

Latest version:

<http://orange-opensource.github.io/fiware-cepheus/apiary/cep/latest>

Abstract

Cepheus-CEP provides a Complex Event Processor (CEP) at the gateway level with a NGSI 9/10 interface for the FIWARE Data Handling GE.

Cepheus-CEP allows to locally process basic events (from data provided by sensors) and generate higher-level aggregated events. All input and output of events is done through HTTP requests conforming to the NGSI information model.

For more information about the Cepheus-CEP project, please refer to the project's [README](https://github.com/Orange-OpenSource/fiware-cepheus/blob/master/README.md) (<https://github.com/Orange-OpenSource/fiware-cepheus/blob/master/README.md>) on Github or at the Cepheus-CEP [manual](http://fiware-cepheus.readthedocs.org/en/latest/) (<http://fiware-cepheus.readthedocs.org/en/latest/>) .

Table of Contents

Abstract	1
Table of Contents	2
API Specification	5
Default	5
Configuration [/v1/admin/config]	5
Retrieve the configuration - GET /v1/admin/config	5
Update the configuration - POST /v1/admin/config	5
NGSI API v1 [/ngsi10, /ngsi9]	6
updateContext - POST /ngsi10/updateContext	6
notifyContext - POST /ngsi10/notifyContext	6
Examples	6
Default	6
Configuration [/v1/admin/config]	6
Retrieve the configuration - GET /v1/admin/config	6
Update the configuration - POST /v1/admin/config	9
NGSI API v1 [/ngsi10, /ngsi9]	12
updateContext - POST /ngsi10/updateContext	12
notifyContext - POST /ngsi10/notifyContext	13
References	16

API Specification

Default

Configuration [/v1/admin/config]

The configuration endpoint defines a single REST endpoint where the whole configuration is available. The endpoint path is `/v1/admin/config`. The endpoint accepts two HTTP verbs : `GET` and `POST`.

For more information about the JSON format, please see [JSON Configuration](#)

Retrieve the configuration - GET /v1/admin/config

This endpoint returns the actual configuration as a JSON object with a `200 Ok` status code.

It can also return a `404 Not found` code, if no configuration is available.

Response 200 (application/json)

Response 404

[Go to example](#)

Update the configuration - POST /v1/admin/config

This endpoint applies a new configuration given in the body as JSON.

The endpoint will return `200 Ok` on a successful operation, or `400 Bad Request` if the new configuration cannot be applied.

Once the new configuration has been successfully applied to the CEP, the configuration is persisted on disk. If the Cepheus-cep is later restarted, it will automatically load the last configuration on startup.

Request (application/json)

Response 200

Response 400

[Go to example](#)

NGSI API v1 [/ngsi10, /ngsi9]

To access the NGSI v1 API, there is two endpoints : `/ngsi10` and `/ngsi9` .

For compatibility purposes with Fiware-Orion and IoT Agents, two alias are also supported : `/v1` and `/v1/registry` .

The Cepheus-CEP exposes only two NGSI operations : `updateContext` and `notifyContext` operations.

See the [NGSI v1 API from the Fiware-Orion project \(http://telefonicaid.github.io/fiware-orion/api/v1/\)](http://telefonicaid.github.io/fiware-orion/api/v1/) for more information on these two requests.

Theses requests will always return a `200 OK` HTTP status return code (required by NGSI V1).

updateContext - POST /ngsi10/updateContext

The `updateContext` request is used to send directly new events to the CEP.

Request (application/json)

Response 200

[Go to example](#)

notifyContext - POST /ngsi10/notifyContext

The `notifyRequest` will be sent by Context Providers that received a `subscribeContext` request from Cepheus-CEP.

Request (application/json)

Response 200

[Go to example](#)

Examples

Default

Configuration [/v1/admin/config]

Retrieve the configuration - GET /v1/admin/config

Response 200 (application/json)

Headers

Content-Type: application/json

Body

```
{
  "host": "http://localhost:8080",
  "in": [
    {
      "id": "RoomX",
      "type": "Room",
      "isPattern": false,
      "attributes": [
        { "name": "temperature", "type": "double" },
        { "name": "floor", "type": "string" }
      ],
      "providers": [
        "http://localhost:8081"
      ]
    }
  ],
  "out": [
    {
      "id": "FloorX",
      "type": "Floor",
      "attributes": [
        { "name": "temperature", "type": "double" }
      ],
      "brokers": [
        { "url": "http://localhost:8081" }
      ]
    }
  ],
  "statements": [
    "INSERT INTO Floor SELECT floor as id, avg(temperature) as tem
perature FROM Room.win:time(10 min) GROUP BY floor OUTPUT LAST E
VERY 10 sec"
  ]
}
```

Schema

```
{
  "title": "Configuration",
  "description": "The Cepheus-CEP configuration",
  "type": "object",
  "properties": {
    "host": { "type": "string" },
    "in": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": { "type": "string" },

```

```

    "type": { "type": "string" },
    "attributes": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": { "type": "string" },
          "type": { "type": "string" },
          "metadata": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "name": { "type": "string" },
                "type": { "type": "string" }
              },
              "required": ["name", "type"]
            }
          }
        },
        "required": ["name", "type"]
      }
    },
    "required": ["name", "type"]
  },
  "providers": {
    "type": "array",
    "items": { "type": "string" }
  },
  "required": ["id", "type", "attributes"]
},
"out": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": { "type": "string" },
      "type": { "type": "string" },
      "attributes": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "name": { "type": "string" },
            "type": { "type": "string" },
            "metadata": {
              "type": "array",
              "items": {
                "type": "object",
                "properties": {
                  "name": { "type": "string" },
                  "type": { "type": "string" }
                },
                "required": ["name", "type"]
              }
            }
          },
          "required": ["name", "type"]
        }
      }
    }
  }
}

