CYGNUS DOCUMENTATION

Version 0

Version 1 (beta)

version 0

retrieve admin login

`http://{{url}}:{{port-cygnus}}/admin/login`

Definition: Method to retrieve the logging level of Cygnus

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.

Return:

In case of performing the method correctly, returns the logging level of cygnus and "`200 OK`".

In case of wrong execution, return "`500 Internal Server Error`".

---

update admin login

`http://{{url}}:{{port-cygnus}}/admin/log?level={{fiware-logLevel}}`

Definition: Method to update the logging level of Cygnus for an admin

Method: PUT

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-logLevel: New level of log we want assigned. The valid loggins level are: `debug`, `info`, `warning`, `error` and `fatal`.

Return:

In case of performing the method correctly, returns "`200 OK`".

In case of wrong execution or invalid level, return "`400 Bad Request`".

---

create agent parameter

`http://{{url}}:{{port-cygnus}}/admin/configuration/agent/cygnus/apache-flume-1.4.0-bin/conf/{{fiware-agent}}?param={{fiware-param}}&value={{fiware-paramValue}}`

Definition: Method to create a new parameter in an agent configuration file.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-agent: Name of agent configuration file.
- fiware-param: Name of parameter we want to create.
- fiware-paramValue: Value of parameter we want to create.

Return:

In case of performing the method correctly, returns "`success: true`".

In case of wrong execution by invalid configuration file, existing value or invalid path to the agent configuration file, return "`success: false`".

retrieve agent parameter

```
http://{{url}}:{{port-cygnus}}/admin/configuration/agent/cygnus/apache-flume-1.4.0-bin/conf/{{fiware-agent}}&param={{fiware-param}}
```

Definition: Method to retrieve a defined parameter in an agent configuration file.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-agent: Name of agent configuration file.
- fiware-param: Name of parameter we want to retrieve.

Return:

In case of performing the method correctly, returns "`success: true`" with the attribute and it value.

In case of wrong execution by invalid configuration file, not found the parameter or invalid path to the agent configuration file, return "`success: false`".

retrieve agent parameters

```
http://{{url}}:{{port-cygnus}}/admin/configuration/agent/cygnus/apache-flume-1.4.0-bin/conf/{{fiware-agent}}
```

Definition: Method to retrieve all parameters in an agent configuration file.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-agent: Name of agent configuration file.

Return:

In case of performing the method correctly, returns "`success: true`" with the attributes and its values.

In case of wrong execution by invalid configuration file or invalid path to the agent configuration file, return "`success: false`".

update agent parameter

```
http://{{url}}:{{port-cygnus}}/admin/configuration/agent/cygnus/apache-flume-1.4.0-bin/conf/{{fiware-agent}}?param={{fiware-param}}&value={{fiware-paramValue}}
```

Definition: Method to create a new parameter in an agent configuration file.

Method: PUT

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-agent: Name of agent configuration file.
- fiware-param: Name of parameter we want to update.

- fiware-paramValue: Value of parameter we want to update.

Return:

In case of performing the method correctly, returns “`success: true`”.

In case of wrong execution by invalid configuration file, not found the parameter or invalid path to the agent configuration file, return “`success: false`”.

---

delete agent parameter

```
http://{{url}}:{{port-cygnus}}/admin/configuration/agent/cygnus/apache-flume-1.4.0-bin/conf/{{fiware-agent}}?param={{fiware-param}}
```

Definition: Method to delete a parameter in an agent configuration file.

Method: DELETE

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-agent: Name of agent configuration file.
- fiware-param: Name of parameter we want to delete.

Return:

In case of performing the method correctly, returns “`success: true`”.

In case of wrong execution by invalid configuration file, not found the attribute or invalid path to the agent configuration file, return “`success: false`”.

