

Simulation using REST API

It is possible to simulate Teamwork Cloud projects on the server by using REST API. The topics below describe all available REST API requests and provide examples of how to use them.

**Prerequisites**
Before starting the simulation, make sure you have [prepared your projects for server-side simulation](#).

**REST API documentation**
For more information about simulation-related REST APIs, see our [REST API documentation](#) in Swagger. There you can find REST API descriptions in both machine-readable and human-readable formats.

You can use the following REST API requests to simulate Teamwork Cloud projects on the server: server-side simulation functionality. You can find it at <https://pypi.org/project/simulation-api/>

- [POST /simulation/api/projects/index.html](#).
- [POST /simulation/api/run/project/<project>/version/<version>/branch/<branch>/element/<element_id>/config/<config>](#)
- [POST simulation/api/step/<simulation_id>](#)
- [POST /simulation/api/start/<simulation_id>](#)
- [GET /simulation/api/status/<simulation_id>](#)
- [POST simulation/api/get/<simulation_id>](#)
- [POST simulation/api/set/<simulation_id>](#)
- [POST /simulation/api/pause/<simulation_id>](#)
- [POST /simulation/api/resume/<simulation_id>](#)
- [GET /simulation/api/result/<simulation_id>](#)
- [GET /simulation/api/running](#)
- [POST /simulation/api/terminate/<simulation_id>](#)
- [GET /simulation/api/hasConfigs/project/<project>/version/<version>/branch/<branch>](#)
- [GET /simulation/api/configs/project/<project>/version/<version>/branch/<branch>/element/<element_id>](#)
- [GET simulation/api/descriptor/project/<project>/version/<version>/branch/<branch>/config/<config>](#)

Get the list of Teamwork Cloud projects

GET /simulation/api/projects

This REST API endpoint provides the list of Teamwork Cloud projects accessible to you depending on your permissions. Only projects that you can read (the Read Resources permission) are returned.

Request and response examples

/simulation/api/projects

[{'id': '1f9fd988-620b-4b3c-8414-7ab96dc53a48',
 'name': 'SpacecraftMassRollup',
 'description': ''},
{'id': 'c125b91e-7d31-4015-b659-6fd98059b8c7',
 'name': 'BouncingBall',
 'description': 'This project demonstrates the execution of the BouncingBall, including fmu.'},
{'id': '485dfc82-7b08-43cd-86b1-953b3725e5b1',
 'name': 'CarBrakingAnalysis',
 'description': 'This project calculates the stopping distance based on car speed and mass.'}]

Run simulation

POST /simulation/api/run/project/<project>/version/<version>/branch/<branch>/element/<element_id>/config/<config>

This REST API request initiates the execution of the specified Simulation Configuration in a particular project. It connects to Teamwork Cloud, finds the element to execute, and initiates the execution if the element exists. After the execution is complete, the request constructs and returns a unique ID (per application) for the given execution.

The following table describes the parameters used in the REST API request:

Parameter	In	Required or optional	Description
project	path	required	The Teamwork Cloud project name or ID.
version	path	optional	The Teamwork Cloud project version.
branch	path	optional	The project branch name or ID. If the branch is omitted, the trunk is used instead.
element_id	path	optional	The server ID of the Instance Specification to be executed.
config	path	required	The Simulation Configuration name or server ID.
?autoStart	query	optional	Starts the simulation automatically after the initialization phase is completed. Available values are: ◦ <i>true</i> (default) - performs the initialization phase and then starts the simulation (the simulation clock is started). ◦ <i>false</i> - only the initialization phase of the simulation is performed. The simulation then waits for an invocation to start the simulation, perform a step, or terminate the simulation.  The ?autoStart query parameter is ignored when multiple iterations are running, e.g. Tradestudy, Montecarlo, or table simulation
?commitResults	query	optional	Commits a new project version with the simulation results. Available values are <i>true</i> or <i>false</i> (default).
?verification	query	optional	Returns the selected verification results. Available values are: ◦ all - all verification results are returned. ◦ none - no verification results are returned. ◦ fail (default) - verification results with the fail status are returned.
?startedFrom	query	optional	Allows specifying the starting location of the simulation. The value of the startedFrom query parameter is provided in the status endpoint response.
inputs	body	optional	The set of input parameters with values that should be provided for the simulation.  Input parameters can be the parameters with the data type, primitive type, or enumeration type.
outputs	body	optional	The set of output parameters that should be obtained after the simulation is complete. If no output parameters are specified, all initialized values are returned.  Output parameters can be the parameters with the data type, primitive type, or enumeration type.

