

REST interface V2

This REST interface is the new standard for communication with odysseus.

Remark: In Client-Server-Scenarios you will need <http://<host>> in the connection information. The default port is 8888 (can be changed)

We provide a OpenAPI specification for the current state of the REST interface https://git.swl.informatik.uni-oldenburg.de/projects/ODY/repos/odysseus_core/browse/common/rest2.common/resources/odysseus.yaml

Examples

The following is an example, step-by-step, of how to install a query via REST.

Example login request:

```
POST /services/login HTTP/1.1
Accept: application/json, */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Content-Length: 45
Content-Type: application/json
Host: 192.168.110.31:8888
User-Agent: HTTPie/1.0.2
{
  "password": "manager",
  "username": "System"
}
```

Example response:

```
HTTP/1.1 200 OK
Accept: application/json, */*
Accept-Encoding: gzip, deflate
Host: 192.168.110.31:8888
Set-Cookie: JSESSIONID=A3B9A93D65CF9812D696F1E612262CA5
User-Agent: HTTPie/1.0.2
content-encoding: gzip
content-length: 64
content-type: application/json
date: Tue, 13 Aug 2019 12:06:20 +0200
server: wso2-http-transport
{
  "token": "jtds2ph0fkhkq8d4dv1fj2qhe3"
}
```

You need to send that token with every following request.

Now let's install a new query via the Rest API.

POST request to /queries. These are the fields you can use:

```
{
  "name": "Give your new query a name",
  "parser": "The name of the parser you want to use. Like PQL or OdysseusScript",
  "state": "Running if you want to immediately start a query when not using OdysseusScript as your parser",
  "queryText": "your query source code goes here"
}
```

Example request installing the first query from the nexmark example (notice the 'Authorization' header):

```
POST /queries HTTP/1.1
Accept: application/json, */*
Accept-Encoding: gzip, deflate
Authorization: Bearer jtds2ph0fkhkq8d4dv1fj2qhe3
Connection: keep-alive
Content-Length: 600
Content-Type: application/json
Host: 192.168.110.31:8888
User-Agent: HTTPie/1.0.2
{
  "parser": "OdysseusScript",
  "queryText": "#PARSER PQL\n#RUNQUERY\nnexmark:person := ACCESS(\n\t{\n\t\tsource='nexmark:person',
\n\t\twrapper='GenericPush',\n\t\ttransport='TCPClient',\n\t\tprotocol='SizeByteBuffer',\n\t\tdataHandler='Tuple',
\n\t\toptions=[\n\t\t\t['host', 'localhost'],\n\t\t\t['port', '65440'],\n\t\t\t['ByteOrder', 'LittleEndian']
\n\t\t],\n\t\tschema=[\n\t\t\t['timestamp', 'STARTTIMESTAMP'],\n\t\t\t['id', 'INTEGER'],\n\t\t\t['name',
'String'],\n\t\t\t['email', 'STRING'],\n\t\t\t['creditcard', 'STRING'],\n\t\t\t['city', 'STRING'],\n\t\t\t['state', 'STRING']\n\t\t]\n\t)\n\n"
```

Example response:

```
HTTP/1.1 200 OK
Accept: application/json, */*
Accept-Encoding: gzip, deflate
Authorization: Bearer jtids2ph0fkhkq8d4dv1fj2qhe3
Host: 192.168.110.31:8888
Set-Cookie: JSESSIONID=BF6F8590C617FD6CE25DD30CA2670CB2
User-Agent: HTTPie/1.0.2
content-encoding: gzip
content-length: 28
content-type: application/json
date: Tue, 13 Aug 2019 12:53:31 +0200
server: wso2-http-transport
[]
```

Now check if the nexmark source has been added, by GET requesting /datastreams:

```
GET /datastreams HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Authorization: Bearer jtids2ph0fkhkq8d4dv1fj2qhe3
Connection: keep-alive
Host: 192.168.110.31:8888
User-Agent: HTTPie/1.0.2
```