---

create instance parameter

```
http://{{url}}:{{port-cygnus}}/admin/configuration/instance/usr/cygnus/conf/{{fiware-instance}}?param={{fiware-param}}&value={{fiware-paramValue}}
```

Definition: Method to create a new parameter in an instance configuration file.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-instance: Name of instance configuration file.
- fiware-param: Name of parameter we want to create.
- fiware-paramValue: Value of parameter we want to create.

Return:

In case of performing the method correctly, returns “`success: true`”.

In case of wrong execution by invalid configuration file, existing value or invalid path to the instance configuration file, return “`success: false`”.

---

retrieve instance parameter

`http://{{url}}:{{port-cygnus}}/admin/configuration/instance/usr/cygnus/conf/{{fiware-instance}}?param={{fiware-param}}`

Definition: Method to retrieve a defined parameter in an instance configuration file.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-instance: Name of agent configuration file.
- fiware-param: Name of parameter we want to retrieve.

Return:

In case of performing the method correctly, returns “`success: true`” with the attribute and it value.

In case of wrong execution by invalid configuration file, not found the parameter or invalid path to the instance configuration file, return “`success: false`”.

---

retrieve instance parameters

`http://{{url}}:{{port-cygnus}}/admin/configuration/instance/usr/cygnus/conf/{{fiware-instance}}`

Definition: Method to retrieve all parameters in an instance configuration file.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-instance: Name of instance configuration file.

Return:

In case of performing the method correctly, returns “`success: true`” with the attributes and its values.

In case of wrong execution by invalid configuration file or invalid path to the instance configuration file, return “`success: false`”.

---

update instance parameter

`http://{{url}}:{{port-cygnus}}/admin/configuration/instance/usr/cygnus/conf/{{fiware-instance}}?param={{fiware-param}}&value={{fiware-paramValue}}`

Definition: Method to update a parameter in an instance configuration file.

Method: PUT

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-instance: Name of instance configuration file.
- fiware-param: Name of parameter we want to update.
- fiware-paramValue: Value of parameter we want to update.

Return:

In case of performing the method correctly, returns "`success: true`".

In case of wrong execution by invalid configuration file, not found the parameter or invalid path to the instance configuration file, return "`success: false`".

---

delete parameter instance

```
http://{{url}}:{{port-cygnus}}/admin/configuration/instance/usr/cygnus/
conf/{{fiware-instance}}?param={{fiware-param}}
```

Definition: Method to delete a parameter in an instance configuration file.

Method: DELETE

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-instance: Name of instance configuration file.
- fiware-param: Name of parameter we want to delete.

Return:

In case of performing the method correctly, returns "`success: true`".

In case of wrong execution by invalid configuration file, not found the attribute or invalid path to the instance configuration file, return "`success: false`".

version 1 (beta)

create agent parameter

```
http://{{url}}:{{port-
cygnus}}/v1/admin/configuration/agent/cygnus/apache-flume-1.4.0-
bin/conf/{{fiware-agent}}?param={{fiware-param}}&value={{fiware-
paramValue}}
```

Definition: Method to create a new parameter in an agent configuration file.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-agent: Name of agent configuration file.
- fiware-param: Name of parameter we want to create.
- fiware-paramValue: Value of parameter we want to create.

Return:

In case of performing the method correctly, returns "`success: true`".

In case of wrong execution by invalid configuration file, existing value or invalid path to the agent configuration file, return "`success: false`".

---

retrieve agent parameter

```
http://{{url}}:{{port-cygnus}}/v1/admin/configuration/agent/cygnus/apache-flume-1.4.0-bin/conf/{{fiware-agent}}&param={{fiware-param}}
```

Definition: Method to retrieve a defined parameter in an agent configuration file.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-agent: Name of agent configuration file.
- fiware-param: Name of parameter we want to retrieve.

Return:

In case of performing the method correctly, returns “`success: true`” with the attribute and it value.

In case of wrong execution by invalid configuration file, not found the parameter or invalid path to the agent configuration file, return “`success: false`”.

