FEWS PI REST Web Service

Introduction

The FEWS PI REST Web Service allows interacting with Delft-FEWS using a REST API.

Developers documentation

For API Developers it is recommended to use the Open API Documentation. Please see: Open API Specification Documentation

WebServices Configuration (since 2022.02)

Since 2022.02 a WebServices xml configuration file is supported (and replaces the deprecated FewsPiServices.properties file). See FEWS WebServices Configuration File (since 2022.02)

Configuration properties

Deprecated since 2022.02. Will be removed in a future release of the Delft-FEWS Web Services. Please use the WebService Configuration instead.

The FewsPiService.properties file can be used to make service specific configurations available. This is a property file that is located in the FEWS configuration in the directory for a standalone system or should be uploaded with the ConfigManager for a live system:

%REGION_HOME%/Config/PiClientConfigFiles/FewsPiService.properties

For more information about the possible properties, see FEWS Web Services configuration FewsPiService.properties.

FEWS PI REST Web Service API

GET timezoneid

Get ID of Configured timezone for the webservice.

Request parameters

- not applicable

Response

- String representation of time zone.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/timezoneid"
```

Example response

```
GMT
```

GET filters

Retrieve Pi Filters configuration.

Request parameters

- *filterId* (string): Subset filter id.
- *documentVersion (string)*: File format version.
- *documentFormat* (string): PI_XML (default) or PI_JSON

Response

- Filters PI-XML or PI-JSON file content.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/filters"
```

Example PI-XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<filters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas/pi_filter.
xsd" version="1.23">
    <filter id="Countries">
        <name>Countries</name>
        <description></description>
        <child id="All">
            <name>All</name>
            <description></description>
        </child>
        <child id="Netherlands">
            <name>Netherlands</name>
            <description></description>
        </child>
    </filter>
</filters>
```

GET filters/actions (2022.02)

Retrieve a json-file which contains the request by which the selected time series from the filter can be retrieved from the database. In addition, detailed information about how to display the time series will be returned.

Request parameters

- *filterId (string):* filter id,
- *parameterGroupId (string):* parameter group id,
- *parameterIds (string):* one or more parameter ids,
- *locationIds (string):* one or more location ids,
- *timeZero (string):* the time zero for which the requests to retrieve the time series will be generated.
- *convertDatum* (boolean): Convert values from relative location height to absolute height values.
- *useDisplayUnits* (boolean): Export values using display units. Also use displayUnits in the generated requests

Response

- Actions for the selected time series from the provided filter

Example request

```
curl "http://localhost:63178/FewsWebServices/rest/fewspiservice/v1/filters/actions?parameterIds=Q.
voorspeld&filterId=discharge&timeZero=2014-01-01T00:00:00Z&locationIds=amerongen_boven"
```

GET topology/actions (since 2022.01)

Get the displaysgroups for a certain topology node

Request parameters

- nodeId: the id of the topology node)
- timeZero: the timezero for which the displaysgroups should be requested. If this parameter is ommitted the time zero will be assumed equal to the current system time
- convertDatum (boolean): Convert values from relative location height to absolute height values.
- useDisplayUnits (boolean): Export values using display units. Also use displayUnits in the generated requests

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/topology/actions?
nodeId=groupNodeA&timeZero=2014-01-01T00:00:00Z"
```

Example response

The response is always in JSON. Below an example.

```
{
  "results" : [ {
    "type" : "PI",
    "requests" : [ {
      "key" : "....",
      "request" : "rest/fewspiservice/v1/timeseries?
timeSeriesType=EXTERNAL_FORECASTING&locationIds=amerongen_beneden&parameterIds=Q.
voorspeld&moduleInstanceIds=Forecast&qualifierIds=NONE&timeStepId=PT10M&endForecastTime=2014-01-01T00%3A00%
3A00Z&useDisplayUnits=true&documentFormat=PI_JSON"
    } ],
    "config" : {
      "timeSeriesDisplay" : {
        "title" : "no name",
        "subplots" : [ {
          "items" : [ {
            "type" : "line",
            "legend" : "Q.voorspeld",
            "color" : "#0000ff",
            "lineStyle" : "solid",
            "lineWidth" : 1.0,
            "locationId" : "amerongen_beneden",
            "yAxis" : {
              "axisPosition" : "left",
              "axisLabel" : "Afvoer (m3/s)"
            },
            "request" : "...."
          } ]
        } ]
      }
    }
  } ]
}
```

GET locations

Retrieve Pi Locations file containing all locations that are available for the passed 'filterId' argument. If not filterId is passed then all locations configured in the pre-defined filter will be returned. The geoDatum used in the response is determined by the geoDatum configured in the locations.xml.

