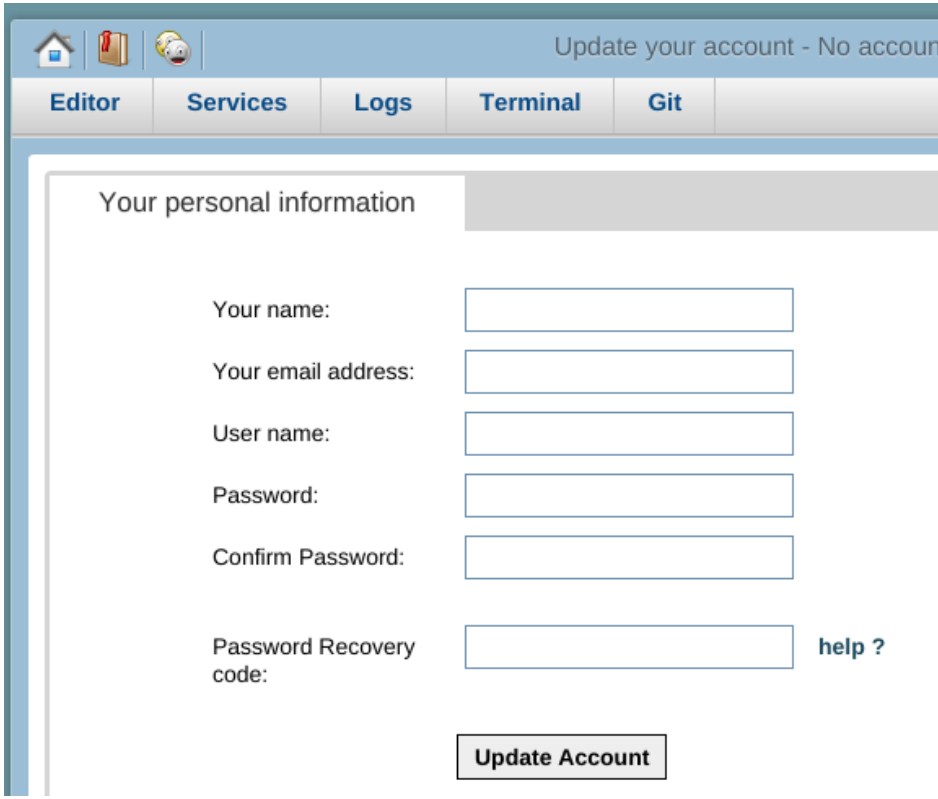
Install DREAM Runner

DREAM

Upon first login, configure your account

Set your name, email, login and password

Enter Password Recovery Code from monitoring interface



FoF.NMP.2012-6
Collaborative Project
Grant Agreement: 314364

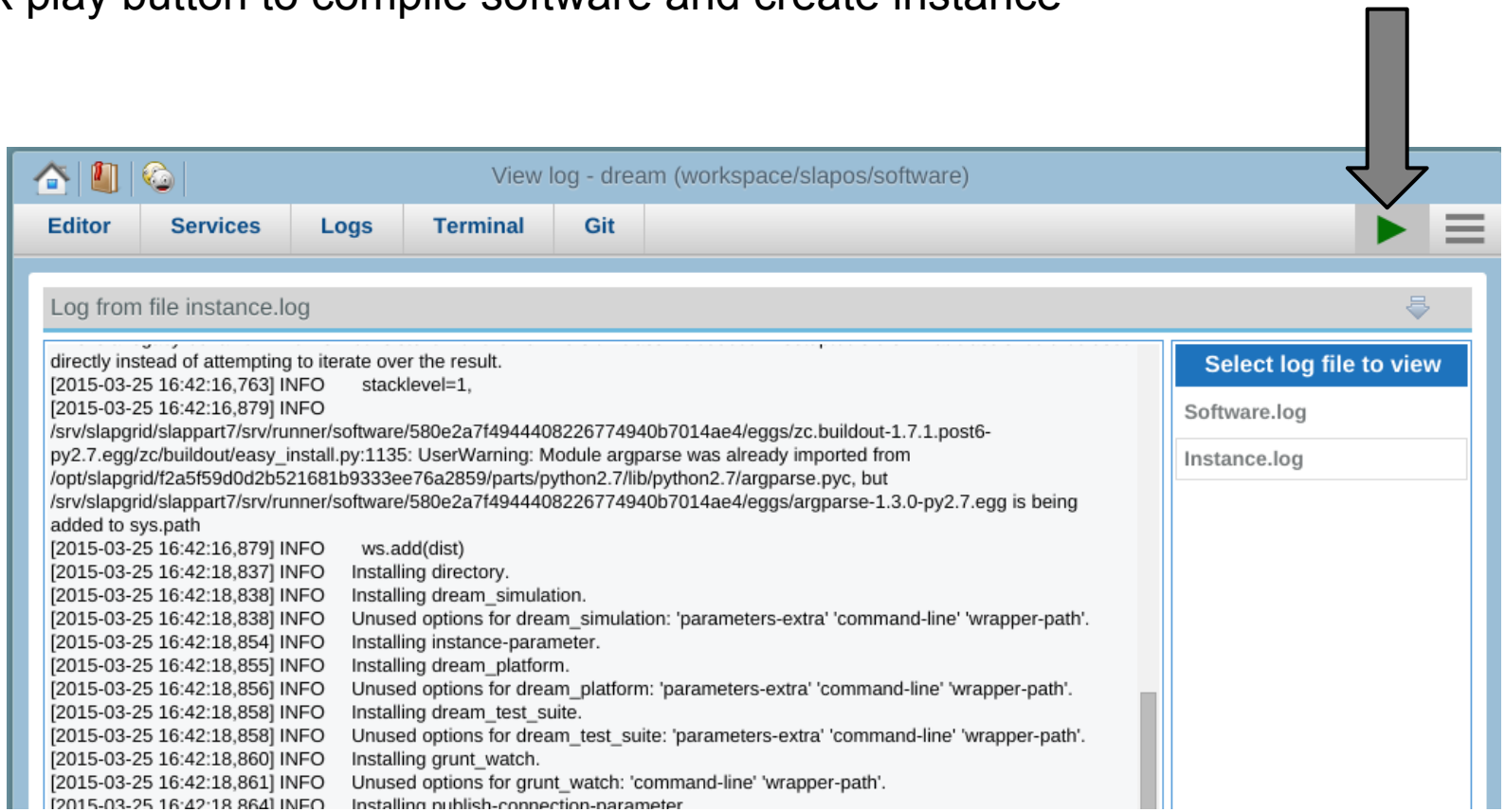
SEVENTH FRAMEWORK PROGRAMME

nexedi

Install DREAM Runner

DREAM

Click play button to compile software and create instance



Install DREAM Runner

DREAM

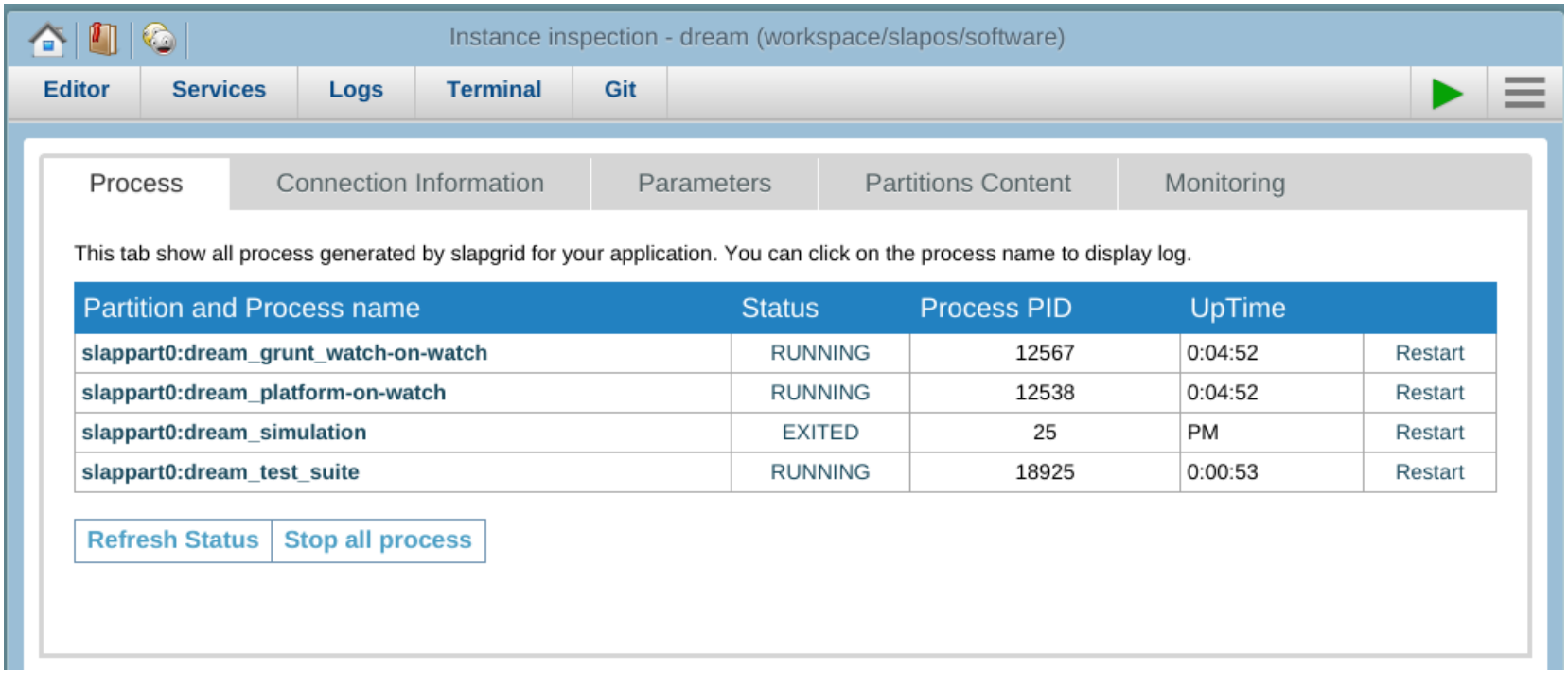
Wait for building state and running state to be Complete

Install DREAM Runner

DREAM

Check your processes are running in “Process” tab.

From this screen, you can also restart processes.



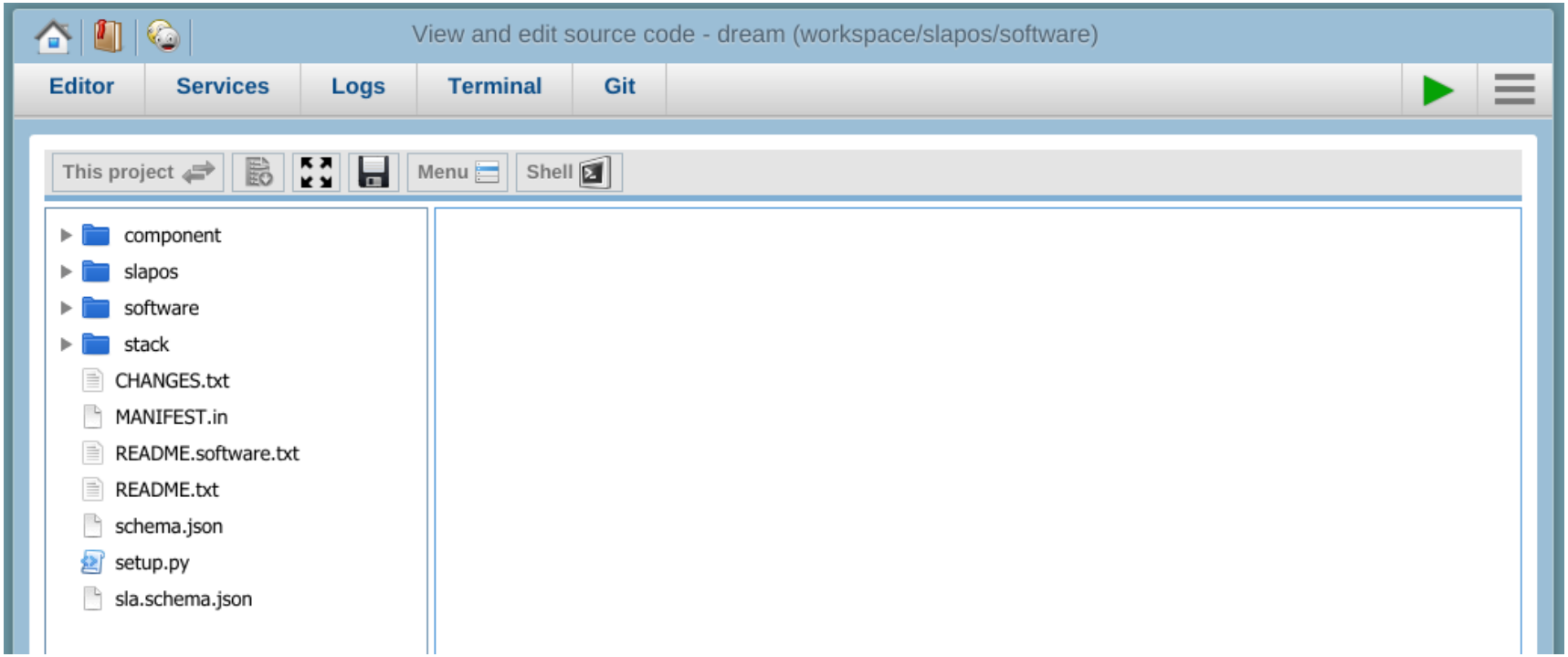
SEVENTH FRAMEWORK PROGRAMME

nexedi

Edit DREAM platform source code

DREAM

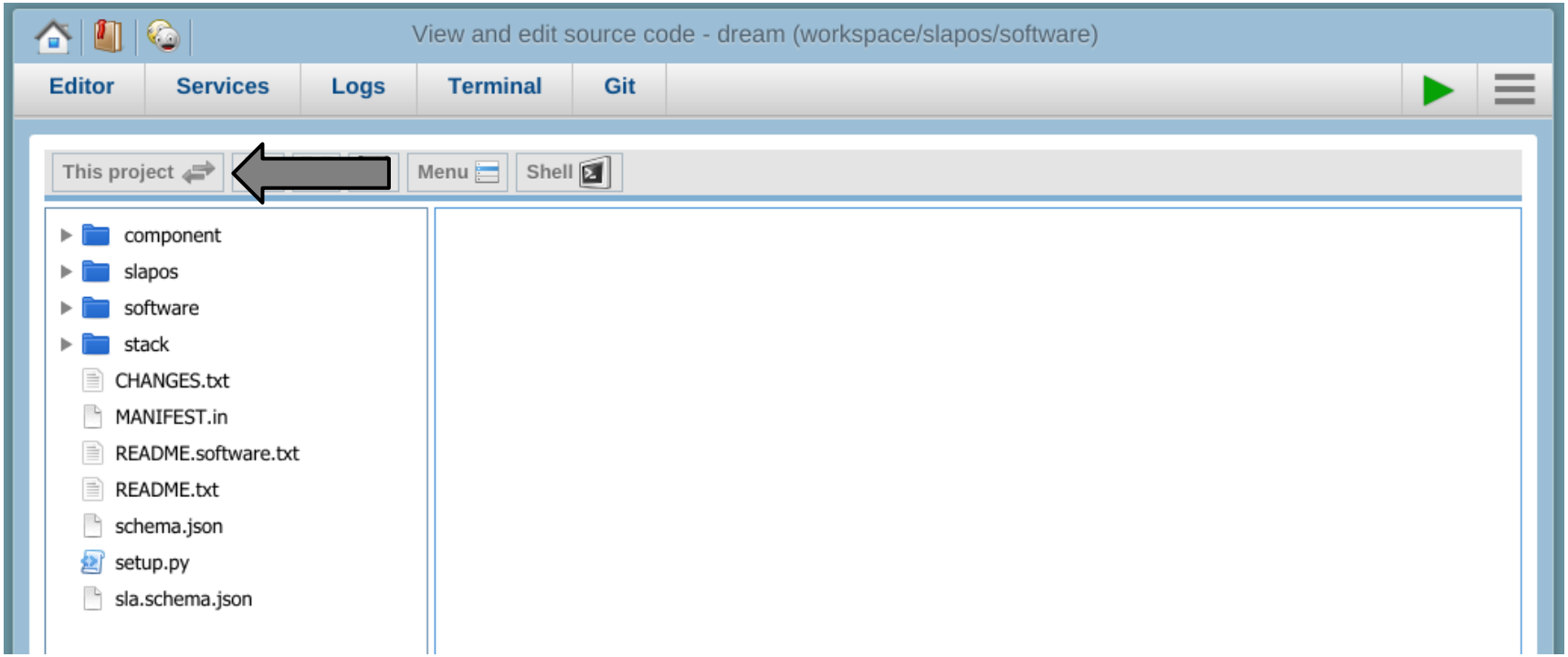
Editor tab let you edit files:



Edit DREAM platform source code

DREAM

To access DREAM source code, first switch from “This project” to “Working dir”



Edit DREAM platform source code

DREAM source code is in:

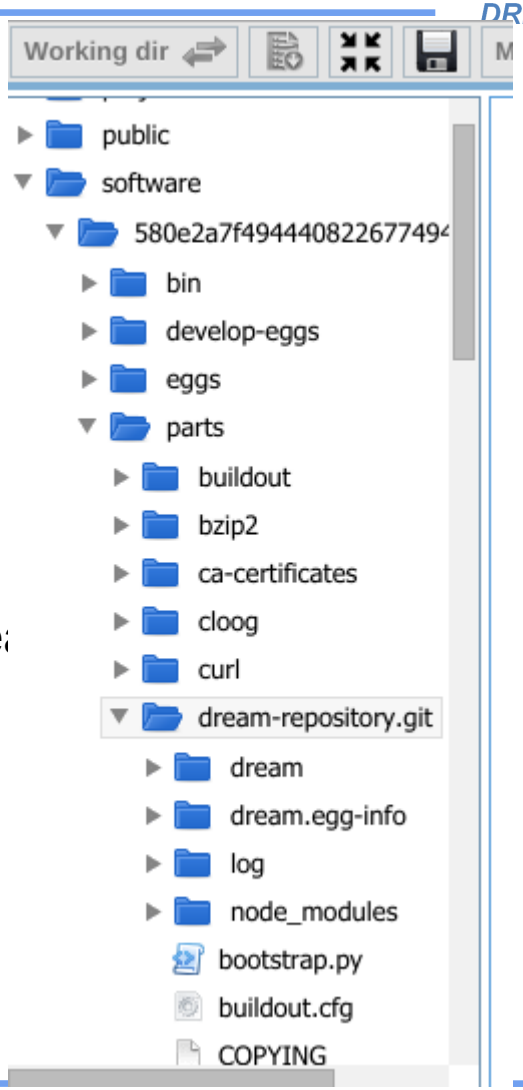
software

(the hash of the software)

parts

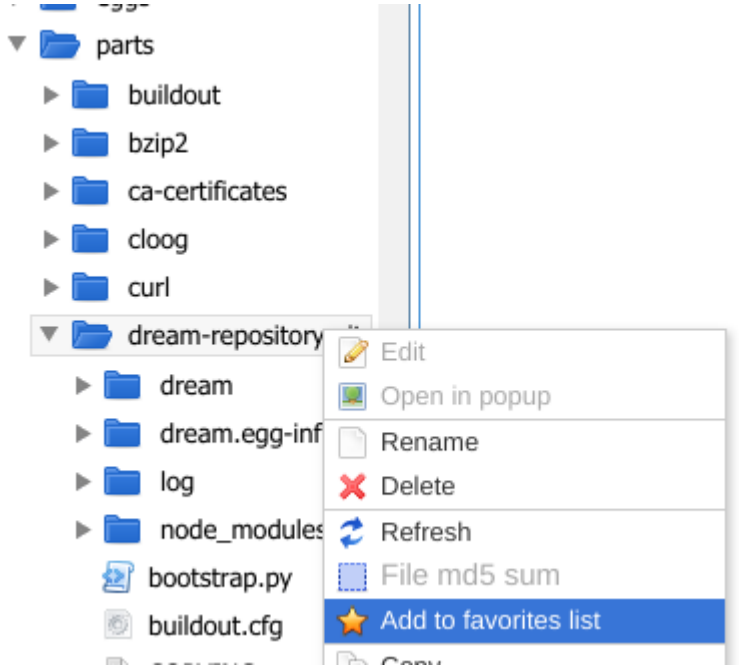
dream-repository.git

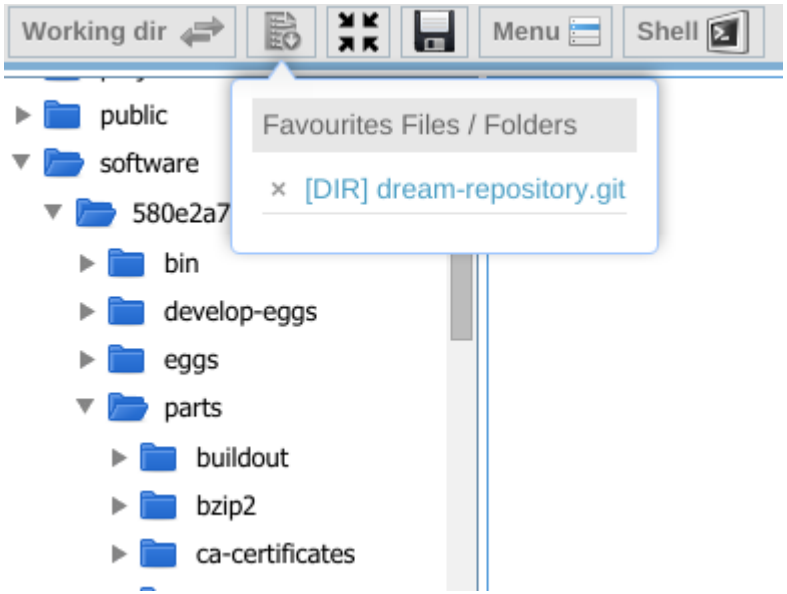
This is a git working copy of http://git.erp5.org/gitweb/dre
git/



Edit DREAM platform source code

Tips: add this folder for favourites, it will be available from the favourites menu

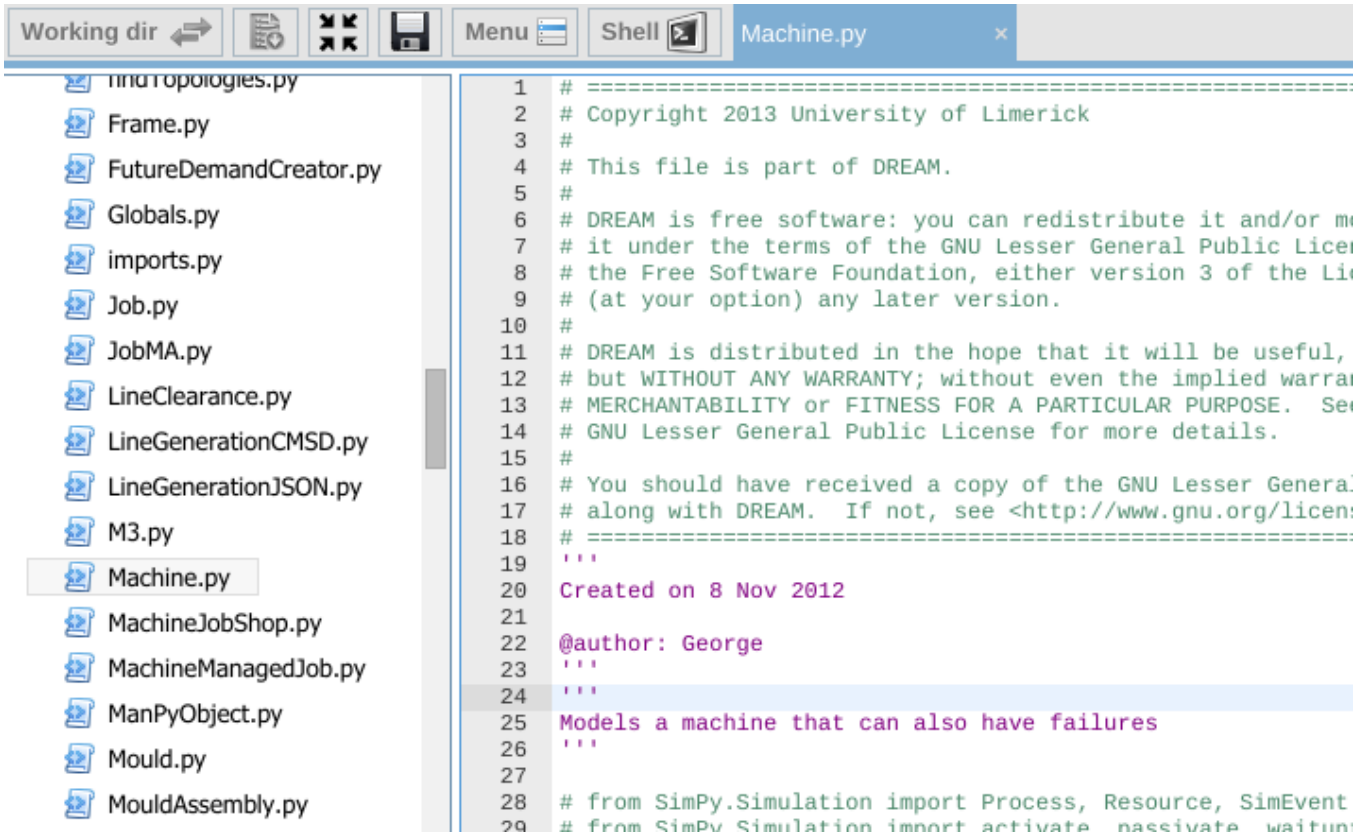




Edit DREAM platform source code

DREAM

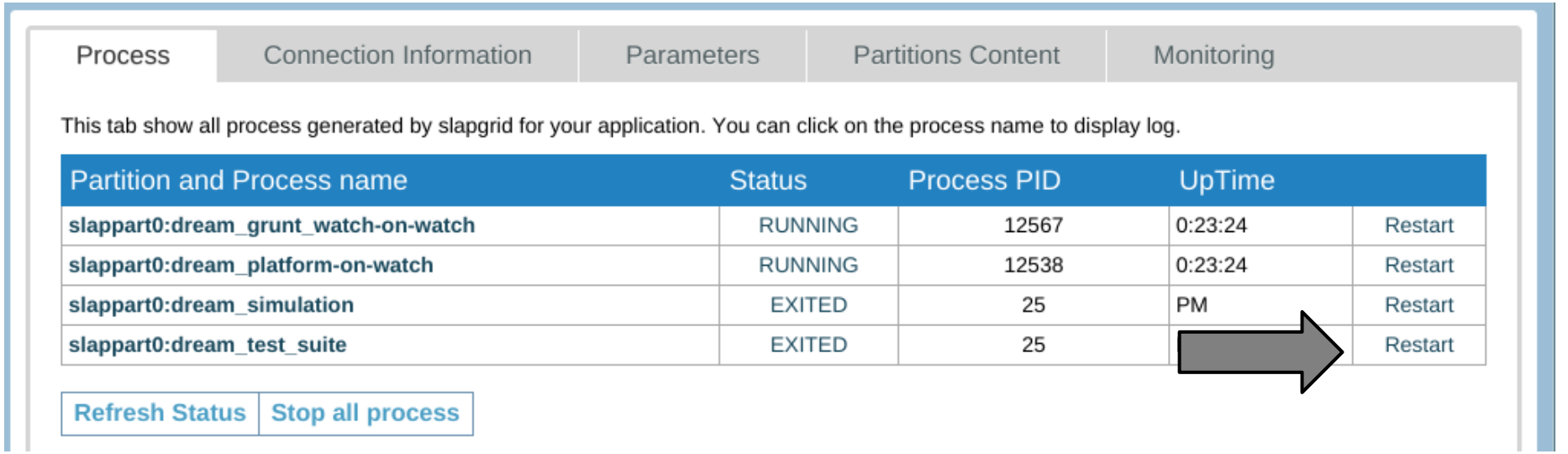
ManPy code can be edited :



Running DREAM test suite

DREAM

After making changes, run tests to validate your changes



FoF.NMP.2012-6
Collaborative Project
Grant Agreement: 314364

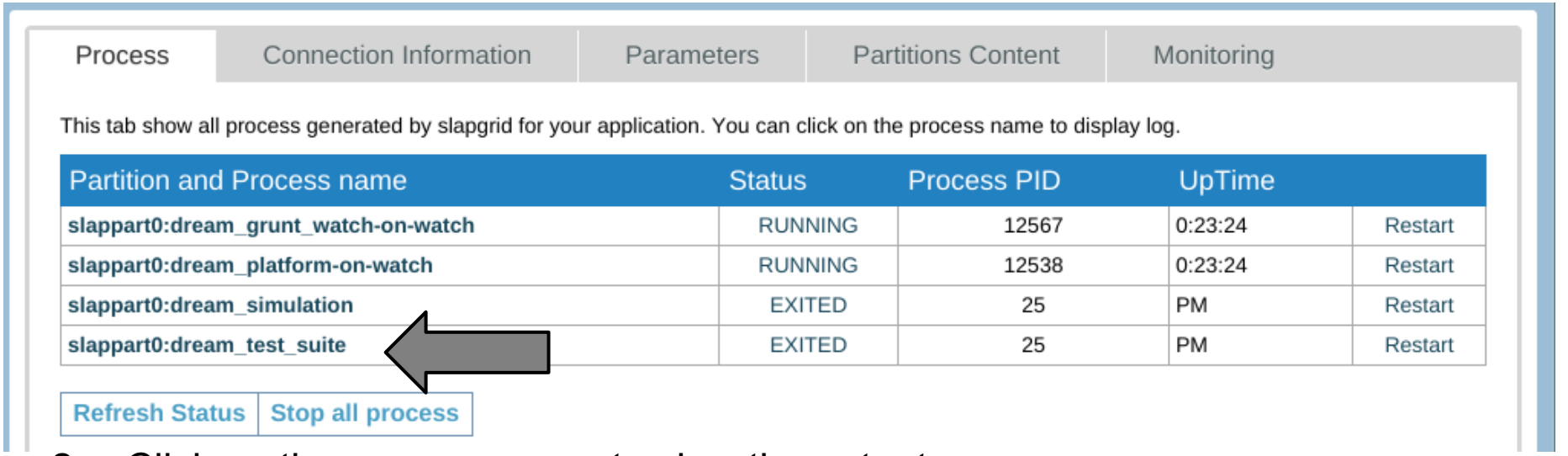
SEVENTH FRAMEWORK PROGRAMME

nexedi

Running DREAM test suite

DREAM

After making changes, run tests to validate your changes



2. Click on the process name to view the output

Accessing DREAM GUI

DREAM

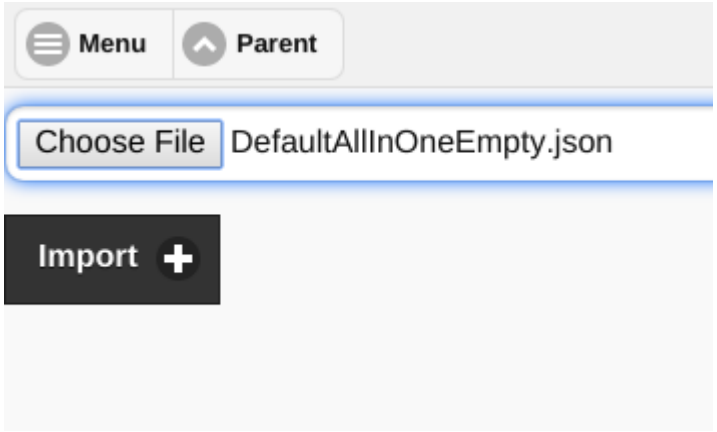
The GUI instance can be accessed from ***custom frontend URL*** in SlapOS parameter page.