---

retrieve agent parameters

```
http://{{url}}:{{port-cygnus}}/v1/admin/configuration/agent/cygnus/apache-flume-1.4.0-bin/conf/{{fiware-agent}}
```

Definition: Method to retrieve all parameters in an agent configuration file.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-agent: Name of agent configuration file.

Return:

In case of performing the method correctly, returns “`success: true`” with the attributes and its values.

In case of wrong execution by invalid configuration file or invalid path to the agent configuration file, return “`success: false`”.

---

update agent parameter

```
http://{{url}}:{{port-cygnus}}/v1/admin/configuration/agent/cygnus/apache-flume-1.4.0-bin/conf/{{fiware-agent}}?param={{fiware-param}}&value={{fiware-paramValue}}
```

Definition: Method to create a new parameter in an agent configuration file.

Method: PUT

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-agent: Name of agent configuration file.

- fiware-param: Name of parameter we want to update.
- fiware-paramValue: Value of parameter we want to update.

Return:
In case of performing the method correctly, returns “`success: true`”.
In case of wrong execution by invalid configuration file, not found the parameter or invalid path to the agent configuration file, return “`success: false`”.

---

delete agent parameter

`http://{{url}}:{{port-cygnus}}/v1/admin/configuration/agent/cygnus/apache-flume-1.4.0-bin/conf/{{fiware-agent}}?param={{fiware-param}}`

Definition: Method to delete a parameter in an agent configuration file.
Method: DELETE
Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-agent: Name of agent configuration file.
- fiware-param: Name of parameter we want to delete.

Return:
In case of performing the method correctly, returns “`success: true`”.
In case of wrong execution by invalid configuration file, not found the attribute or invalid path to the agent configuration file, return “`success: false`”.

---

create instance parameter

`http://{{url}}:{{port-cygnus}}/v1/admin/configuration/instance/usr/cygnus/ conf/{{fiware-instance}}?param={{fiware-param}}&value={{fiware-paramValue}}`

Definition: Method to create a new parameter in an instance configuration file.
Method: POST
Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-instance: Name of instance configuration file.
- fiware-param: Name of parameter we want to create.
- fiware-paramValue: Value of parameter we want to create.

Return:
In case of performing the method correctly, returns “`success: true`”.
In case of wrong execution by invalid configuration file, existing value or invalid path to the instance configuration file, return “`success: false`”.

---

retrieve instance parameter

```
http://{{url}}:{{port-
cygnus}}/v1/admin/configuration/instance/usr/cygnus/ conf/{{fiware-
instance}}?param={{fiware-param}}
```

Definition: Method to retrieve a defined parameter in an instance configuration file.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-instance: Name of agent configuration file.
- fiware-param: Name of parameter we want to retrieve.

Return:

In case of performing the method correctly, returns “`success: true`” with the attribute and it value.

In case of wrong execution by invalid configuration file, not found the parameter or invalid path to the instance configuration file, return “`success: false`”.

---

retrieve instance parameters

```
http://{{url}}:{{port-
cygnus}}/v1/admin/configuration/instance/usr/cygnus/ conf/{{fiware-
instance}}
```

Definition: Method to retrieve all parameters in an instance configuration file.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-instance: Name of instance configuration file.

Return:

In case of performing the method correctly, returns “`success: true`” with the attributes and its values.

In case of wrong execution by invalid configuration file or invalid path to the instance configuration file, return “`success: false`”.

---

update instance parameter

```
http://{{url}}:{{port-
cygnus}}/v1/admin/configuration/instance/usr/cygnus/ conf/{{fiware-
instance}}?param={{fiware-param}}&value={{fiware-paramValue}}
```

Definition: Method to update a parameter in an instance configuration file.

Method: PUT

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-instance: Name of instance configuration file.
- fiware-param: Name of parameter we want to update.

- fiware-paramValue: Value of parameter we want to update.