Request and response examples

```
/simulation/api/run/project/SpacecraftMassRollup/branch/2/element/519701b0-bec8-4f51-aad5-152a5411d60b/config/spacecraft mass analysis?resultAsJson=true
```

```
{"state": "INITIALIZING",
"simulationId": "c7944233-a1cb-4310-a62f-48bce07dd71c",
"elapsedTime": 1591}
```

```
/simulation/api/run/project/CarBrakingAnalysis/config/Vehicle Analysis no Matlab
```

```
{"state": "RUNNING",
"simulationId": "cf83877b-86cc-466c-89a1-af97619a9a86",
"elapsedTime": 1821}
```

```
/simulation/api/run/project/8715fcfa-fd34-4928-8480-13f4439cec3d/element/519701b0-bec8-4f51-aad5-152a5411d60b/config/7915fcfa-fd88-4910-8560-13f4439cec3d
```

```
{"state": "QUEUED",
"simulationId": "b7bdf933-f58d-4e7e-b73b-8370c60485cd",
"queueNumber": 1,
"elapsedTime": 0}
```

Request example with a request body

```
/simulation/api/run/project/SpacecraftMassRollup/config/spacecraft mass analysis
```

Request body:

```
{
  "inputs":
  {
    "propulsion.tank.me": 38,
    "propulsion.thruster.me": 30,
    "telecom.antenna.me": 19,
    "telecom.amplifier.me": 8
  },
  "outputs":
  [
    "propulsion.me",
    "telecom.me"
  ]
}
```

Response:

```
{"state": "RUNNING",
"simulationId": "b7bdf933-f58d-4e7e-b73b-8370c60485cd",
"elapsedTime": 1480}
```

Perform a time step

```
POST simulation/api/step/<simulation_id>
```

This REST API request performs a single time step of the specified simulation. To run the simulation by step, it should be run with the **?autoStart** query parameter set to *false*.

The following table describes the parameters used in the REST API request:

Parameter	In	Required or optional	Description
simulation_id	path	required	The ID of a specific simulation.

Start simulation

POST /simulation/api/start/<simulation_id>

This REST API request starts the execution of the specified simulation. If the initialization phase is still in progress, the start endpoint is memorized and sent when the initialization phase is completed.

The following table describes the parameters used in the REST API request:

Parameter	In	Required or optional	Description
simulation_id	path	required	The ID of a specific simulation.

Get simulation status

GET /simulation/api/status/<simulation_id>

This REST API request gets the status, simulation time, queued number, and elapsed time of a specific simulation. It returns the actual elapsed time of the simulation thread and the simulation state (INITIALIZED, INITIALIZING, READY, QUEUED, RUNNING, COMPLETED, TERMINATING, TERMINATED, ERROR). If the simulation is placed in a waiting line, the QUEUED state and queued number will be provided. The user ID is also returned if the user has the administrator role.

The following table describes the parameters used in the REST API request:

Parameter	In	Required or optional	Description
simulation_id	path	required	The ID of a specific simulation.

Request and response examples

/simulation/api/status/55089221-8673-4391-aa3e-a8e774ecfa48

```
{'state': 'RUNNING',
'simulationId': '55089221-8673-4391-aa3e-a8e774ecfa48',
'simulationTime': '0 ms',
'ui': '/simulation/api/ui/55089221-8673-4391-aa3e-a8e774ecfa48/CoffeeMachine.html',
'project': 'CoffeeMachine',
'config': 'Coffee Machine Web',
'elapsedTime': 6598}
```