Request parameters

- *filterId* (string): Filter id.
- *parameterIds(string):* Since 2022.02 parameter id(s) which will be used to filter the time series of the filter
- *parameterGroupId(string):* Since 2022.02 parameter group id which will be used to filter the time series of the filter
- *documentVersion* (string): File format version.
- *documentFormat* (string): PI_XML (default), PI_JSON or GEO_JSON (since 2021.02)
- *showAttributes* (boolean) Since 2017.02: toggle to show location attributes.
- *includeLocationRelations* (boolean) Since 2019.02: toggle to include locationRelations. Optional, default is false. For xml format avalable from version 1.26 or higher.
- *includeTimeDependency* (boolean) Since 2019.02: toggle to include timeDependency. Optional, default is false. For xml format avalable from version 1.26 or higher.

Response

- Locations in PI-XML, GEO_JSON or PI-JSON file content.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/locations?
showAttributes=true&documentVersion=1.24"
```

Example PI-XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<Locations xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas/pi_locations.
xsd" version="1.24">
    <geoDatum>WGS 1984</geoDatum>
         <location locationId="locA">
        <shortName>A</shortName>
        <lat>9.0</lat>
        <lon>9.0</lon>
        <x>9.0</x>
        <y>9.0</y>
        <z>9.0</z>
        <relation id="UPSTREAM">
            <relatedLocationId>locB</relatedLocationId>
        </relation>
        <relation id="UPSTREAM">
            <relatedLocationId>locC</relatedLocationId>
        </relation>
        <attribute id="TEST_ATTRIBUTE" name="TEST_ATTRIBUTE">
            <startDateTime>1890-01-01T00:00:00+0000</startDateTime>
            <endDateTime>1950-01-01T00:00:00+0000</endDateTime>
            <text>A1</text>
        </attribute>
        <attribute id="TEST_ATTRIBUTE" name="TEST_ATTRIBUTE">
            <startDateTime>1950-01-01T00:00:00+0000</startDateTime>
            <text>A2</text>
        </attribute>
    </location>
    <location locationId="locB">
        <shortName>B</shortName>
        <lat>7.0</lat>
        <lon>7.0</lon>
        <x>7.0</x>
        <y>7.0</y>
        <z>7.0</z>
        <relation id="UPSTREAM">
            <relatedLocationId>locC</relatedLocationId>
            <startDateTime>1930-01-01T00:00:00+0000</startDateTime>
            <endDateTime>1980-01-01T00:00:00+0000</endDateTime>
        </relation>
        <relation id="UPSTREAM">
            <relatedLocationId>locF</relatedLocationId>
            <endDateTime>1980-01-01T00:00:00+0000</endDateTime>
        </relation>
        <relation id="UPSTREAM">
            <relatedLocationId>locE</relatedLocationId>
            <startDateTime>1930-01-01T00:00:00+0000</startDateTime>
        </relation>
        <attribute id="TEST_ATTRIBUTE" name="TEST_ATTRIBUTE">
            <startDateTime>1930-01-01T00:00:00+0000</startDateTime>
            <endDateTime>2000-01-01T00:00:00+0000</endDateTime>
            <text>B</text>
        </attribute>
    </location>
    <location locationId="locC">
        <shortName>C</shortName>
        <lat>5.0</lat>
        <lon>5.0</lon>
        <x>5.0</x>
        <y>5.0</y>
        <z>5.0</z>
    </location>
    <location locationId="locD">
        <shortName>D</shortName>
        <lat>3.0</lat>
        <lon>3.0</lon>
        <x>3.0</x>
        <y>3.0</y>
        <z>3.0</z>
        <relation id="UPSTREAM">
```

```
            <relatedLocationId>locB</relatedLocationId>
        </relation>
        <relation id="UPSTREAM">
            <relatedLocationId>locE</relatedLocationId>
        </relation>
        <attribute id="TEST_ATTRIBUTE" name="TEST_ATTRIBUTE">
            <startDateTime>1930-01-01T00:00:00+0000</startDateTime>
            <endDateTime>1980-01-01T00:00:00+0000</endDateTime>
            <text>D</text>
        </attribute>
    </location>
    <location locationId="locE">
        <shortName>E</shortName>
        <lat>2.0</lat>
        <lon>2.0</lon>
        <x>2.0</x>
        <y>2.0</y>
        <z>2.0</z>
        <relation id="UPSTREAM">
            <relatedLocationId>locF</relatedLocationId>
        </relation>
        <attribute id="TEST_ATTRIBUTE" name="TEST_ATTRIBUTE">
            <startDateTime>1910-01-01T00:00:00+0000</startDateTime>
            <endDateTime>1930-01-01T00:00:00+0000</endDateTime>
            <text>E1</text>
        </attribute>
        <attribute id="TEST_ATTRIBUTE" name="TEST_ATTRIBUTE">
            <startDateTime>1930-01-01T00:00:00+0000</startDateTime>
            <endDateTime>1950-01-01T00:00:00+0000</endDateTime>
            <text>E2</text>
        </attribute>
        <attribute id="TEST_ATTRIBUTE" name="TEST_ATTRIBUTE">
            <startDateTime>1950-01-01T00:00:00+0000</startDateTime>
            <endDateTime>1980-01-01T00:00:00+0000</endDateTime>
            <text>E3</text>
        </attribute>
    </location>
    <location locationId="locF">
        <shortName>F</shortName>
        <lat>1.0</lat>
        <lon>1.0</lon>
        <x>1.0</x>
        <y>1.0</y>
        <z>1.0</z>
        <attribute id="TEST_ATTRIBUTE" name="TEST_ATTRIBUTE">
            <text>F</text>
        </attribute>
    </location>
</Locations>
```