Return:
In case of performing the method correctly, returns "`success: true`".
In case of wrong execution by invalid configuration file, not found the parameter or invalid path to the instance configuration file, return "`success: false`".

---

delete parameter instance

`http://{{url}}:{{port-cygnus}}/v1/admin/configuration/instance/usr/cygnus/ conf/{{fiware-instance}}?param={{fiware-param}}`

Definition: Method to delete a parameter in an instance configuration file.
Method: DELETE
Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-instance: Name of instance configuration file.
- fiware-param: Name of parameter we want to delete.

Return:
In case of performing the method correctly, returns "`success: true`".
In case of wrong execution by invalid configuration file, not found the attribute or invalid path to the instance configuration file, return "`success: false`".

version

retrieve version

`http://{{url}}:{{port-cygnus}}/v1/version`

Definition: Method to retrieve the version of the running software.
Method: GET
Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.

Return:
In case of performing the method correctly, returns "`success: true`" and the version.
In case of wrong execution return "`success: false`".

grouping rules

create grouping rules

`http://{{url}}:{{port-cygnus}}/v1/groupingrules`

Definition: Method to create a new grouping rules, to join related data.
Method: POST
Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.

Body:

```
        {
            "regex": "SmartSpot",
            "destination": "othersmartspots",
            "fiware_service_path": "smartspot",
            "fields": ["entityType"]
        }
```

Return:

In case of performing the method correctly, returns “`success: true`”.
In case of wrong execution return “`success: false`”.

---

retrieve grouping rules

`http://{{url}}:{{port-cygnus}}/v1/groupingrules`

Definition: Method to retrieve all grouping rules.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.

Return:

In case of performing the method correctly, returns:

```
{
    "grouping_rules": [
        {
            "destination": "group_id",
            "fields": [
                "entityType"
            ],
            "fiware_service_path": "service_path",
            "id": 1,
            "regex": "entity_id"
        }
     ],
    "success": "true"
}
```

In case of wrong execution return “`success: false`”.

---

update grouping rules

`http://{{url}}:{{port-cygnus}}/v1/groupingrules?id={{fiware-groupID}}`

Definition: Method to update a grouping rules.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-groupID: Name of group we want to update.

Body:

```
        {
            "regex": "SmartSpot",
            "destination": "othersmartspots",
            "fiware_service_path": "smartspot",
            "fields": ["entityType"]
        }
```

Return:

In case of performing the method correctly, returns “`success: true`”.

In case of wrong execution return “`success: false`”.

---

delete grouping rules

`http://{{url}}:{{port-cygnus}}/v1/groupingrules?id={{fiware-groupID}}`

Definition: Method to delete a grouping rules defined as a parameter.

Method: DELETE

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-groupID: Name of group we want to delete.

Return:

In case of performing the method correctly, returns “`success: true`”.

In case of wrong execution return “`success: false`”.

subcriptions

create subscription for Orion v1

`http://{{url}}:{{port-cygnus}}/v1/subscriptions?id=ngsi_version=1`

Definition: Method to create a subscription to one or many entities in Onion.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.

Body:

```
{
    "subscription":{
          "entities": [
              {
                  "type": "entity_type",
                  "isPattern": "false",
                  "id": "entity_id"
              }
          ],
          "attributes": [],
          "reference": "http://reference_host:reference_port",
          "duration": "P1M",
          "notifyConditions": [
              {
                  "type": "ONCHANGE",
                  "condValues": []
              }
          ],
          "throttling": "PT5S"
    },
    "endpoint":{
          "host":"endpoint_host",
          "port":"endpoint_port",
          "ssl":"false",
          "xauthtoken":"234123123123123"
    }
}
```

- isPattern: Currently is hasn't use. Always 'false'.
- id: To which entity it wants to subscribe. In this case, to all entities of those type.
- reference: URL of client that it want to subscribe.
- duration: The duration of the subscription. In ISO 8601 standard format.
- notifyConditions: Define the launcher to notify the subscriptions. In this case, when change a value of attributes passed in the point "condValues" of an entity will launch a notification.
- throttling: Specify a minimum inter-notification arrival time. In this case, 5 seconds.
- endpoint: URL of Orion server.