/simulation/api/status/ce8c8215-0515-43fd-9d34-92d1d7a95d87

```
{'state': 'COMPLETED',
'simulationId': 'ce8c8215-0515-43fd-9d34-92d1d7a95d87',
'simulationTime': '3000 ms',
'project': 'SpacecraftMassRollup',
'config': 'spacecraft mass analysis',
'elapsedTime': 6598}
```

/simulation/api/status/b7bdf933-f58d-4e7e-b73b-8370c60485cd

```
{'state': 'QUEUED',
'queueNumber': 3,
'simulationId': 'b7bdf933-f58d-4e7e-b73b-8370c60485cd',
'simulationTime': '0 ms',
'project': 'CarBrakingAnalysis',
'elapsedTime': 52}
```

Get simulation variables

POST simulation/api/get/<simulation_id>

This REST API request returns a list of simulation variables during simulation.

The following table describes the parameters used in the REST API request:

Parameter	In	Required or optional	Description
simulation_id	path	required	The ID of the running simulation.
variables	body	optional	The set of simulation variables that should be obtained. If no variables are specified, the values of all simulation variables are returned.

Request and response examples

/simulation/api/get/ce8c8215-0515-43fd-9d34-92d1d7a95d87

```
{
  "variables":
  [
    "propulsion.tank.me",
    "propulsion.thruster.me",
    "telecom.antena.me",
    "telecom.amplifier.me"
  ]
}
```

Set simulation variables

POST simulation/api/set/<simulation_id>

This REST API request specifies values for a provided list of simulation variables during simulation.

The following table describes the parameters used in the REST API request:

Parameter	In	Required or optional	Description
simulation_id	path	required	The ID of the running simulation.
variables	body	required	The set of simulation variables with values that should be used during the current simulation.
context	body	optional	The context ID.
propertyPath	body	optional	The property path of the context specified using property names or IDs.

Request and response examples

/simulation/api/set/ce8c8215-0515-43fd-9d34-92d1d7a95d87

Request body:

```
{
  "variables":
  {
    "propulsion.tank.me": 38,
    "propulsion.thruster.me": 30,
    "telecom.antena.me": 19,
    "telecom.amplifier.me": 8
  }
}
```

Pause simulation

```
POST /simulation/api/pause/<simulation_id>
```

This REST API request pauses the execution of the specified simulation.

The following table describes the parameters used in the REST API request:

Parameter	In	Required or optional	Description
simulation_id	path	required	The ID of the running simulation.

Resume simulation

```
POST /simulation/api/resume/<simulation_id>
```

This REST API request resumes the execution of the specified simulation.

The following table describes the parameters used in the REST API request:

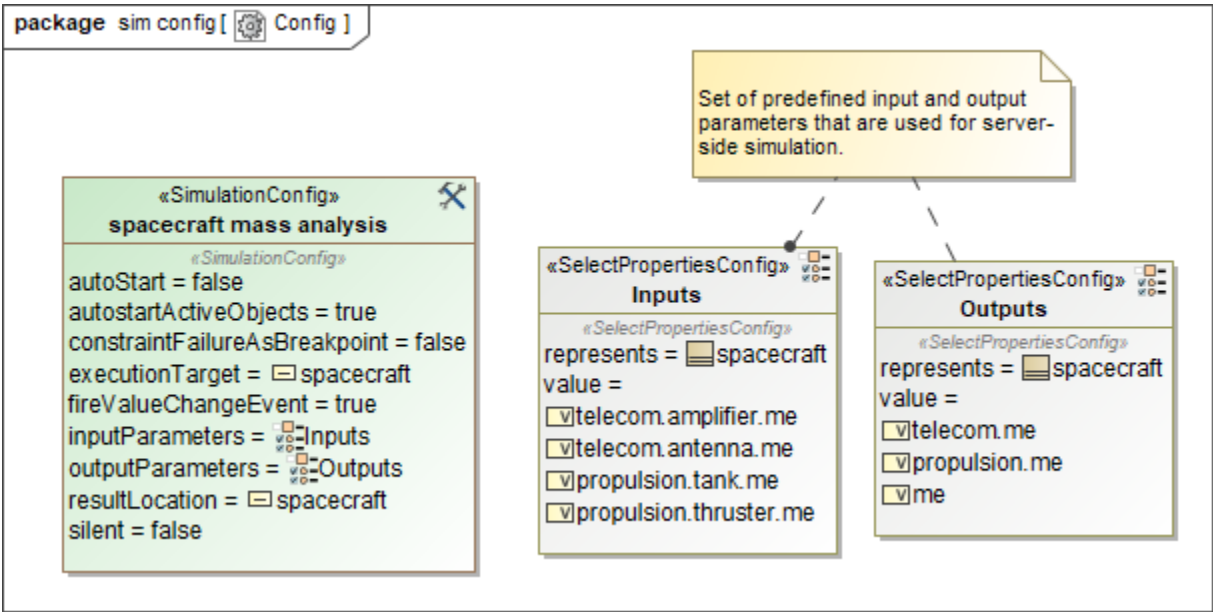
Parameter	In	Required or optional	Description
simulation_id	path	required	The ID of the running simulation.