Example PI-JSON response

```
 {
  "version" : "1.26",
  "geoDatum" : "WGS 1984",
  "locations" : [ {
    "locationId" : "locA",
    "shortName" : "A",
    "lat" : "9.0",
    "lon" : "9.0",
    "x" : "9.0",
    "y" : "9.0",
    "z" : "9.0",
    "attributes" : [ {
      "name" : "TEST_ATTRIBUTE",
      "type" : "text",
      "id" : "TEST_ATTRIBUTE",
      "startDateTime" : "1890-01-01T00:00:00+0000",
      "endDateTime" : "1950-01-01T00:00:00+0000",
```

```
    "value" : "A1"
  }, {
    "name" : "TEST_ATTRIBUTE",
    "type" : "text",
    "id" : "TEST_ATTRIBUTE",
    "startDateTime" : "1950-01-01T00:00:00+0000",
    "value" : "A2"
  } ],
  "relations" : [ {
    "id" : "UPSTREAM",
    "relatedLocationId" : "locB"
  }, {
    "id" : "UPSTREAM",
    "relatedLocationId" : "locC"
  } ]
}, {
  "locationId" : "locB",
  "shortName" : "B",
  "lat" : "7.0",
  "lon" : "7.0",
  "x" : "7.0",
  "y" : "7.0",
  "z" : "7.0",
  "attributes" : [ {
    "name" : "TEST_ATTRIBUTE",
    "type" : "text",
    "id" : "TEST_ATTRIBUTE",
    "startDateTime" : "1930-01-01T00:00:00+0000",
    "endDateTime" : "2000-01-01T00:00:00+0000",
    "value" : "B"
  } ],
  "relations" : [ {
    "id" : "UPSTREAM",
    "relatedLocationId" : "locC",
    "startDateTime" : "1930-01-01T00:00:00+0000",
    "endDateTime" : "1980-01-01T00:00:00+0000"
  }, {
    "id" : "UPSTREAM",
    "relatedLocationId" : "locF",
    "endDateTime" : "1980-01-01T00:00:00+0000"
  }, {
    "id" : "UPSTREAM",
    "relatedLocationId" : "locE",
    "startDateTime" : "1930-01-01T00:00:00+0000"
  } ]
}, {
  "locationId" : "locC",
  "shortName" : "C",
  "lat" : "5.0",
  "lon" : "5.0",
  "x" : "5.0",
  "y" : "5.0",
  "z" : "5.0",
  "attributes" : [ ],
  "relations" : [ ]
}, {
  "locationId" : "locD",
  "shortName" : "D",
  "lat" : "3.0",
  "lon" : "3.0",
  "x" : "3.0",
  "y" : "3.0",
  "z" : "3.0",
  "attributes" : [ {
    "name" : "TEST_ATTRIBUTE",
    "type" : "text",
    "id" : "TEST_ATTRIBUTE",
    "startDateTime" : "1930-01-01T00:00:00+0000",
    "endDateTime" : "1980-01-01T00:00:00+0000",
    "value" : "D"
  } ],
```

```
    "relations" : [ {
      "id" : "UPSTREAM",
      "relatedLocationId" : "locB"
    }, {
      "id" : "UPSTREAM",
      "relatedLocationId" : "locE"
    } ]
  }, {
    "locationId" : "locE",
    "shortName" : "E",
    "lat" : "2.0",
    "lon" : "2.0",
    "x" : "2.0",
    "y" : "2.0",
    "z" : "2.0",
    "attributes" : [ {
      "name" : "TEST_ATTRIBUTE",
      "type" : "text",
      "id" : "TEST_ATTRIBUTE",
      "startDateTime" : "1910-01-01T00:00:00+0000",
      "endDateTime" : "1930-01-01T00:00:00+0000",
      "value" : "E1"
    }, {
      "name" : "TEST_ATTRIBUTE",
      "type" : "text",
      "id" : "TEST_ATTRIBUTE",
      "startDateTime" : "1930-01-01T00:00:00+0000",
      "endDateTime" : "1950-01-01T00:00:00+0000",
      "value" : "E2"
    }, {
      "name" : "TEST_ATTRIBUTE",
      "type" : "text",
      "id" : "TEST_ATTRIBUTE",
      "startDateTime" : "1950-01-01T00:00:00+0000",
      "endDateTime" : "1980-01-01T00:00:00+0000",
      "value" : "E3"
    } ],
    "relations" : [ {
      "id" : "UPSTREAM",
      "relatedLocationId" : "locF"
    } ]
  }, {
    "locationId" : "locF",
    "shortName" : "F",
    "lat" : "1.0",
    "lon" : "1.0",
    "x" : "1.0",
    "y" : "1.0",
    "z" : "1.0",
    "attributes" : [ {
      "name" : "TEST_ATTRIBUTE",
      "type" : "text",
      "id" : "TEST_ATTRIBUTE",
      "value" : "F"
    } ],
    "relations" : [ ]
  } ]
}
```