Return:

In case of performing the method correctly, returns "success: true" with information about the subscription (duration, throttling, subscriptionId).
In case of wrong execution return "success: false".

---

create subscription for Orion v2

http://{{url}}:{{port-cygnus}}/v1/subscriptions?id=ngsi_version=2

Definition: Method to create a subscription to one or many entities in Onion.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.

Body:

```
{
    "subscription":{
        "description": "subscription_description",
        "subject": {
            "entities": [
                {
                    "idPattern": ".*",
                    "type": "entity_type"
                }
            ],
            "condition": {
                "attrs": [
                    "attribute_entity"
                ],
                "expression": {
                      "q": "attribute_entity>40"
                }
            }
        },
        "notification": {
            "http": {
                "url": "http://localhost:1234"
            },
            "attrs": [
                "attribute_entity"
            ]
        },
        "expires": "2016-05-05T14:00:00.00Z",
        "throttling": 5
    },
    "endpoint":{
        "host":"<endpoint_host>",
        "port":"<endpoint_port>",
        "ssl":"false",
        "xauthtoken":"QsENv67AJj7blC2qJ0YvfSc5hMWYrs"
    }
}
```

- type: To which entity it wants to subscribe. In this case, to all entities of those type.
- `condition:` Conditions that have to be fulfill for launch to notification.
- `expires:` The duration of the subscription, when it expires.
- `notification:` URL of client that it want to subscribe (cygnus_url) and the attributes that we want know.
- `throttling:` Specify a minimum inter-notification arrival time. In this case, 5 `seconds.`
- `endpoint:` URL of Orion server.

Return:

In case of performing the method correctly, returns “`success: true`” with information about the subscription (`duration, throttling, subscriptionId`).
In case of wrong execution return “`success: false`”.

retrieve subscription

`http://{{url}}:{{port-cygnus}}/v1/subscriptions?id=ngsi_version=2&subscription_id={{fiware-subscription}}`

Definition: Method to retrieve a subscription to one or many entities in Onion of cygnus.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- orion_version:

Return:

In case of performing the method correctly, returns “`success: true`” with information about the subscription.
In case of wrong execution by inexistent subscription id, invalid or not implemented NGSI version or missing parameters return “`success: false`”.

retrieve subscriptions

`http://{{url}}:{{port-cygnus}}/v1/subscriptions?id=ngsi_version=2`

Definition: Method to retrieve all subscriptions in Onion of cygnus.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- orion_version: version of Orion to do this method.

Return:

In case of performing the method correctly, returns "`success: true`" with information about all subscriptions.

In case of wrong execution by invalid or not implemented NGSI version or missing parameters return "`success: false`".

---

delete subscription

`http://{{url}}:{{port-cygnus}}/v1/subscriptions?id=ngsi_version={{orion_version}}&subscription_id={{subscription_id}}`

Definition: Method to delete a subscription in Onion of cygnus.

Method: DELETE

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- orion_version: version of Orion to do this method.
- subscription_id: id of subscription we want to delete.

Return:

In case of performing the method correctly, returns "`success: true, result: subscription deleted`".

In case of wrong execution by invalid or not implemented NGSI version, wrong subscription id, missing authentication token or missing fields return "`success: false`".

logs

create appender

`http://{{url}}:{{port-cygnus}}/v1/admin/log/appenders?transient={{fiware-transient}}`

Definition: Method to create an appender in a running logger.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-trasient: if `false` the appenders are retrieved from Cygnus, if `true` are retrieved from file.