Example response:

```
HTTP/1.1 200 OK
Accept: */*
Accept-Encoding: gzip, deflate
Authorization: Bearer jtids2ph0fkhkq8d4dv1fj2qhe3
Host: 192.168.110.31:8888
Set-Cookie: JSESSIONID=DF701C0AEAC5470508C823579A6C7BD8
User-Agent: HTTPie/1.0.2
content-encoding: gzip
content-length: 321
content-type: application/json
date: Tue, 13 Aug 2019 13:03:04 +0200
server: wso2-http-transport
[
```

```
{
  "name": "nexmark:person",
  "owner": "System",
  "schema": {
    "attributes": [
      {
        "attributename": "timestamp",
        "datatype": {
          "type": "BASE",
          "uri": "StartTimestamp"
        },
        "sourcename": "nexmark:person"
      },
      {
        "attributename": "id",
        "datatype": {
          "type": "BASE",
          "uri": "Integer"
        },
        "sourcename": "nexmark:person"
      },
      {
        "attributename": "name",
        "datatype": {
          "type": "BASE",
          "uri": "String"
        },
        "sourcename": "nexmark:person"
      },
      {
        "attributename": "email",
        "datatype": {
          "type": "BASE",
          "uri": "String"
        },
        "sourcename": "nexmark:person"
      },
      {
        "attributename": "creditcard",
        "datatype": {
          "type": "BASE",
          "uri": "String"
        }
      }
    ]
  }
}
```

```

    },
    "sourcename": "nexmark:person"
  },
  {
    "attributename": "city",
    "datatype": {
      "type": "BASE",
      "uri": "String"
    },
    "sourcename": "nexmark:person"
  },
  {
    "attributename": "state",
    "datatype": {
      "type": "BASE",
      "uri": "String"
    },
    "sourcename": "nexmark:person"
  }
],
"metaschema": [
  {
    "attributes": [
      {
        "attributename": "start",
        "datatype": {
          "type": "BASE",
          "uri": "Timestamp"
        },
        "sourcename": "TimeInterval"
      },
      {
        "attributename": "end",
        "datatype": {
          "type": "BASE",
          "uri": "Timestamp"
        },
        "sourcename": "TimeInterval"
      }
    ],
    "metaattributeClass": "de.uniol.inf.is.odysseus.core.metadata.ITimeInterval",
    "typeClass": "de.uniol.inf.is.odysseus.core.collection.Tuple",
    "uri": "TimeInterval"
  }
],
"typeClass": "de.uniol.inf.is.odysseus.core.collection.Tuple",
"uri": "nexmark:person"
},
"type": "source"
},
{
  "name": "nexmark:person",
  "owner": "System",
  "schema": {
    "attributes": [
      {
        "attributename": "timestamp",
        "datatype": {
          "type": "BASE",
          "uri": "StartTimestamp"
        },
        "sourcename": "nexmark:person"
      },
      {
        "attributename": "id",
        "datatype": {
          "type": "BASE",
          "uri": "Integer"
        },
        "sourcename": "nexmark:person"
      },
      {
        "attributename": "name",
        "datatype": {
          "type": "BASE",
          "uri": "String"
        },
        "sourcename": "nexmark:person"
      },
      {
        "attributename": "email",
        "datatype": {
          "type": "BASE",
          "uri": "String"
        }
      }
    ]
  }
}

```

```

    },
    "sourcename": "nexmark:person"
  },
  {
    "attributename": "creditcard",
    "datatype": {
      "type": "BASE",
      "uri": "String"
    },
    "sourcename": "nexmark:person"
  },
  {
    "attributename": "city",
    "datatype": {
      "type": "BASE",
      "uri": "String"
    },
    "sourcename": "nexmark:person"
  },
  {
    "attributename": "state",
    "datatype": {
      "type": "BASE",
      "uri": "String"
    },
    "sourcename": "nexmark:person"
  }
],
"metaschema": [
  {
    "attributes": [
      {
        "attributename": "start",
        "datatype": {
          "type": "BASE",
          "uri": "Timestamp"
        },
        "sourcename": "TimeInterval"
      },
      {
        "attributename": "end",
        "datatype": {
          "type": "BASE",
          "uri": "Timestamp"
        },
        "sourcename": "TimeInterval"
      }
    ],
    "metaattributeClass": "de.uniol.inf.is.odysseus.core.metadata.ITimeInterval",
    "typeClass": "de.uniol.inf.is.odysseus.core.collection.Tuple",
    "uri": "TimeInterval"
  }
],
"typeClass": "de.uniol.inf.is.odysseus.core.collection.Tuple",
"uri": "nexmark:person"
},
"type": "access"
}
]

```