Get simulation results

```
GET /simulation/api/result/<simulation_id>
```

This REST API request returns the results of the specified simulation. You can specify what output parameters should be returned in the run REST API body. If output parameters are not specified, all output parameters are obtained. The verification results are returned based on the value of the verification parameter of the Run Simulation endpoint. If the CSV export is specified for the Simulation Configuration, the CSV export results are returned in JSON format. If a Time Series or Timeline chart is specified for the Simulation Configuration, the chart data is returned in JSON format as well.

The figure below demonstrates how to specify input and output parameters.





- Output parameters can be the parameters with the data type, primitive type, or enumeration type.
- If the OutputParameters tag of the executed Simulation Configuration is specified, the results are provided according to the OutputParameters tag value.

The following table describes the parameters used in the REST API request:

Parameter	In	Required or optional	Description
simulation_id	path	required	The ID of the running simulation.

Request and response examples

/simulation/api/results/ce8c8215-0515-43fd-9d34-92d1d7a95d87

```
{
  "outputs": {
    "me": 98.0,
    "propulsion.me": 68.0,
    "propulsion.tank.me": 38.0,
    "propulsion.thruster.me": 30.0,
    "telecom.me": 30.0,
    "telecom.antenna.me": 10.0,
    "telecom.amplifier.me": 20.0
  },
  "csv exports": {
    "'SpaceCraftResults'": "me,propulsion.me,propulsion.tank.me,propulsion.thruster.me,telecom.me,telecom.amplifier.me,telecom.antenna.me\n98.0000,68.0000,38.0000,30.0000,30.0000,20.0000,10.0000\n98.0000,68.0000,38.0000,30.0000,30.0000,20.0000,10.0000\n98.0000,68.0000,38.0000,30.0000,30.0000,20.0000,10.0000\n"
  }
}
```

/simulation/api/results/309c8e47-4bdc-49be-b10a-aeacd9697584

```
{'outputs': {'requiredStoppingDistance': 49.09927575033121,
'stoppingDistance': 36.32807361129315},
'verification': [{'property': 'brakeLifeExpectancy',
'status': 'fail',
'value': 56413.07910758578,
'requirements': [{'id': '3',
'text': 'Brake pads shall have a projected life of at least 57500 kilometers under normal driving conditions, as per industry standard assumptions.',
'constraint': 'brakeLifeExpectancy>=57500',
'status': 'fail',
'margin': -1086.92089241422,
'timestamp': 0}]}],
{'property': 'wheel.brake.rotor.outerDiameter',
'status': 'fail',
'value': 0.25,
'requirements': [{'id': '6',
'text': 'The brake rotors shall have a 0.28 meter diameter.',
'constraint': 'outerDiameter==0.28',
'status': 'fail',
'margin': -0.03,
'timestamp': 0}]}]}
```

Get the list of running simulations

GET /simulation/api/running

This REST API request gets the list of all currently running simulations including the queued simulations that are placed in a waiting line.