Body:

```
{
    "appender": {
        "name":"appender_id",
        "class":"appender_class"
      },
      "pattern": {
        "layout":".....",
        "ConversionPattern":"....."
     }
  }
```

Return:

In case of performing the method correctly, returns “`success: true, result: Appender ‘appender_id’ put.`”

In case of wrong execution by invalid transient or without JSON return “`success: false`”.

---

retrieve appender

`http://{{url}}:{{port-cygnus}}/v1/admin/log/appenders?name={{fiware-appender}}&transient={{fiware-transient}}`

Definition: Method to retrieve a defined appender.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-trasient: if `false` the appenders are retrieved from Cygnus, if `true` are retrieved from file.
- fiware-appender: Name of appender we want to retrieve

Return:

In case of performing the method correctly, returns “`success: true`” join to appender information.

In case of wrong execution by invalid transient or invalid appender name return “`success: false`”.

---

retrieve appenders

`http://{{url}}:{{port-cygnus}}/v1/admin/log/appenders?transient={{fiware-transient}}`

Definition: Method to retrieve all appenders in a running logger.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-trasient: if `false` the appenders are retrieved from Cygnus, if `true` are retrieved from file.

Return:

In case of performing the method correctly, returns “`success: true`” join to the information about all appenders.

In case of wrong execution by invalid transient or appenders not be shown return “`success: false`”.

---

update appender

```
http://{{url}}:{{port-cygnus}}/v1/admin/log/appenders?transient={{fiware-transient}}
```

Definition: Method to update an appender in a running logger.

Method: PUT

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-trasient: if `false` the appenders are retrieved from Cygnus, if `true` are retrieved from file.

Body:

```
{
    "appender": {
        "name":"appender_id",
        "class":"appender_class"
      },
      "pattern": {
        "layout":".....",
        "ConversionPattern":"....."
     }
  }
```

Return:

In case of performing the method correctly, returns “`{"success": "true","result": "Appender 'appender_id' updated successfully”}`”

In case of wrong execution by invalid transient or without JSON return “`success: false`”.

---

delete appender

```
http://{{url}}:{{port-cygnus}}/v1/admin/log/appenders?name={{fiware-appender}}&transient={{fiware-transient}}
```

Definition: Method to delete a defined appender.

Method: DELETE

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-trasient: if `false` the appenders are retrieved from Cygnus, if `true` are retrieved from file.
- fiware-appender: Name of appender we want to delete

Return:
In case of performing the method correctly, returns “`success: true`”.
In case of wrong execution by invalid transient or invalid appender name return “`success: false`”.

---

delete appenders

```
http://{{url}}:{{port-cygnus}}/v1/admin/log/appenders?
transient={{fiware-transient}}
```

Definition: Method to delete all appenders.
Method: DELETE
Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-trasient: if `false` the appenders are retrieved from Cygnus, if `true` are retrieved from file.

Return:
In case of performing the method correctly, returns “`success: true`”.
In case of wrong execution by invalid transient return “`success: false`”.

---

create logger

```
http://{{url}}:{{port-cygnus}}/v1/admin/log/loggers?transient={{fiware-
transient}}
```

Definition: Method to create a logger in a running Cygnus.
Method: POST
Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-trasient: if `false` the loggers are retrieved from Cygnus, if `true` are retrieved from file.

Body:

```
{
    "logger": {
        "name":"logger_id",
        "level":"....."
    }
}
```

Return:
In case of performing the method correctly, returns “`success: true, result: logger ‘logger_id’ put.`”
In case of wrong execution by invalid transient or without JSON return “`success: false`”.

---

retrieve logger

`http://{{url}}:{{port-cygnus}}/v1/admin/log/loggers?name={{fiware-logger}}&transient={{fiware-transient}}`

Definition: Method to retrieve a defined logger.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-trasient: if `false` the logger are retrieved from Cygnus, if `true` are retrieved from file.
- fiware-logger: Name of logger we want to retrieve

Return:

In case of performing the method correctly, returns "`success: true`" join to logger information.