```



```

    }
  },
  "brokers": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "url": { "type": "string" },
        "serviceName": { "type": "string" },
        "servicePath": { "type": "string" },
        "authToken": { "type": "string" }
      },
      "required": [ "url" ]
    }
  },
  "required": [ "id", "type", "attributes" ]
},
"statements": {
  "type": "array",
  "items": { "type": "string" }
},
"required" : [ "host", "in", "out", "statements" ]
}

```

Response 404

[Go to specification](#)

Update the configuration - POST /v1/admin/config

Request (application/json)

Headers

Content-Type: application/json

Body

```

{
  "host": "http://localhost:8080",
  "in": [
    {
      "id": "RoomX",
      "type": "Room",
      "isPattern": false,
      "attributes": [
        { "name": "temperature", "type": "double" },
        { "name": "floor", "type": "string" }
      ]
    }
  ],

```

```

        "providers":[
            "http://localhost:8081"
        ]
    },
    ],
    "out":[
        {
            "id":"FloorX",
            "type":"Floor",
            "attributes":[
                { "name":"temperature", "type":"double" }
            ],
            "brokers":[
                { "url":"http://localhost:8081" }
            ]
        }
    ],
    "statements":[
        "INSERT INTO Floor SELECT floor as id, avg(temperature) as tem
perature FROM Room.win:time(10 min) GROUP BY floor OUTPUT LAST E
VERY 10 sec"
    ]
}

```

Schema

```

{
    "title": "Configuration",
    "description": "The Cepheus-CEP configuration",
    "type": "object",
    "properties": {
        "host": { "type": "string" },
        "in": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "id": { "type": "string" },
                    "type": { "type": "string" },
                    "attributes": {
                        "type": "array",
                        "items": {
                            "type": "object",
                            "properties": {
                                "name": { "type": "string" },
                                "type": { "type": "string" },
                                "metadata": {
                                    "type": "array",
                                    "items": {
                                        "type": "object",
                                        "properties": {
                                            "name": { "type": "string" },
                                            "type": { "type": "string" }
                                        }
                                    },
                                    "required": ["name", "type"]
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
    },
    },
    "required": ["name", "type"]
}
},
"providers": {
    "type": "array",
    "items": { "type": "string" }
}
},
"required": ["id", "type", "attributes"]
}
},
"out": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "id": { "type": "string" },
            "type": { "type": "string" },
            "attributes": {
                "type": "array",
                "items": {
                    "type": "object",
                    "properties": {
                        "name": { "type": "string" },
                        "type": { "type": "string" },
                        "metadata": {
                            "type": "array",
                            "items": {
                                "type": "object",
                                "properties": {
                                    "name": { "type": "string" },
                                    "type": { "type": "string" }
                                },
                                "required": ["name", "type"]
                            }
                        }
                    }
                }
            }
        },
        "required": ["name", "type"]
    }
}
},
"brokers": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "url": { "type": "string" },
            "serviceName": { "type": "string" },
            "servicePath": { "type": "string" },
            "authToken": { "type": "string" }
        },
        "required": [ "url" ]
    }
}
},

```

```

        "required": ["id", "type", "attributes"]
    },
    "statements": {
        "type": "array",
        "items": { "type": "string" }
    }
},
"required": ["host", "in", "out", "statements"]
}

```

Response 200

Response 400

[Go to specification](#)

NGSI API v1 [/ngsi10, /ngsi9]

updateContext - POST /ngsi10/updateContext

Request (application/json)

Headers

Content-Type: application/json

Body

```

{
  "contextElements": [
    {
      "type": "Room",
      "isPattern": "false",
      "id": "room11",
      "attributes": [
        {
          "name": "temperature",
          "type": "double",
          "value": "24",
        },
        {
          "name": "floor",
          "type": "string",
          "value": "3",
        }
      ]
    }
  ],
  "updateAction": "UPDATE"
}

```

Response 200

Body

```
{
  "contextElementResponses": [
    {
      "contextElement": {
        "id": "Room11",
        "type": "Room",
        "isPattern": false,
        "attributes": [
          {
            "name": "temperature",
            "type": "double",
            "value": "15",
            "metadatas": []
          },
          {
            "name": "floor",
            "type": "string",
            "value": "Floor1",
            "metadatas": []
          }
        ]
      },
      "statusCode": {
        "code": "200",
        "reasonPhrase": "OK",
        "detail": "All is OK"
      }
    }
  ]
}
```

[Go to specification](#)

notifyContext - POST /ngsi10/notifyContext

Request (application/json)

Headers

Content-Type: application/json

Body

```
{
  "subscriptionId": "DEADBEEF",
  "originator": "http://localhost:8080",
  "contextElements": [
```

```
{
  "type": "Room",
  "isPattern": "false",
  "id": "room11",
  "attributes": [
    {
      "name": "temperature",
      "type": "double",
      "value": "24",
    },
    {
      "name": "floor",
      "type": "string",
      "value": "3",
    }
  ]
}
```

Response 200

Body

```
{
  "responseCode": {
    "code": "200",
    "reasonPhrase": "OK",
    "detail": "All is OK"
  }
}
```

[Go to specification](#)

References

- README (<https://github.com/Orange-OpenSource/fiware-cepheus/blob/master/README.md>)
- manual (<http://fiware-cepheus.readthedocs.org/en/latest/>)
- NGSI v1 API from the Fiware-Orion project(<http://telefonicaid.github.io/fiware-orion/api/v1/>)