Key	Value
backend_url	https://[2001:67c:1254:e:1c::b4c0]:50005
2_info	In order to set up your account, get the recovery-code from the monitoring interface. Before read the notification on monitor_info.
ssh_command	ssh 2001:67c:1254:e:1c::b4c0 -p 22222
url	https://softinst59074.host.vifib.net
public_url	https://softinst59072.host.vifib.net/public/
webdav_url	https://softinst59072.host.vifib.net/share/
git_public_url	https://[2001:67c:1254:e:1c::b4c0]:9684/git-public/
git_private_url	https://[2001:67c:1254:e:1c::b4c0]:9684/git/
custom-frontend-url	https://softinst59073.host.vifib.net
access_url	https://softinst59074.host.vifib.net/login
monitor_backend_url	https://[2001:67c:1254:e:1c::b4c0]:9684
1_info	On your first run, Use "access_url" to setup you account. Then you can use both "url" or "access_url". Or "backend_url" if you want to use ipv6. Set up your account in the webrunner in order to use webdav, and being able to clone your git repositories from the runner.
monitor_url	https://softinst59072.host.vifib.net

Accessing DREAM GUI

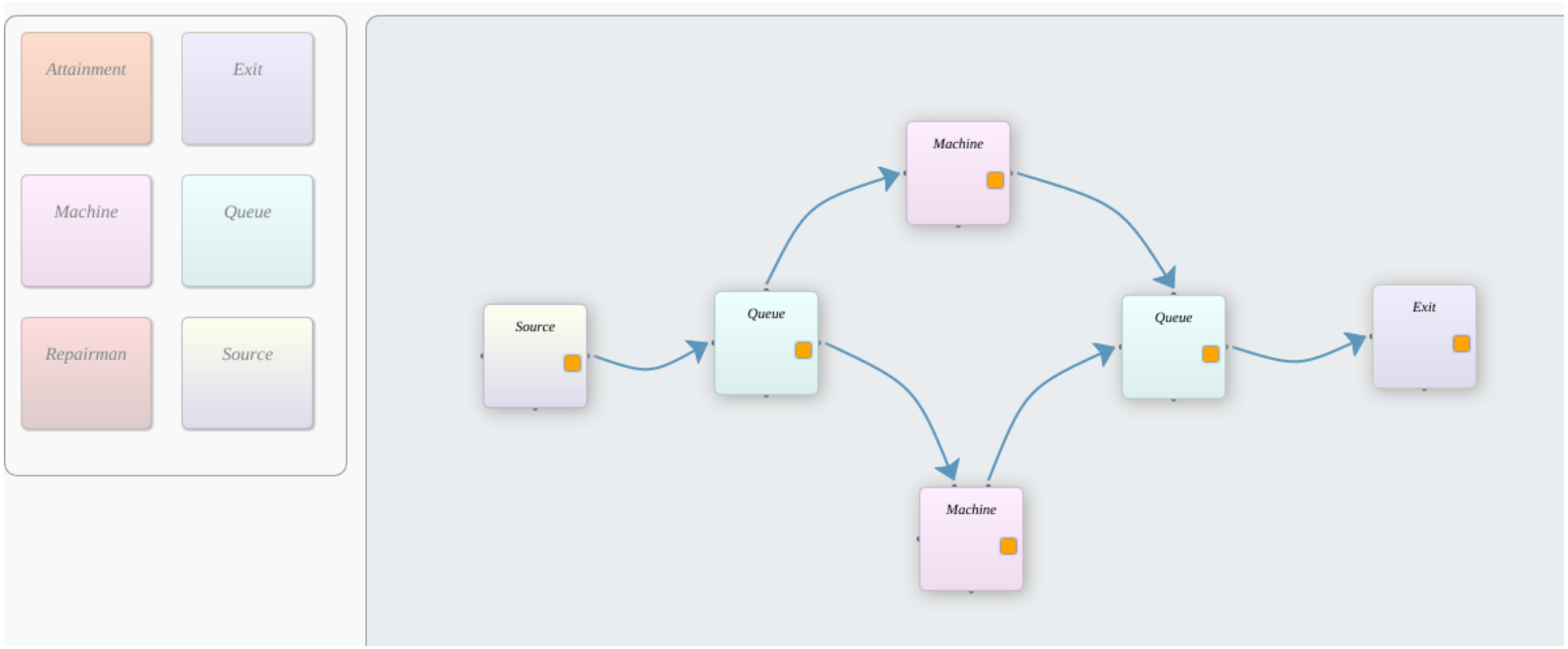
DREAM

Load a model, for example https://raw.githubusercontent.com/nexedi/dream/master/dream/simulation/Examples/GUI_instances/DefaultAllInOneEmpty.json



Accessing DREAM GUI

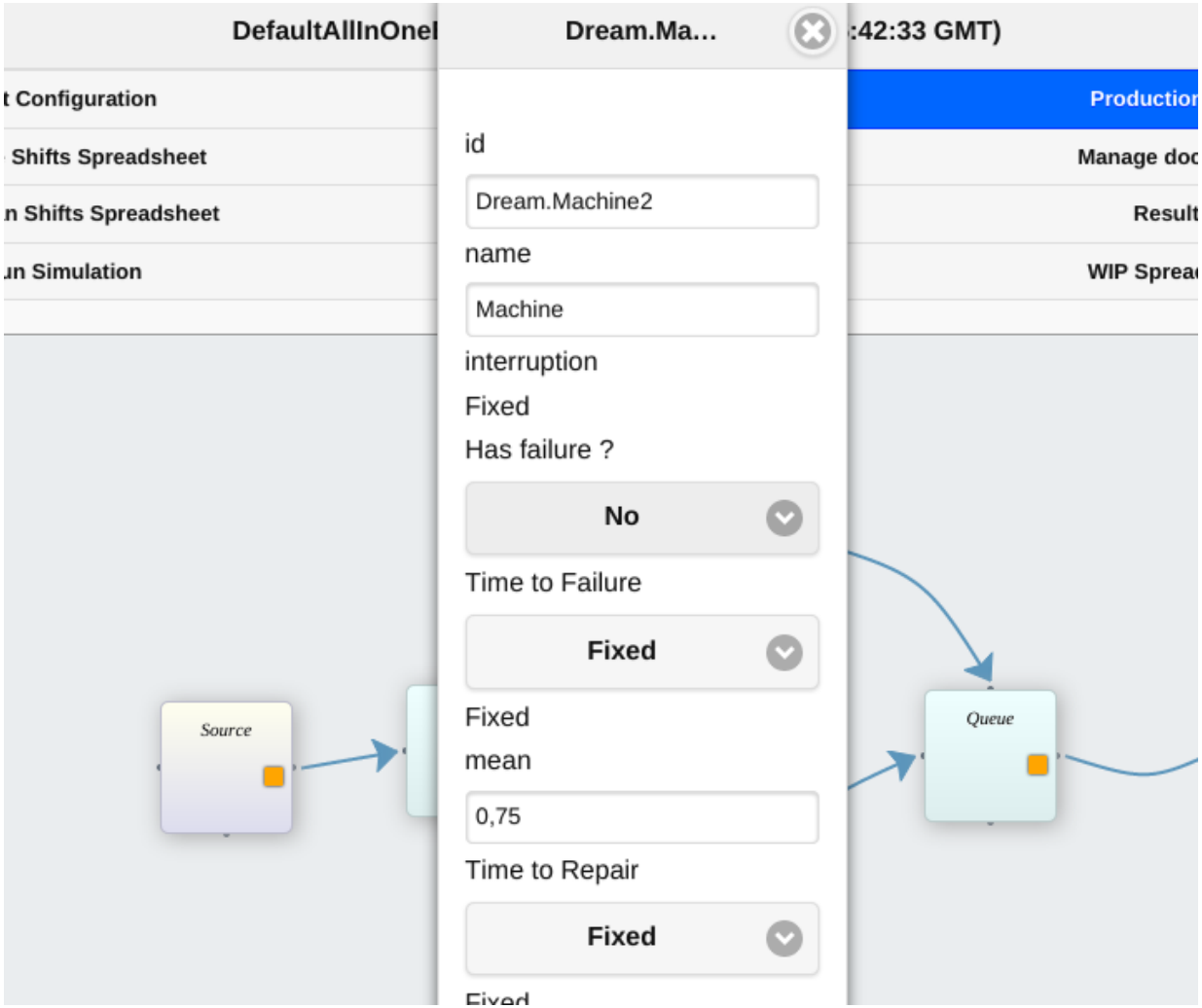
Create a simple model



Accessing DREAM GUI

DREAM

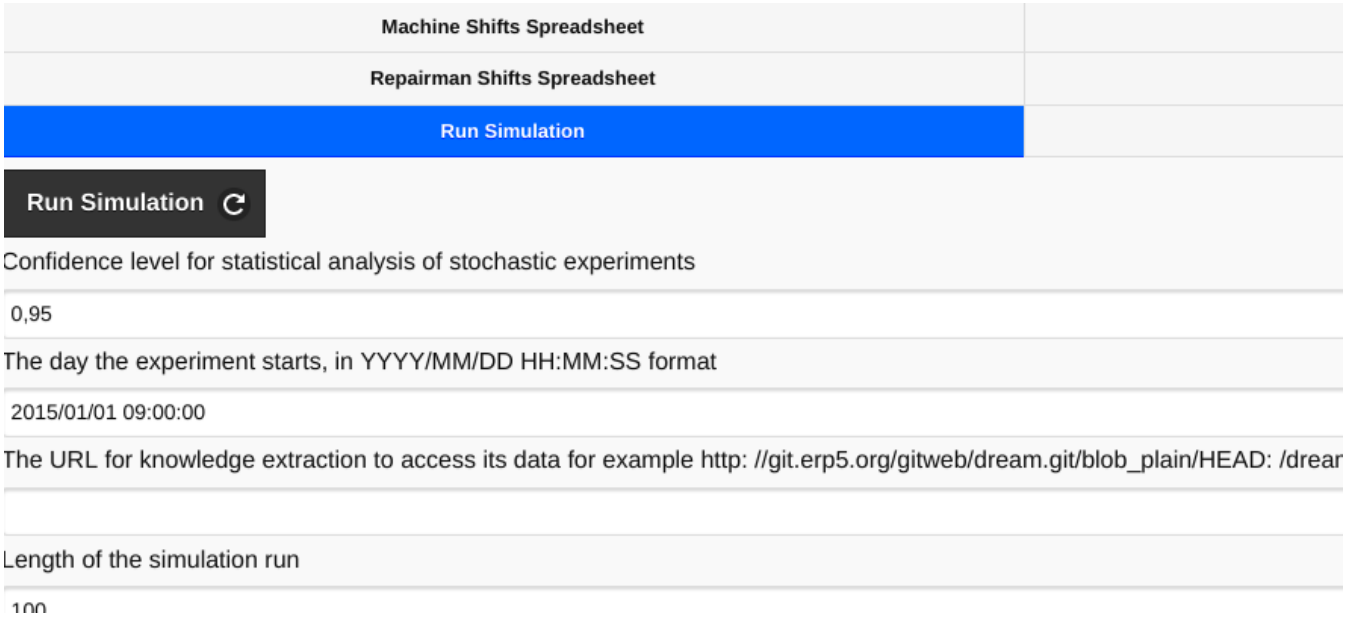
Edit model properties



Accessing DREAM GUI

DREAM

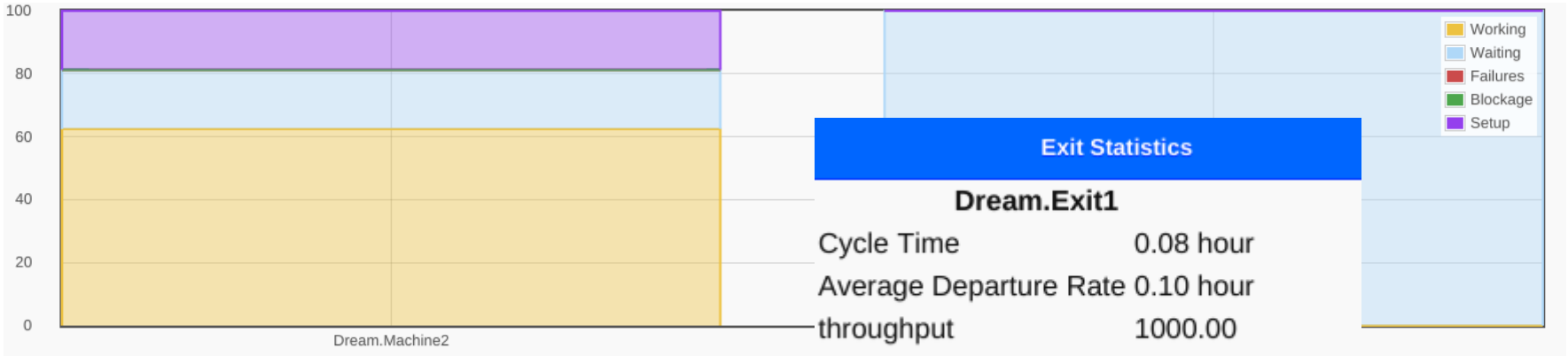
Running simulation



Accessing DREAM GUI

DREAM

Visualise results



Exit Statistics

Dream.Exit1

Cycle Time	0.08 hour
Average Departure Rate	0.10 hour
throughput	1000.00

Exit Statistics	Queue Statistics	Station Utilization	Tabular_Results

Exit Id	Throughput	Takt Time	Lifespan
Dream.Exit1	1000	0.0998999999999986	0.08125000000000017