In case of wrong execution by invalid transient or invalid logger name return "`success: false`".

retrieve loggers

`http://{{url}}:{{port-cygnus}}/v1/admin/log/loggers?transient={{fiware-transient}}`

Definition: Method to retrieve all loggers in a running Cygnus.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-trasient: if `false` the appenders are retrieved from Cygnus, if `true` are retrieved from file.

Return:

In case of performing the method correctly, returns "`success: true`" join to the information about all loggers.

In case of wrong execution by invalid transient or loggers not be shown return "`success: false`".

update logger

`http://{{url}}:{{port-cygnus}}/v1/admin/log/loggers?transient={{fiware-transient}}`

Definition: Method to update a logger in a running Cygnus.

Method: PUT

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.

- fiware-trasient: if `false` the loggers are retrieved from Cygnus, if `true` are retrieved from file.

Body:

```
{
    "appender": {
        "name":"logger_id",
        "class":"logger_class"
      },
      "pattern": {
        "layout":".....",
        "ConversionPattern":"....."
     }
  }
```

Return:
In case of performing the method correctly, returns “`{"success": "true","result": "Logger 'logger_id' updated successfully”}`”
In case of wrong execution by invalid transient or without JSON return “`success: false`”.

---

delete logger

`http://{{url}}:{{port-cygnus}}/v1/admin/log/loggers?name={{fiware-logger}}&transient={{fiware-transient}}`
Definition: Method to delete a defined logger.
Method: DELETE
Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-trasient: if `false` the loggers are retrieved from Cygnus, if `true` are retrieved from file.
- fiware-appender: Name of logger we want to delete

Return:
In case of performing the method correctly, returns “`success: true`”.
In case of wrong execution by invalid transient or invalid logger name return “`success: false`”.

---

delete loggers

`http://{{url}}:{{port-cygnus}}/v1/admin/log/loggers?transient={{fiware-transient}}`
Definition: Method to delete all loggers.
Method: DELETE
Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.

- fiware-trasient: if `false` the loggers are retrieved from Cygnus, if `true` are retrieved from file.

Return:
In case of performing the method correctly, returns “`success: true`”.
In case of wrong execution by invalid transient return “`success: false`”.

Metrics

The metrics defined in a Cygnus agent are:

- **incomingTransactions**. Number of incoming transactions. In other words, number of NGSI notifications received.
- **incomingTransactionRequestSize**. Total size of the requests related to incoming transactions (bytes).
- **incomingTransactionResponseSize**. Total size of the responses related to incoming transactions (bytes).
- **incomingTransactionError**. Number of incoming transactions causing an error.
- serviceTime. Average time between transaction requests reception and transaction responses sending.
- **outgoingTransactions**. Number of outgoing transactions. In other words, number of persistence operations.
- **outgoingTransactionRequestSize**. Total size of the requests related to outgoing transactions (bytes).
- **outgoingTransactionResponseSize**. Total size of the responses related to outgoing transactions (bytes).
- **outgoingTransactionError**. Number of outgoing transactions causing an error.

retrieve metrics

```
http://{{url}}:{{port-cygnus}}/v1/admin/metrics?reset={{fiware-reset}}
```

Definition: Method to retrieve all metrics for a Cygnus agent.
Method: GET
Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.
- fiware-reset: if `true` return metrics and inmediatelly after they are deleted , if `false` metrics don’t delete.

Return:
In case of performing the method correctly, returns “`200 OK`” join to all metrics information.

delete metrics

`http://{{url}}:{{port-cygnus}}/v1/admin/metrics`

Definition: Method to delete all metrics for a Cygnus agent (putting counters to zero).

Method: DELETE

Parameters:

- url: Link to the service that will be consulted.
- port-cygnus: Port to connect with the service.

Return:

In case of performing the method correctly, returns “`200 OK`”.