Request and response examples

/simulation/api/running

```
{
  "running": [
    "f5c574a8-cd9c-4dbd-aae4-b1d18f048721",
    "ac1afe0a-c093-4eb1-865e-081c381d7f9f"
  ],
  "queued": [
    "ee99c51f-a8e8-40a4-a109-8f97aa579a33",
    "7c92a019-08b5-49bd-8c67-874613dcc1ae",
    "67840dbf-2071-4408-be76-8027e8a7aa48"
  ]
}
```

Terminate simulation

POST /simulation/api/terminate/<simulation_id>

This REST API request terminates the specified simulation.

The following table describes the parameters used in the REST API request:

Parameter	In	Required or optional	Description
simulation_id	path	required	The ID of a specific simulation.

Request and response examples

/simulation/api/terminate/4e35bc60-2c66-48e8-96e2-80f5270c08cf

/simulation/api/status/4e35bc60-2c66-48e8-96e2-80f5270c08cf

```
{
  "state": "TERMINATED",
  "elapsedTime": 11702
}
```

Check for Simulation Configurations

GET /simulation/api/hasConfigs/project/<project>/version/<version>/branch/<branch>

This REST API request checks if the project has any Simulation Configurations.

The following table describes the parameters used in the REST API request:

Parameter	In	Required or optional	Description
project	path	required	The Teamwork Cloud project name or ID.
version	path	optional	The Teamwork Cloud project version.
branch	path	optional	The project branch name or ID. If the branch is omitted, the trunk is used instead.
version	path	optional	The Teamwork Cloud project version.

Request and response examples

/simulation/api/hasConfigs/project/SpacecraftMassRollup


```
{
  "hasConfigs": true
}
```

Get Simulation Configurations

GET /simulation/api/configs/project/<project>/version/<version>/branch/<branch>/element/<element_id>

This REST API request retrieves element IDs, names, and descriptions of the Simulation Configurations available for the given executable element.

If the executable element is an Instance Specification, the method returns the following Simulation Configurations:

- Whose execution target is the classifier of the executable element.
- Whose execution target is the Instance Specification having at least one classifier matching those of the executable element.

The following table describes the parameters used in the REST API request:

Parameter	In	Required or optional	Description
project	path	required	The Teamwork Cloud project name or ID.
version	path	optional	The Teamwork Cloud project version.
branch	path	optional	The project branch name or ID. If the branch is omitted, the trunk is used instead.
element_id	path	optional	The server ID of the Instance Specification to be executed.
?filter	query	optional	Returns Simulations Configurations with a specified UI tag for a particular project. The only possible value for the filter parameter is ui . Only Simulations Configurations having at least one of the following UI elements are returned: • Frame • Table • TimeSeries chart

Request and response examples

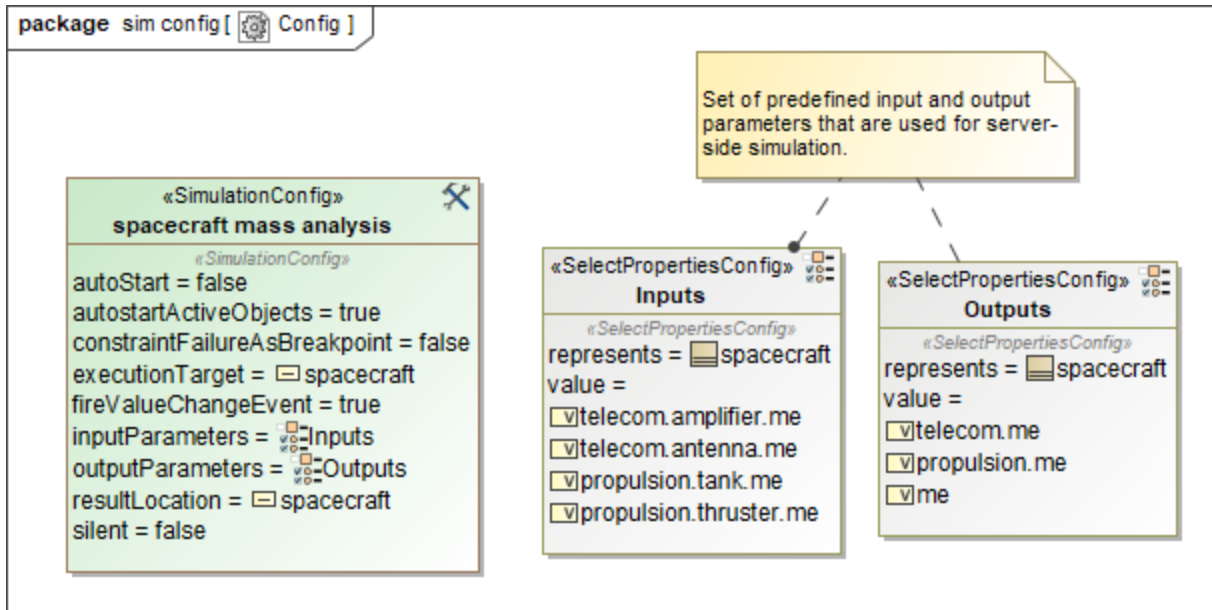
/simulation/api/configs/project/SpacecraftMassRollup

```
{
  "id": "7915fcfa-fd88-4910-8560-13f4439cec3d",
  "name": "spacecraft mass analysis",
  "description": ""
}
```