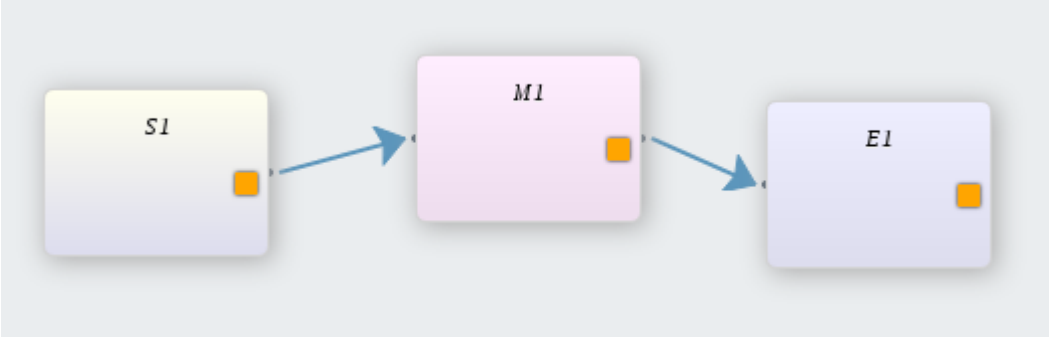
Introduction to JSON format

Only nodes

```
{
  "node": {
    "S1": {
      "name": "S1"
    },
    "M1": {
      "name": "M1"
    },
    "E1": {
      "name": "E1"
    }
  },
  "edge": { }
}
```

Simple Representation of a Directed Graph

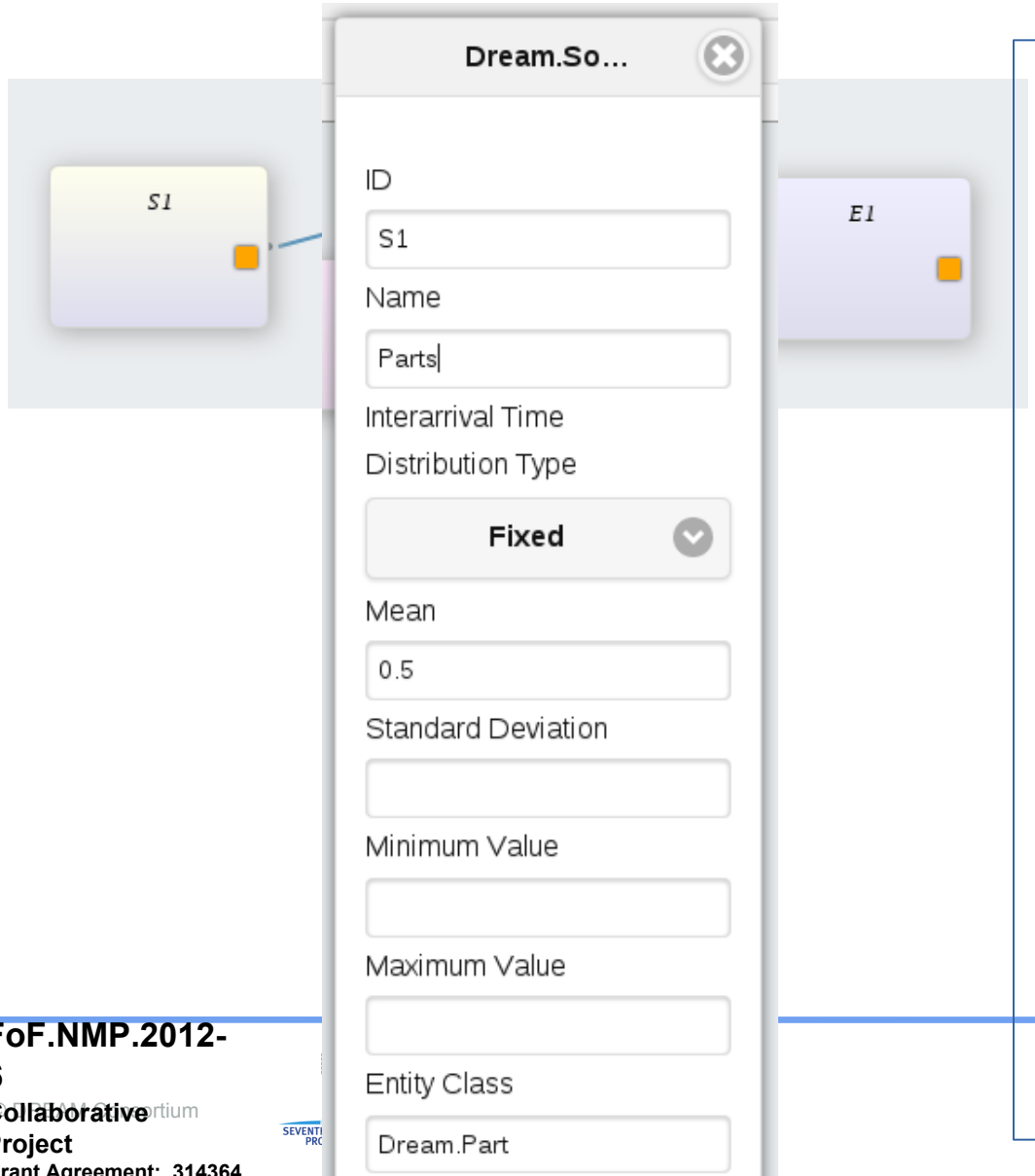
Introduction to JSON format



Nodes connected by edges

```
{
  "node": {
    "S1": {
      "name": "S1"
    },
    "M1": {
      "name": "M1"
    },
    "E1": {
      "name": "E1"
    }
  },
  "edge": {
    "edge1": {
      "source" : "S1",
      "destination" :"M1",
    },
    "edge2":
      "source" : "M1",
      "destination" :"E1",
    }
  }
}
```

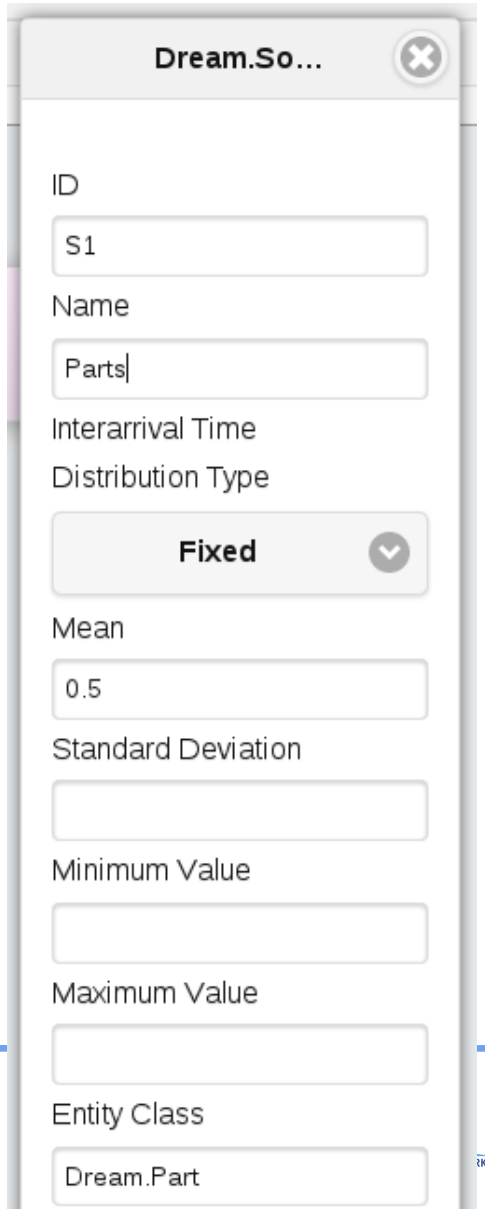
Introduction to JSON format



Properties of nodes

```
{
  "node": {
    "S1": {
      "name": "Parts",
      "_class": "Dream.Source",
      "interarrivalTime": {
        "distributionType": "Fixed",
        "mean": 0.5
      },
      "entity": "Dream.Part"
    },
    "M1": {
      "name": "M1"
    },
    "E1": {
      "name": "E1"
    }
  },
  "edge": {
    "edge1": {
      "source" : "S1",
      "destination" :"M1",
    },
    "edge2":
      "source" : "M1",
      "destination" :"E1",
    }
  }
}
```

Introduction to JSON format



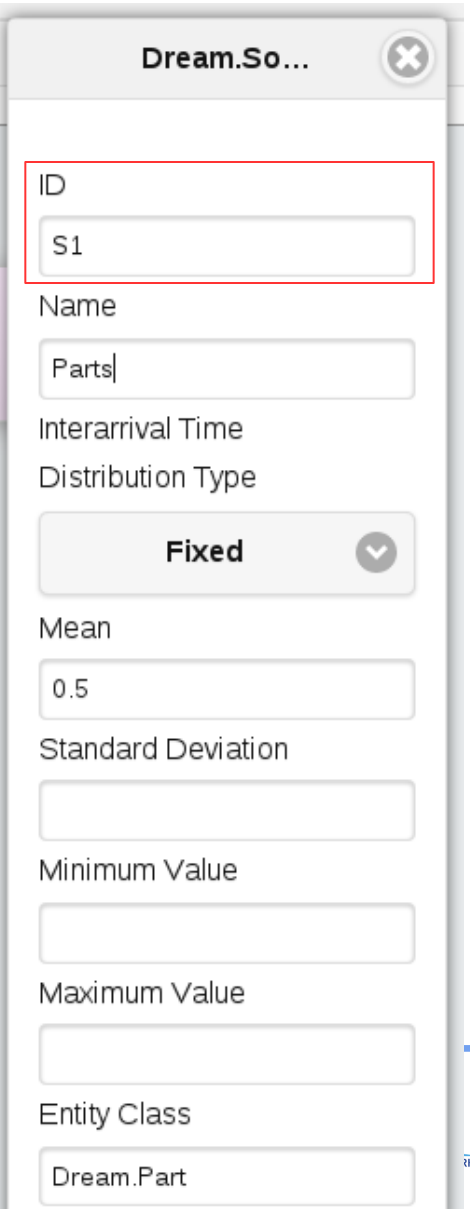
- What are the editable properties ?
- Which value type (text, number) ?
- What possible values ?
- What default value ?

Introduction to JSON format

DREAM

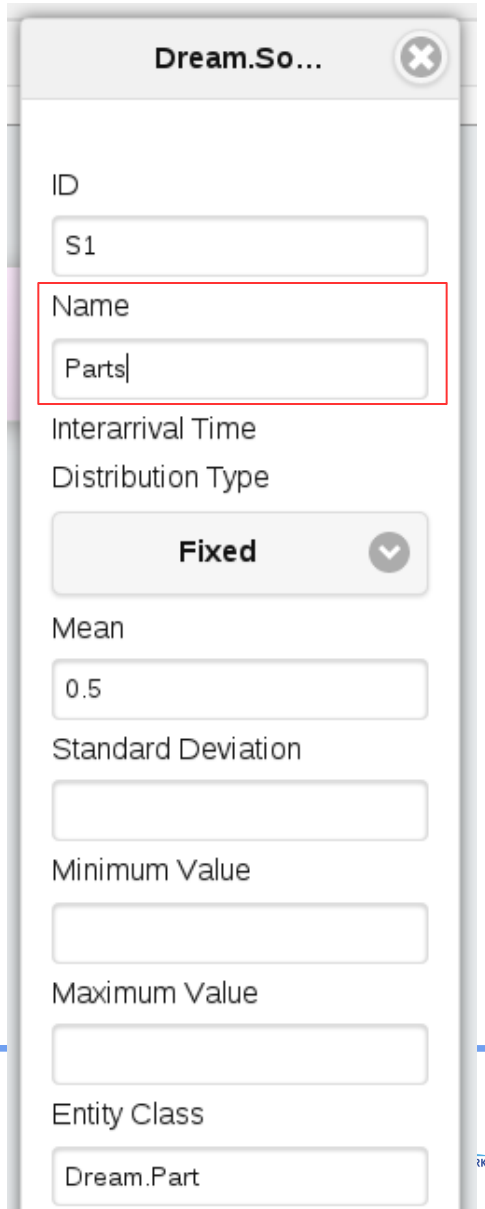
Properties are defined as JSON Schema

```
"class_definition": {
    "Dream.Source" : {
        "properties" : {
            "id" : {
                "type" : "string",
                "default" : "S",
                "required" : true,
                "name": "ID",
                "description": "The ID of this source"
            },
            "name" : {
                "type" : "string",
                "default" : "Source"
            },
            "interarrivalTime" : {
                "type": "object",
                "properties": {
                    "distributionType": {
                        "type": "string",
                        "enum": [ "Fixed",
                                  "Normal",
                                  "Exponential"],
                    "default": "Fixed"
                },
                "mean": {
                    "type": "number",
                    "default": 0.5
            }
```



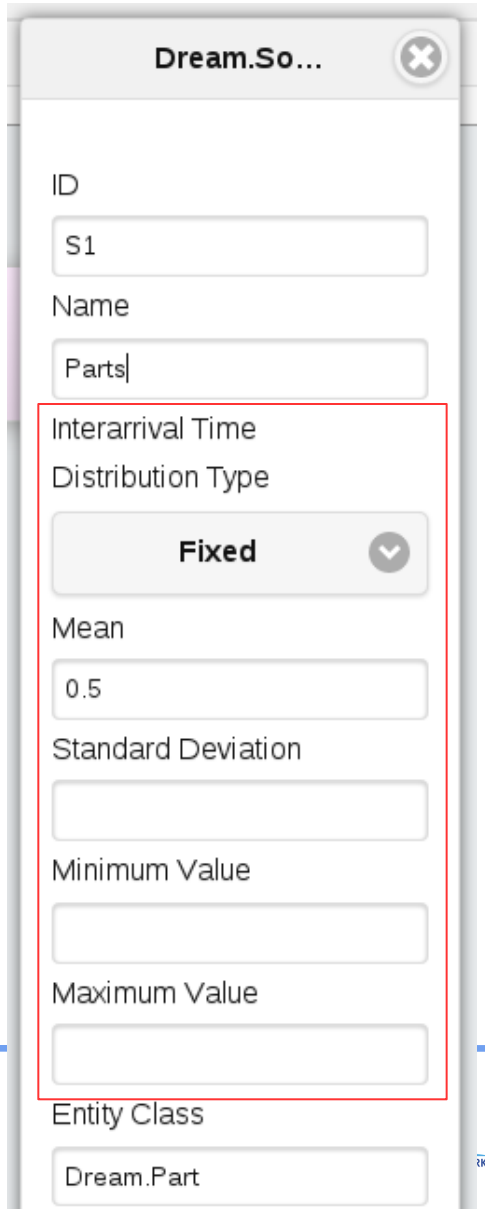
nexedi

Introduction to JSON format



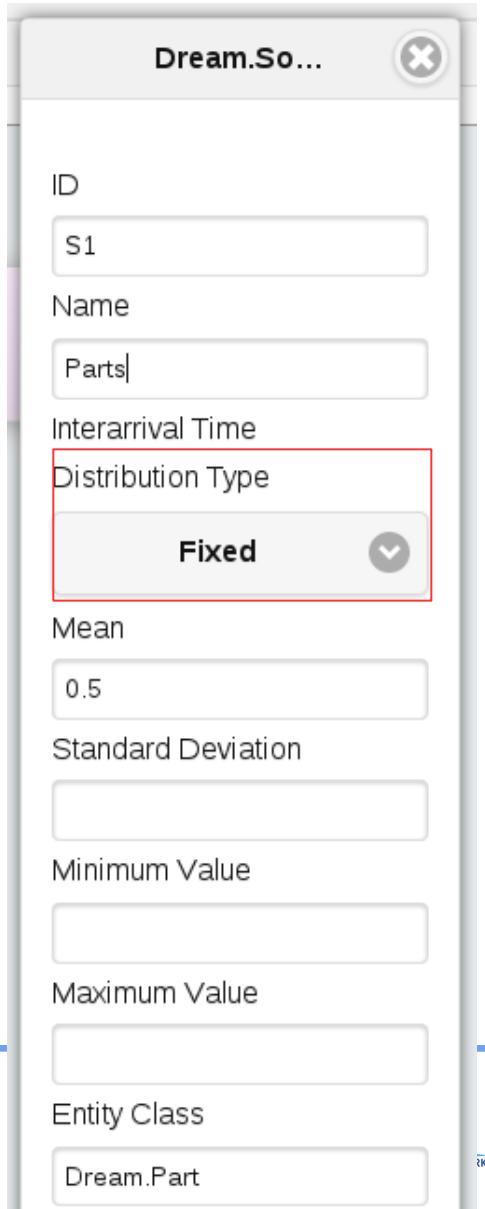

```
"class_definition": {
    "Dream.Source" : {
        "properties" : {
            "id" : {
                "type" : "string",
                "default" : "S",
                "required" : true,
                "name": "ID",
                "description": "The ID of this source"
            },
            "name" : {
                "type" : "string",
                "default" : "Source"
            },
            "interarrivalTime" : {
                "type": "object",
                "properties": {
                    "distributionType": {
                        "type": "string",
                        "enum": [ "Fixed",
                                  "Normal",
                                  "Exponential"],
                        "default": "Fixed"
                    },
                    "mean": {
                        "type": "number",
                        "default": 0.5
                }
```

Introduction to JSON format



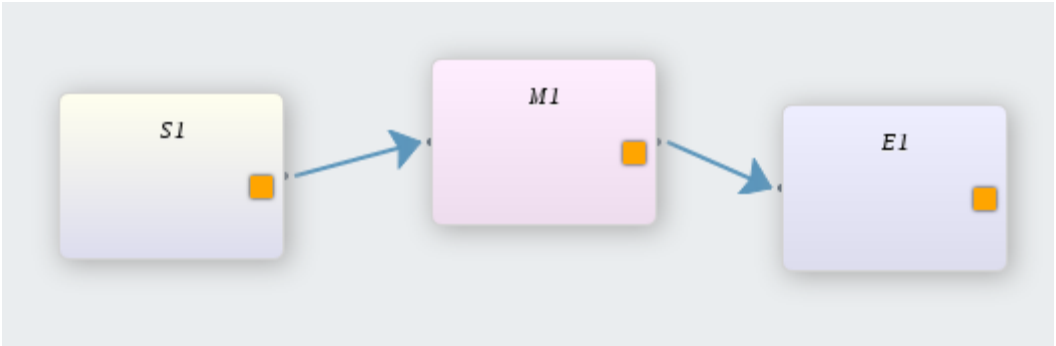

```
"class_definition": {
    "Dream.Source" : {
        "properties" : {
            "id" : {
                "type" : "string",
                "default" : "S",
                "required" : true,
                "name": "ID",
                "description": "The ID of this source"
            },
            "name" : {
                "type" : "string",
                "default" : "Source"
            },
            "interarrivalTime" : {
                "type": "object",
                "properties": {
                    "distributionType": {
                        "type": "string",
                        "enum": [ "Fixed",
                                  "Normal",
                                  "Exponential"],
                        "default": "Fixed"
                    },
                    "mean": {
                        "type": "number",
                        "default": 0.5
                }
```

Introduction to JSON format




```
"class_definition": {
    "Dream.Source" : {
        "properties" : {
            "id" : {
                "type" : "string",
                "default" : "S",
                "required" : true,
                "name": "ID",
                "description": "The ID of this source"
            },
            "name" : {
                "type" : "string",
                "default" : "Source"
            },
            "interarrivalTime" : {
                "type": "object",
                "properties": {
                    "distributionType": {
                        "type": "string",
                        "enum": [ "Fixed",
                                  "Normal",
                                  "Exponential"],
                        "default": "Fixed"
                    },
                    "mean": {
                        "type": "number",
                        "default": 0.5
            }
```

Introduction to JSON format

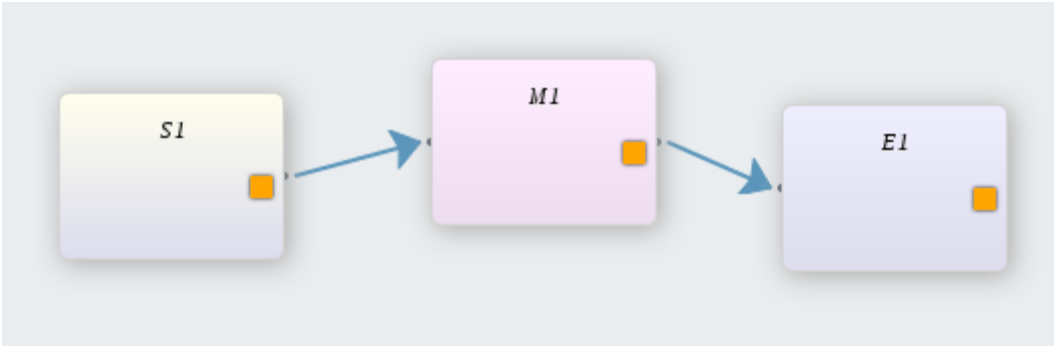


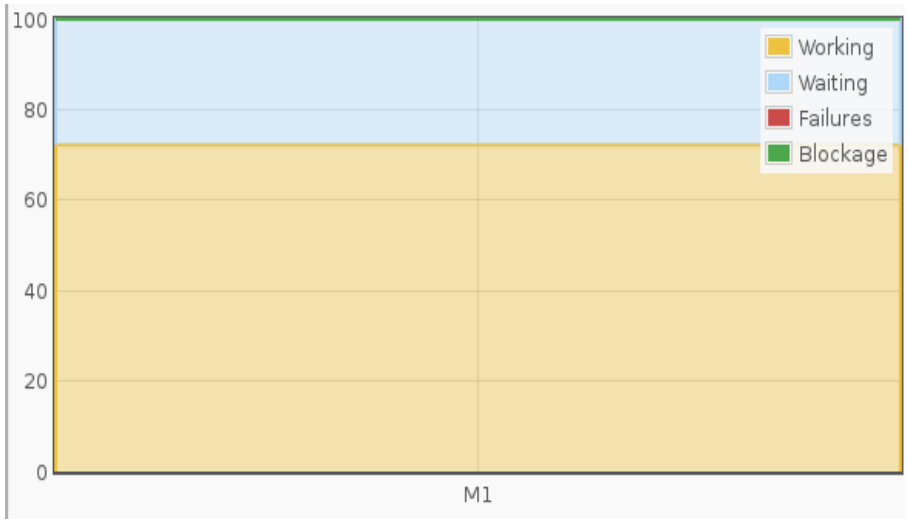

```
{
  "graph": {
     "node": { … },
     "edge": { … }
  },
  "class_definition": { … },
  "input": [
     "WipSpreadsheet" : {
     ["Order 1", "2014/03/15",
"1", "PM1", "Design" … ],
     …
  ]
},
}
```

Order ID	Due Date	Priority	Project Manager	Part	Part Type	Sequence		
Order 1	2014/03/15	1	PM1	Design	Design	CAD		
				Part1	Basic	CAM-MILL-EDM-MILL-MASS		
				Part2	Basic	CAM-MILL-EDM-MILL-EDM-MAS		
				Assemble	Mould	MASS-IM	2-12	
Order 2	2014/03/14	1	PM1	Design	Design	CAD	6	
				Part1	Basic	CAM-MILL-EDM-MILL-MASS	8-4-2-8-0	
				Part2	Basic	CAM-MILL-EDM-MASS	20-15-8-8-0	
				Assemble	Mould	MASS-IM	1-12	
Order 3	2014/03/15	1	PM1	Design	Design	CAD	6	
				Part1	Basic	CAM-MILL-EDM-MASS	8-4-2-0	
				Part2	Basic	CAM-MILL-EDM-MASS	7-15-8-8-0	
				Assemble	Mould	MASS-IM	1-3	

General purpose tabular inputs in the same JSON

Introduction to JSON format





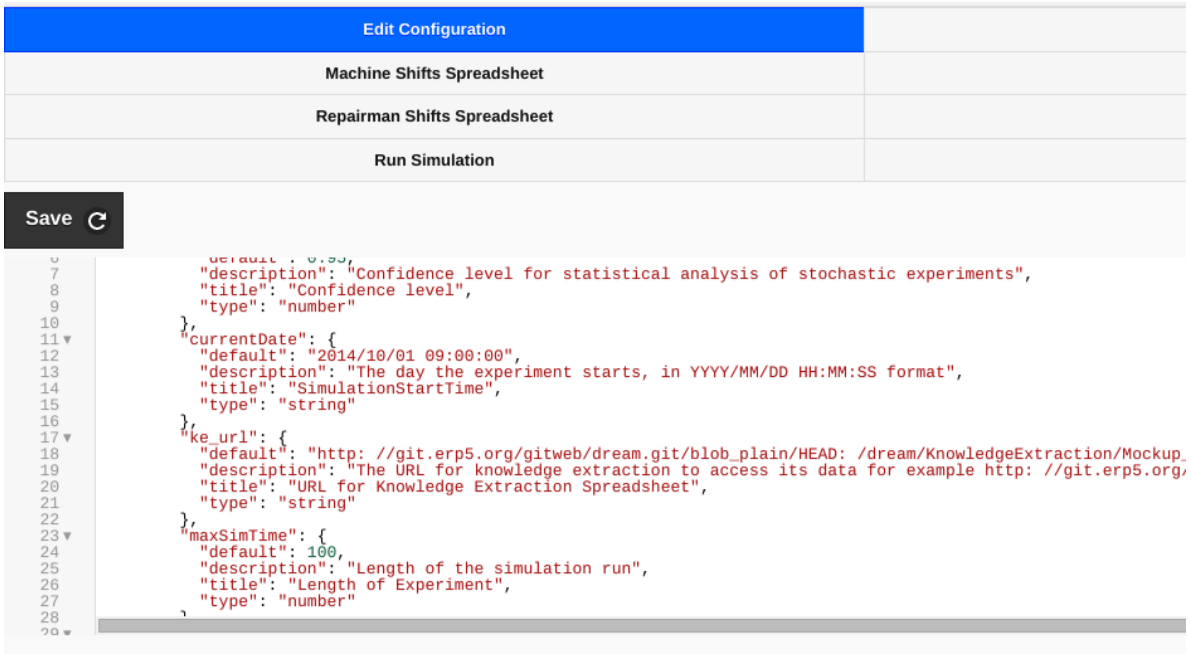

```
{
  "graph": {
     "node": { … },
     "edge": { … }
  },
  "class_definition": { … },
  "input": { … },
  "output": {
    "M1": {
      "working_ratio": 0.7,
      "waiting_ratio": 0.3,
      "failure_ratio": 0,
      "bloacage_ratio": 0
  }
}
```

Outputs - results of simulation - are also in the same JSON

Introducing Edit Configuration tab

DREAM

Live edition of JSON



SEVENTH FRAMEWORK PROGRAMME

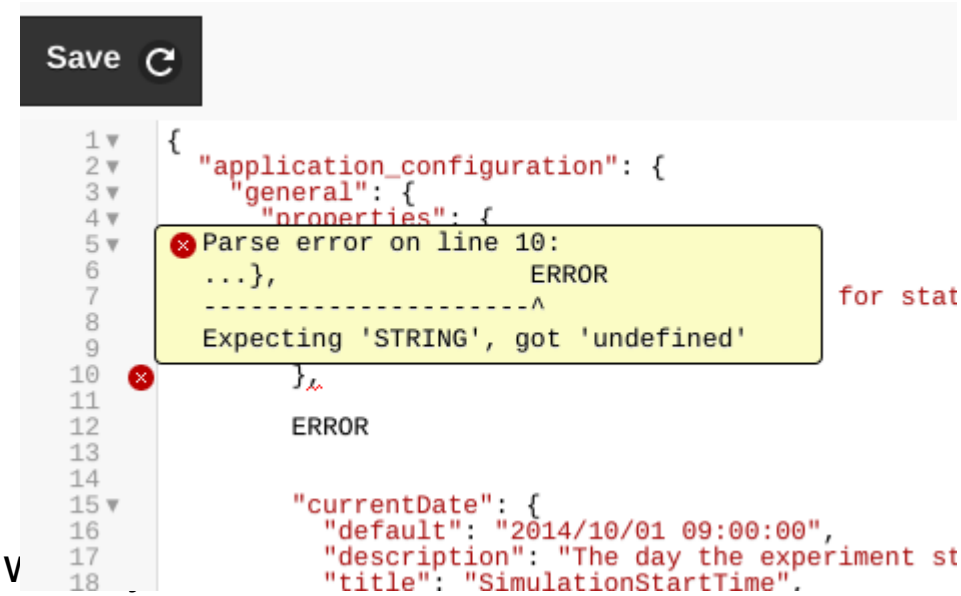
nexedi

Introducing Edit Configuration tab

DREAM

Live edition of JSON

Features non obtrusive assistance w



Graph editor: adding a property on node

DREAM

On Edit configuration tab:

Locate the json schema definition of Dream.Queue in *class_definition*

```
{
  "application_configuration": {↔},
  "class_definition": {
    "Dream.Edge": {↔},
    "Dream.EventGenerator": {↔},
    "Dream.Exit": {↔},
    "Dream.Machine": {↔},
    "Dream.Queue": {
      "_class": "node",
      "allOf": [
        {
          "$ref": "#/node"
        },
        {
          "properties": {
            "capacity": {
              "$ref": "#/definitions/_capacity",
              "required": true
            },
            "id": {
              "default": "Q",
              "type": "string"
            },
```

Graph editor: adding a property on node

DREAM

On Edit configuration tab:

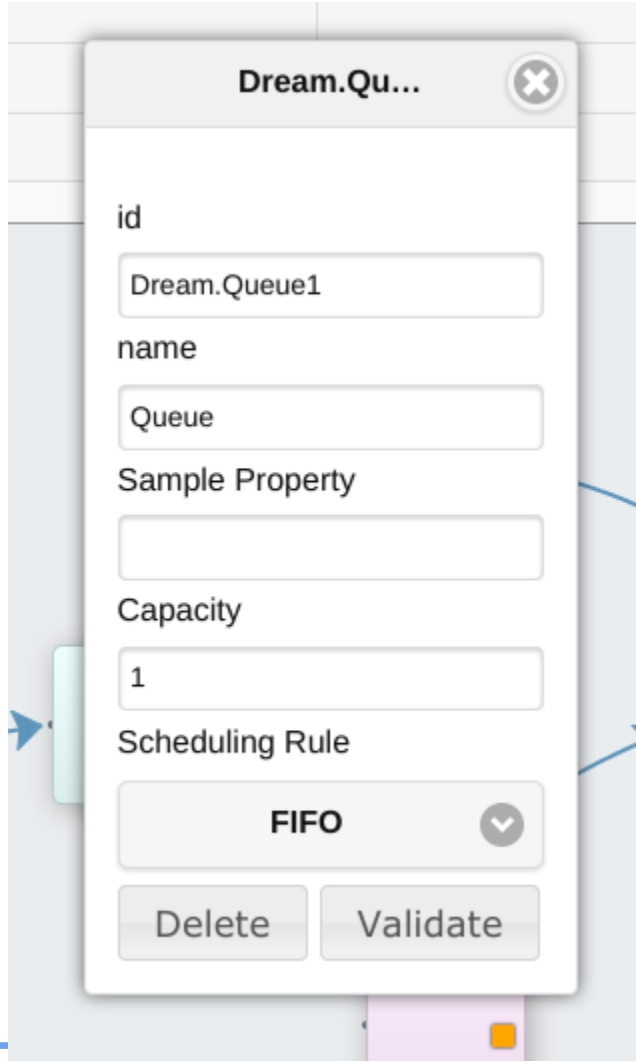
Add a new property, in this example it is called “Sample property”

```
264 ▸    "Dream.EventGenerator": {↔},
329 ▸    "Dream.Exit": {↔},
358 ▸    "Dream.Machine": {↔},
433 ▾    "Dream.Queue": {
434        "_class": "node",
435 ▾      "allOf": [
436 ▾        {
437            "$ref": "#/node"
438          },
439 ▾        {
440 ▾          "properties": {
441 ▾            "sample_property": {
442                "name": "Sample Property",
443                "description": "The tooltip to display.",
444                "type": "string"
445              },
446 ▾            "capacity": {
447                "$ref": "#/definitions/_capacity",
448                "required": true
449              },
450 ▾            "id": {
451                "default": "Q",
452                "type": "string"
```

Graph editor: adding a property on node

On “Production Line” tab, we have a new property when editing a Dream.Queue.

Set value “user value” and save



Graph editor: adding a property on node

DREAM

Go back to “Edit configuration” tab, we can see that the sample_property is saved as a property on the node.

SEVENTH FRAMEWORK PROGRAMME

nexedi

Application configuration: adding a tab

DREAM

On Edit configuration tab:

Locate the input action definition in *application_definition*

input is for input action

output is for results visualisation

```
{
  "application_configuration": {
    "general": {↔},
    "input": {
      "debug": {
        "gadget": "Input_viewDebugJson",
        "title": "Edit Configuration",
        "type": "object_view"
      },
      "view": {
        "gadget": "Input_viewProductionLine",
        "title": "Production Line",
        "type": "object_view"
      },
      "view_machine_shift_spreadsheet": {
        "configuration": {
          "columns": [
            {
              "format": "date-time",
              "name": "Date",
              "type": "string"
            },
            {
```

Application configuration: adding a tab

DREAM

Add an action:

gadget: The gadget to use

title: The name to display on the tab

type: Always object_view for now

configuration: Gadget configuration, passed to gadget constructor.

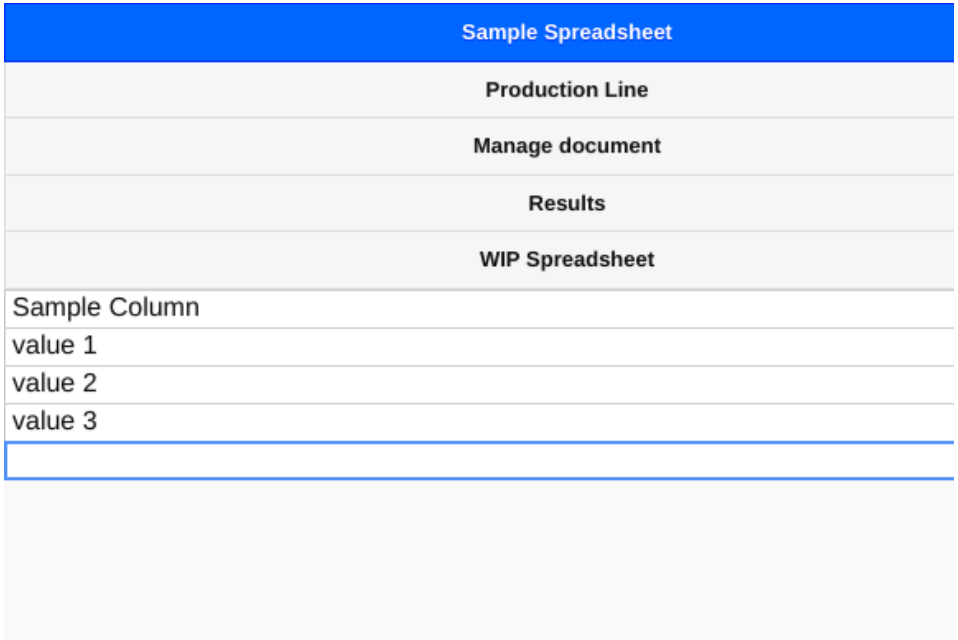
- *columns*: columns definition
- *input_id*: key for this input

```
        }
      }
    },
    "input": {
      "sample_spreadsheet": {
        "gadget": "Input_viewSpreadsheet",
        "title": "Sample Spreadsheet",
        "type": "object_view",
        "configuration": {
          "input_id": "sample_spreadsheet_input",
          "columns": [
            {
              "name": "Sample Column",
              "type": "string"
            }
          ]
        }
      },
      "debug": {
```

Application configuration: adding a tab

DREAM

The new tab appears. Set some values



Application configuration: adding a tab

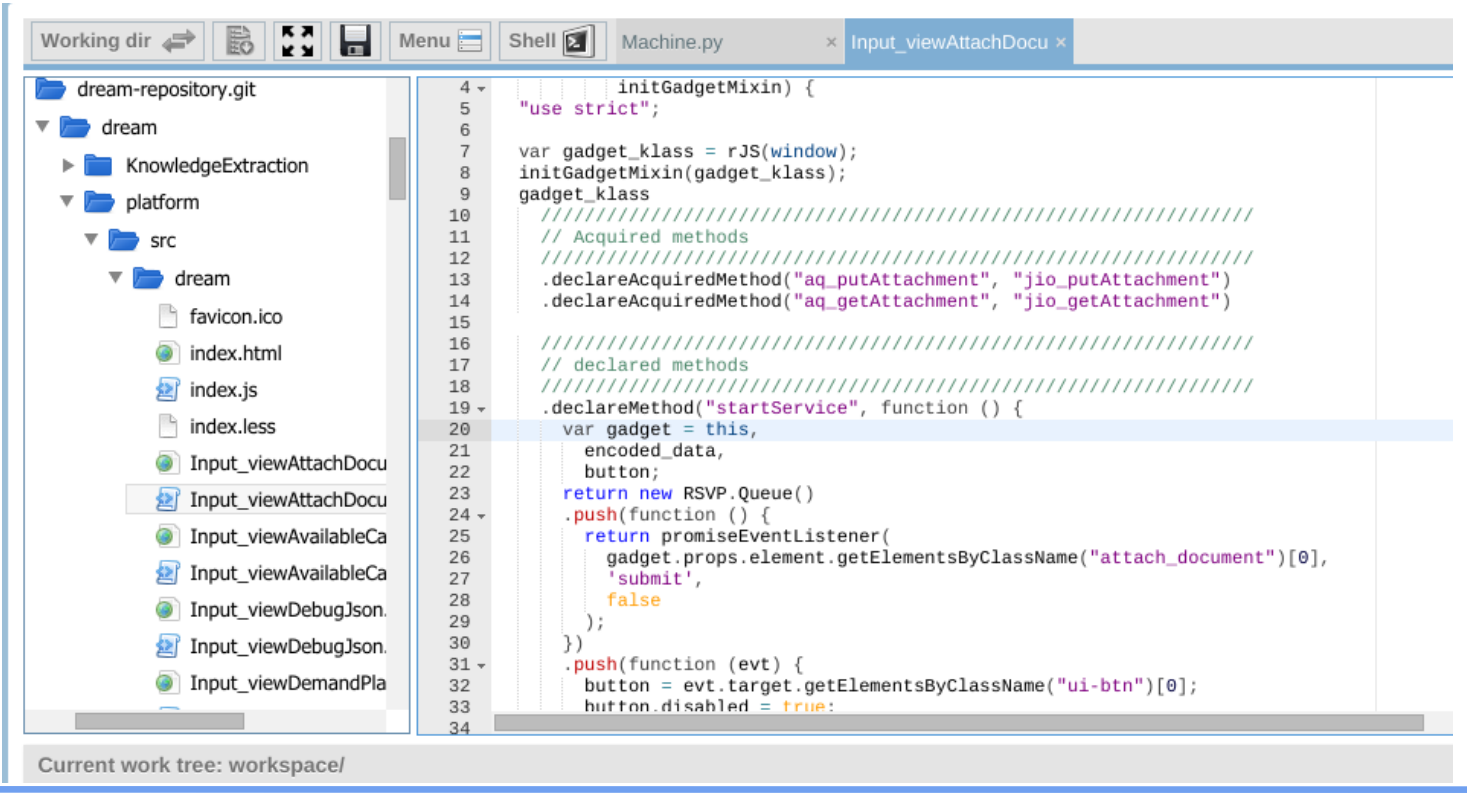
DREAM

The values are saved in the JSON, as input / (the input_id configured)

```
267 ▸   "class_definition": {↔},
929     "constraints": {},
930 ▸   "general": {↔},
941 ▸   "graph": {↔},
1079 ▾  "input": {
1080 ▸    "repairman_shift_spreadsheet": [↔],
1094 ▸    "wip_spreadsheet": [↔],
1106 ▸    "machine_shift_spreadsheet": [↔],
1130 ▾    "sample_spreadsheet_input": [
1131 ▾      {
1132          "name": "Sample Column"
1133        },
1134 ▾      {
1135          "name": "value 1"
1136        },
1137 ▾      {
1138          "name": "value 2"
1139        },
1140 ▾      {
1141          "name": "value 3"
1142        },
1143 ▾      {
1144          "name": null
```

Application configuration: adding a tab

To add a completely new gadget, locate the gadget folder in dream-repository.git/dream/platform/src/dream



Application configuration: adding a tab

DREAM

- Clone existing gadget:
  - html file, usually Input_viewXXXGadget.html
  - javascript file, usually Input_viewXXXGadget.js
- Make sure html file reference the javascript properly
- Debugging tips:
  - code you edit is in dream/platform/src/
  - grunt compiles code in dream/platform/static
  - if syntax error, files in static are not updated !
  - static files have different lines numbers
- use console.log (better with firebug)
- http://learn.renderjs.org/docs/index.html

SEVENTH FRAMEWORK PROGRAMME

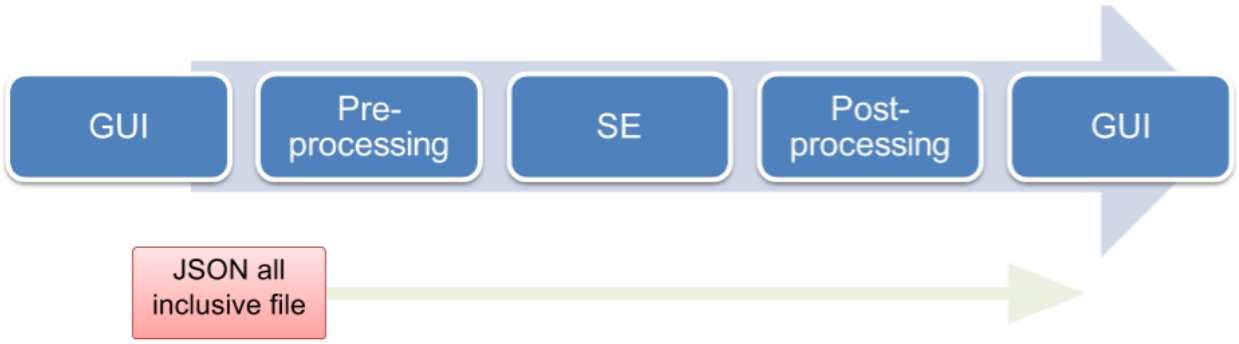
nexedi

Pre/Post Processing plugins

DREAM

JSON data goes through :

- pre processing plugins
- simulation engine
- post processing plugins



Pre/Post Processing plugins

DREAM

Plugins configured in JSON, under *application_configuration*

```
{
  "application_configuration": {
    "general": {↔},
    "input": {↔},
    "output": {↔},
    "post_processing": {
      "description": "",
      "plugin_list": [
        {
          "_class": "dream.plugins.DefaultTabularExit.DefaultTabularExit"
        },
        {
          "_class": "dream.plugins.PostProcessStationUtilization.PostProcessStationUtilization",
          "family": "Server",
          "output_id": "station_utilization"
        },
        {
          "_class": "dream.plugins.PostProcessQueueStatistics.PostProcessQueueStatistics",
          "output_id": "queue_statistics"
        }
      ]
    },
    "pre_processing": {
```