Get Simulation Configuration descriptor

GET simulation/api/descriptor/project/<project>/version/<version>/branch/<branch>/config/<config>

This REST API request retrieves Simulation Configuration data (description, execution target, time step, input and output parameters) from the specified project. The set of input and output parameters is specified using the **Input Parameters** and **Output Parameters** properties of a Simulation Configuration. The figure below demonstrates how to specify input and output parameters.



Specifying input and output parameters.

The following table describes the parameters used in the REST API request:

Parameter	In	Required or optional	Description
project	path	required	The Teamwork Cloud project name or ID.
version	path	optional	The Teamwork Cloud project version.
branch	path	optional	The project branch name or ID. If the branch is omitted, the trunk is used instead.
config	path	required	The Simulation Configuration name or server ID.

Request and response examples

simulation/api/descriptor/project/SpacecraftMassRollup/config/spacecraft mass analysis

```
{'config': 'spacecraft mass analysis',
'description': 'Simulation Config that is dedicated to run spacecraft mass rollup.',
'model': 'spacecraft',
'parameters': {'inputs': [{'parameter': 'telecom.amplifier.me',
'value': 32.0,
'type': 'mass[kilogram]',
'unit': 'kilogram',
'requirements': [{'id': '1',
'text': 'Estimated mass shall be less than allocated mass'}]},
{'parameter': 'telecom.antenna.me',
'value': 32.0,
'type': 'mass[kilogram]',
'unit': 'kilogram',
'requirements': [{'id': '1',
'text': 'Estimated mass shall be less than allocated mass'}]},
{'parameter': 'propulsion.tank.me',
'value': 68.0,
'type': 'mass[kilogram]',
'unit': 'kilogram',
'requirements': [{'id': '1',
'text': 'Estimated mass shall be less than allocated mass'}]},
{'parameter': 'propulsion.thruster.me',
'value': 68.0,
'type': 'mass[kilogram]',
'unit': 'kilogram',
'requirements': [{'id': '1',
'text': 'Estimated mass shall be less than allocated mass'}]}],
'outputs': [{'parameter': 'telecom.me',
```

```

    'value': 32.0,
    'type': 'mass[kilogram]',
    'unit': 'kilogram',
    'requirements': [{ 'id': '1',
      'text': 'Estimated mass shall be less than allocated mass' } ] },
  { 'parameter': 'propulsion.me',
    'value': 68.0,
    'type': 'mass[kilogram]',
    'unit': 'kilogram',
    'requirements': [{ 'id': '1',
      'text': 'Estimated mass shall be less than allocated mass' } ] },
  { 'parameter': 'me',
    'value': 95.0,
    'type': 'mass[kilogram]',
    'unit': 'kilogram',
    'requirements': [{ 'id': '1',
      'text': 'Estimated mass shall be less than allocated mass' } ] } ] } ] }

```

simulation/api/descriptor/project/BouncingBall/config/Run BouncingBall

```

{ 'config': 'Run BouncingBall',
  'description': 'Simulation Config dedicated to run simulation with bouncingBall.fmu.',
  'model': 'bouncingBall',
  'timeStep': '10 ms'
}

```