Python class for plugin

Parameters for the plugin

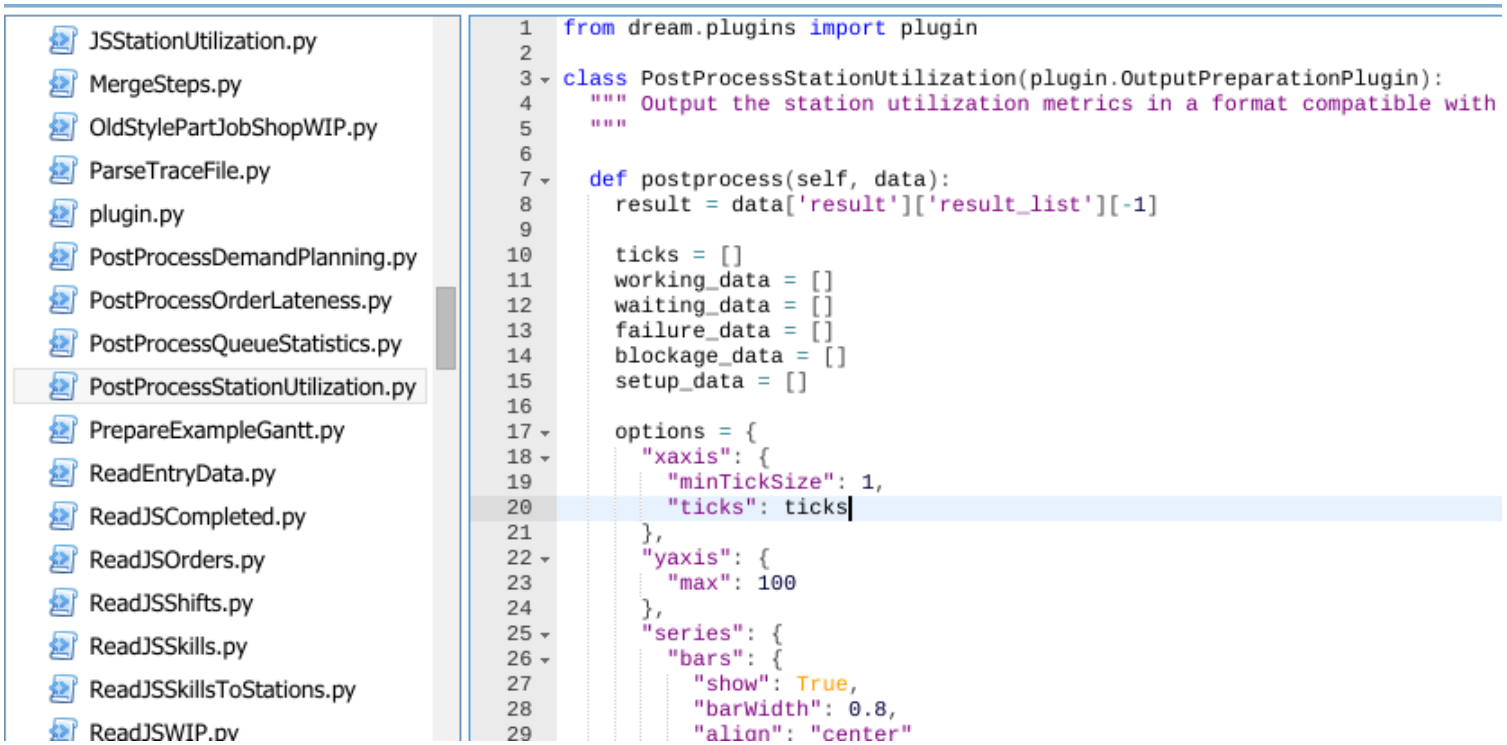
SEVENTH FRAMEWORK PROGRAMME

nexedi

Pre/Post Processing plugins

DREAM

Plugins are python class, interface is postprocess(self, data) or preprocess(self, data)



SEVENTH FRAMEWORK PROGRAMME

nexedi

Pre/Post Processing plugins

DREAM

Plugin configuration defined in json is in self.configuration_dict

```
49        "label": "Setup",
50        "data": setup_data
51      }];
52
53      out = result[self.configuration_dict['output_id']] = {
54        "series": series,
55        "options": options
56      }
57
58      i = 0
59      for obj in result['elementList']:
60        if obj.get('family') == self.configuration_dict.get('family'):
61          if obj['results']['working_ratio']:
62            working_data.append((i, obj['results']['working_ratio'][0]))
63          if obj['results']['waiting_ratio']:
64            waiting_data.append((i, obj['results']['waiting_ratio'][0]))
65          if obj['results']['failure_ratio']:
66            failure_data.append((i, obj['results']['failure_ratio'][0]))
67          if obj['results']['blockage_ratio']:
68            blockage_data.append((i, obj['results']['blockage_ratio'][0]))
69          if obj['results']['setup_ratio']:
70            setup_data.append((i, obj['results']['setup_ratio'][0]))
71
72          ticks.append((i, obj.get('name', obj['id'])))
73          i += 1
74
75      return data
```

Pre/Post Processing plugins

DREAM

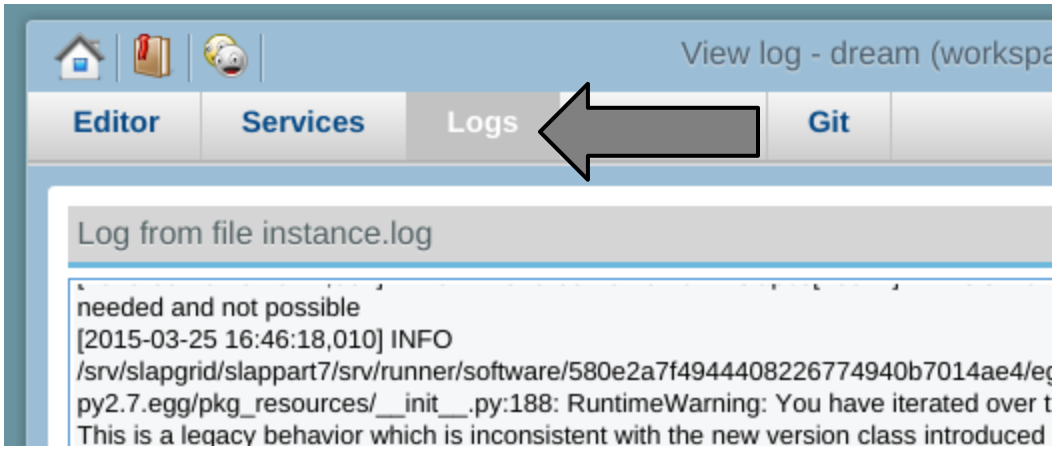
Plugins debugging tips:

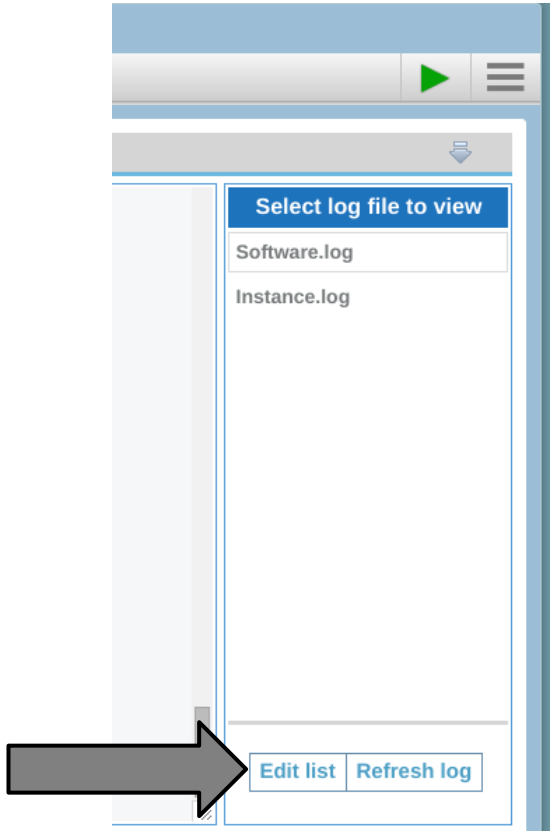
- Use existing Knowledge Extraction Tool objects (from dream/KnowledgeExtraction/ folder)
- Use dream.plugin.TimeSupport to convert from and to simulation clock time
- Log using self.logger.info (standard python logger https://docs.python.org/2/library/logging.html)
- Use dream.plugin.Debug in the chain to dump the json
- Write unit tests (examples in dream/tests/testGUIModels.py)

Accessing the application logs

DREAM

Add a file in log panel (1/2)

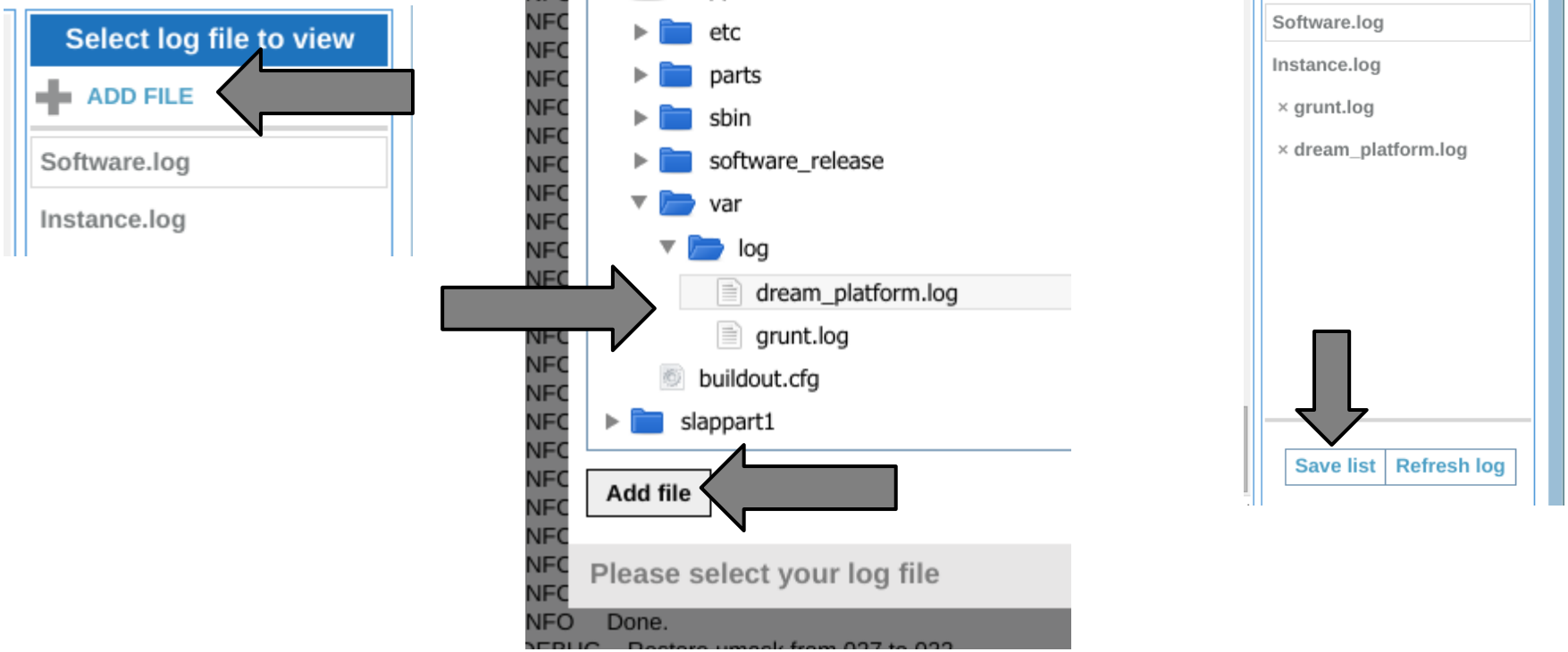




Accessing the application logs

DREAM

Add a file in log panel (2/2)

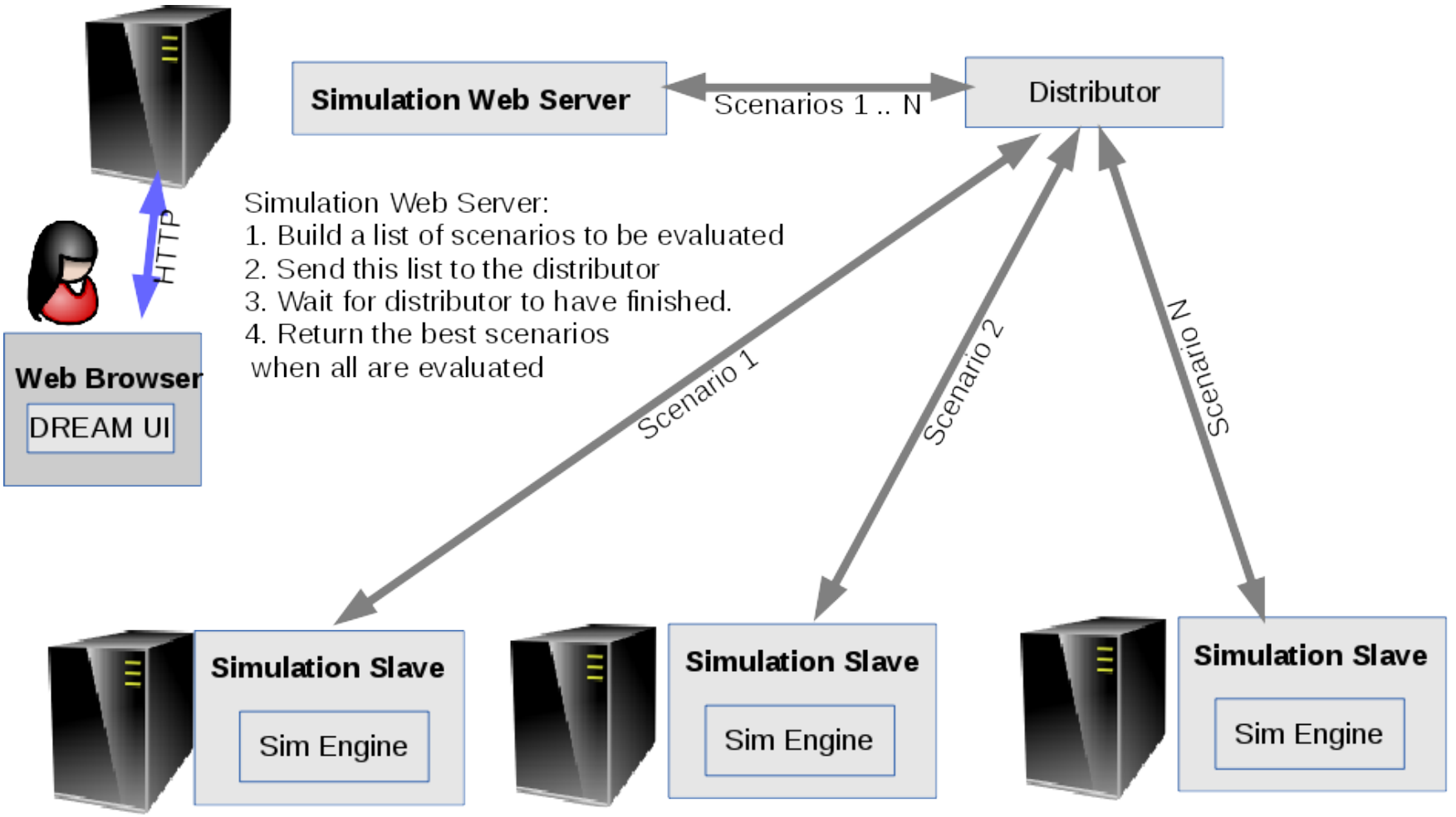


SEVENTH FRAMEWORK PROGRAMME

nexedi

Cloud Execution

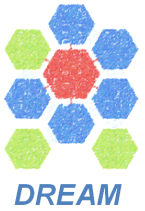
DREAM



SEVENTH FRAMEWORK PROGRAMME

nexedi

Cloud Execution



- Install DREAM distributor http://git.erp5.org/gitweb/slapos.git/blob_plain/refs/heads/dream:/software/erp5/software.cfg
- Note the custom frontend URL for later reference
- Install some DREAM simulation nodes http://git.erp5.org/gitweb/slapos.git/blob_plain/refs/heads/dream:/software/testnode/software.cfg set the distributor URL as “test-suite-master-url” parameter. Or using the following XML Parameter for web runner:

```
<instance>
  <parameter id="parameter-test-node-title">Simulation Node 1</parameter>
<parameter id="parameter-test-suite-master-url">https://login:
password@softinstXXX.host.vifib.
net/erp5/portal_task_distribution/dream_distributor</parameter>
  <parameter id="slapos-software">software/erp5testnode</parameter>
  <parameter id="slapos-reference">dream</parameter>
</instance>
```


Cloud Execution

DREAM

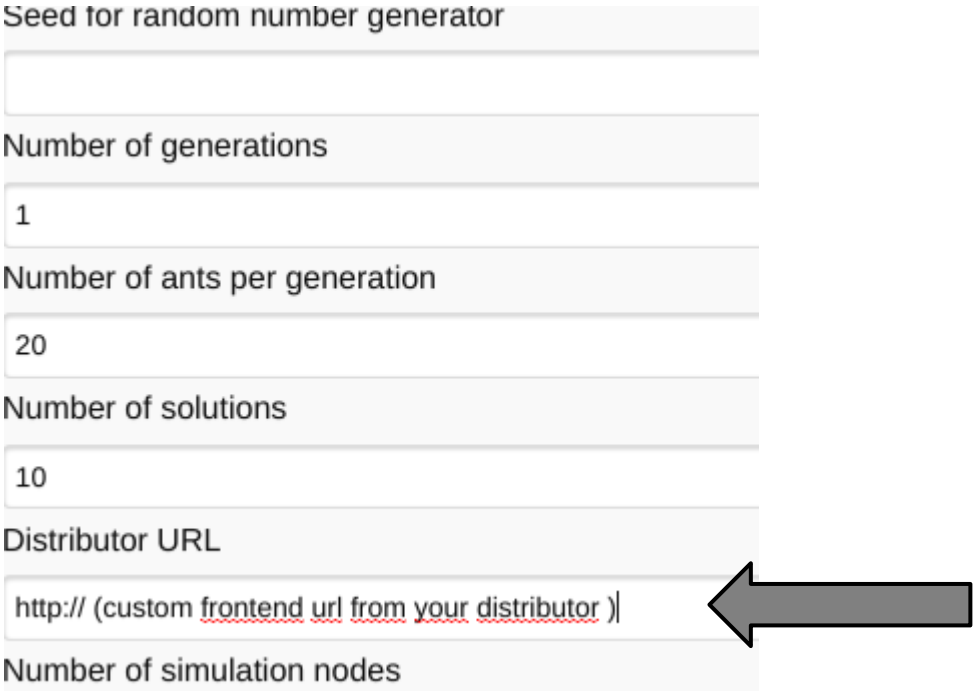
Be sure to use dream.plugin.ACO.ACO as processing plugin class in your model

```
763 ▸   "output": {↔},
839 ▾   "pre_processing": {
840 ▾     "plugin_list": [
841 ▾       {
842           "_class": "dream.plugins.Debug.Debug",
843           "argument": "Argument Value"
844         },
845 ▾       {
846           "_class": "dream.plugins.OldStylePartJobShopWIP.OldStylePartJobShopWIP",
847           "input_id": "old_style_part_jobshop_spreadsheet"
848         }
849       ]
850   },
851 ▾   "processing_plugin": {
852     "_class": "dream.plugins.ACO.ACO"
853   },
854 ▾   "post_processing": {
855     "plugin_list": []
856   },
857 ▾   "general": {
858 ▾     "properties": {
859 ▾       "numberOfReplications": {
```

Cloud Execution

DREAM

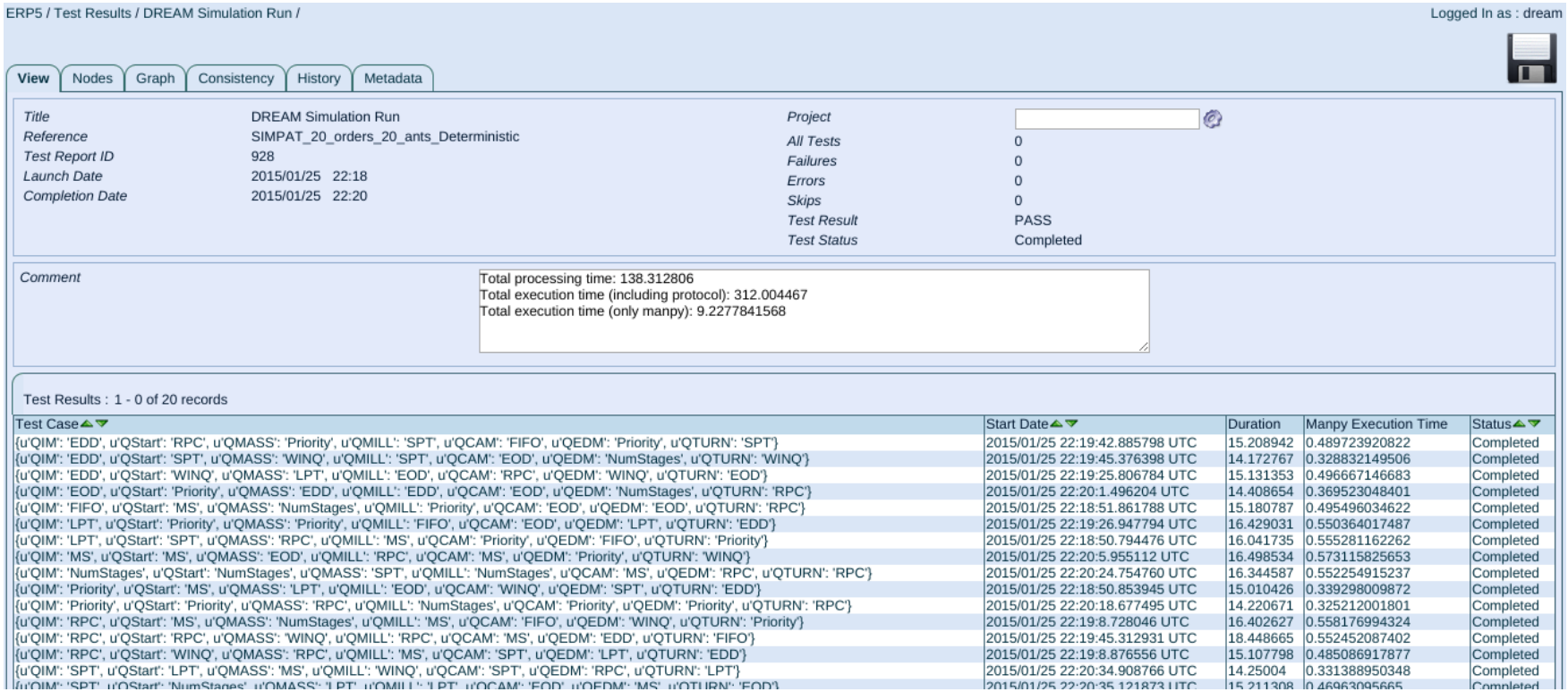
Be sure to specify the distributor URL in general properties



Cloud Execution

DREAM

View progress in “Test Result Module” of the distributor



ERP5 / Test Results / DREAM Simulation Run /

Logged In as : dream

View | Nodes | Graph | Consistency | History | Metadata

Title	DREAM Simulation Run	Project	
Reference	SIMPAT_20_orders_20_ants_Deterministic	All Tests	0
Test Report ID	928	Failures	0
Launch Date	2015/01/25 22:18	Errors	0
Completion Date	2015/01/25 22:20	Skips	0
		Test Result	PASS
		Test Status	Completed

Comment

Total processing time: 138.312806
Total execution time (including protocol): 312.004467
Total execution time (only manpy): 9.2277841568

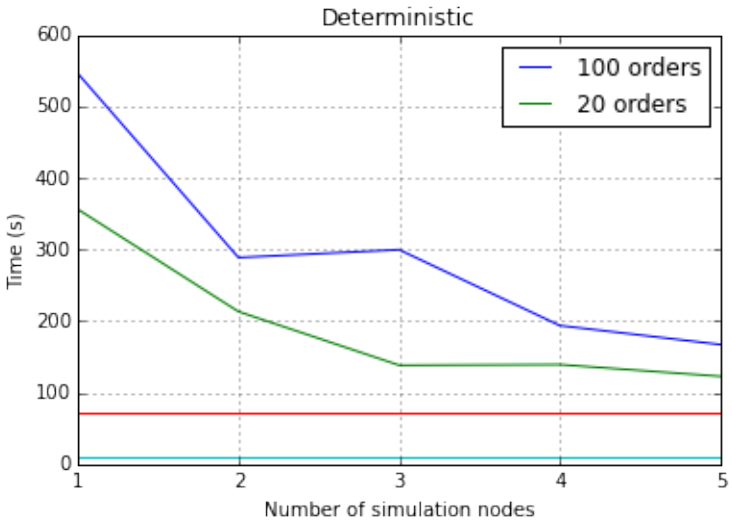
Test Results : 1 - 0 of 20 records

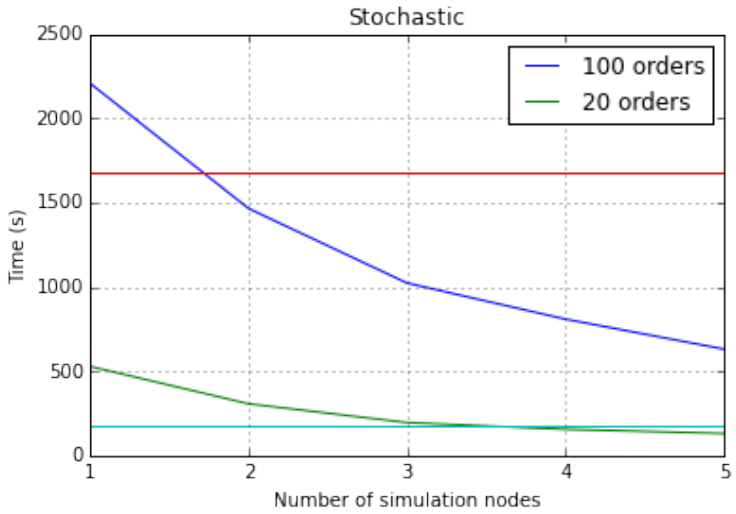
Test Case	Start Date	Duration	Manpy Execution Time	Status
{u'QIM': 'EDD', u'QStart': 'RPC', u'QMASS': 'Priority', u'QMILL': 'SPT', u'QCAM': 'FIFO', u'QEDM': 'Priority', u'QTURN': 'SPT'}	2015/01/25 22:19:42.885798 UTC	15.208942	0.489723920822	Completed
{u'QIM': 'EDD', u'QStart': 'SPT', u'QMASS': 'WINQ', u'QMILL': 'SPT', u'QCAM': 'EOD', u'QEDM': 'NumStages', u'QTURN': 'WINQ'}	2015/01/25 22:19:45.376398 UTC	14.172767	0.328832149506	Completed
{u'QIM': 'EDD', u'QStart': 'WINQ', u'QMASS': 'LPT', u'QMILL': 'EOD', u'QCAM': 'RPC', u'QEDM': 'WINQ', u'QTURN': 'EOD'}	2015/01/25 22:19:25.806784 UTC	15.131353	0.496667146683	Completed
{u'QIM': 'EOD', u'QStart': 'Priority', u'QMASS': 'EDD', u'QMILL': 'EDD', u'QCAM': 'EOD', u'QEDM': 'NumStages', u'QTURN': 'RPC'}	2015/01/25 22:20:1.496204 UTC	14.408654	0.369523048401	Completed
{u'QIM': 'FIFO', u'QStart': 'MS', u'QMASS': 'NumStages', u'QMILL': 'Priority', u'QCAM': 'EOD', u'QEDM': 'EOD', u'QTURN': 'RPC'}	2015/01/25 22:18:51.861788 UTC	15.180787	0.495496034622	Completed
{u'QIM': 'LPT', u'QStart': 'Priority', u'QMASS': 'Priority', u'QMILL': 'FIFO', u'QCAM': 'EOD', u'QEDM': 'LPT', u'QTURN': 'EDD'}	2015/01/25 22:19:26.947794 UTC	16.429031	0.550364017487	Completed
{u'QIM': 'LPT', u'QStart': 'SPT', u'QMASS': 'RPC', u'QMILL': 'MS', u'QCAM': 'Priority', u'QEDM': 'FIFO', u'QTURN': 'Priority'}	2015/01/25 22:18:50.794476 UTC	16.041735	0.555281162262	Completed
{u'QIM': 'MS', u'QStart': 'MS', u'QMASS': 'EOD', u'QMILL': 'RPC', u'QCAM': 'MS', u'QEDM': 'Priority', u'QTURN': 'WINQ'}	2015/01/25 22:20:5.955112 UTC	16.498534	0.573115825653	Completed
{u'QIM': 'NumStages', u'QStart': 'NumStages', u'QMASS': 'SPT', u'QMILL': 'NumStages', u'QCAM': 'MS', u'QEDM': 'RPC', u'QTURN': 'RPC'}	2015/01/25 22:20:24.754760 UTC	16.344587	0.552254915237	Completed
{u'QIM': 'Priority', u'QStart': 'MS', u'QMASS': 'LPT', u'QMILL': 'EOD', u'QCAM': 'WINQ', u'QEDM': 'SPT', u'QTURN': 'EDD'}	2015/01/25 22:18:50.853945 UTC	15.010426	0.339298009872	Completed
{u'QIM': 'Priority', u'QStart': 'Priority', u'QMASS': 'RPC', u'QMILL': 'NumStages', u'QCAM': 'Priority', u'QEDM': 'Priority', u'QTURN': 'RPC'}	2015/01/25 22:20:18.677495 UTC	14.220671	0.325212001801	Completed
{u'QIM': 'RPC', u'QStart': 'MS', u'QMASS': 'NumStages', u'QMILL': 'MS', u'QCAM': 'FIFO', u'QEDM': 'WINQ', u'QTURN': 'Priority'}	2015/01/25 22:19:8.728046 UTC	16.402627	0.558176994324	Completed
{u'QIM': 'RPC', u'QStart': 'RPC', u'QMASS': 'WINQ', u'QMILL': 'RPC', u'QCAM': 'MS', u'QEDM': 'EDD', u'QTURN': 'FIFO'}	2015/01/25 22:19:45.312931 UTC	18.448665	0.552452087402	Completed
{u'QIM': 'RPC', u'QStart': 'WINQ', u'QMASS': 'RPC', u'QMILL': 'MS', u'QCAM': 'SPT', u'QEDM': 'LPT', u'QTURN': 'EDD'}	2015/01/25 22:19:8.876556 UTC	15.107798	0.485086917877	Completed
{u'QIM': 'SPT', u'QStart': 'LPT', u'QMASS': 'MS', u'QMILL': 'WINQ', u'QCAM': 'SPT', u'QEDM': 'RPC', u'QTURN': 'LPT'}	2015/01/25 22:20:34.908766 UTC	14.25004	0.331388950348	Completed
{u'QIM': 'SPT', u'QStart': 'NumStages', u'QMASS': 'LPT', u'QMILL': 'LPT', u'QCAM': 'EOD', u'QEDM': 'MS', u'QTURN': 'EOD'}	2015/01/25 22:20:35.121873 UTC	15.211308	0.46963095665	Completed

Cloud Execution

DREAM

Cloud execution only has benefit for long running scenarios





SEVENTH FRAMEWORK PROGRAMME

nexedi