GET parameters

Retrieve PI Parameters file containing all parameters that are available for the passed 'filterId' argument. Parameters are also returned if no time series are available for a parameter. If no filterId is passed then all parameters configured in the pre-defined filter will be returned.

Request parameters

- *showAttributes (boolean): Since 2021.02. toggle to show parameter attributes. Default is false.*
- *documentVersion* (string): File format version.
- *documentFormat* (string): PI_XML (default) or PI_JSON

Response

- Parameters PI-XML or PI-JSON file content.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/parameters"
```

Example PI-XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<timeseriesparameters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews
/PI" xsi:schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas
/pi_timeseriesparameters.xsd" version="1.25">
    <parameter id="T.obs.mean" parameterGroup="Temperature">
        <name>Observed Monthly Average Temperature</name>
        <parameterType>instantaneous</parameterType>
        <unit>oC</unit>
        <displayUnit>oC</displayUnit>
        <usesDatum>false</usesDatum>
    </parameter>
</timeseriesparameters>
```

Example PI-JSON response

```
{
  "version" : "1.25",
  "timeSeriesParameters" : [ {
    "id" : "T.obs.mean",
    "name" : "Observed Monthly Average Temperature",
    "parameterType" : "instantaneous",
    "unit" : "oC",
    "displayUnit" : "oC",
    "usesDatum" : "false",
    "parameterGroup" : "Temperature"
  } ]
}
```

GET parameters/nodes (since 2021.02)

Get all parameter nodes and its parameters that are available in the region config. For each parameterNode the parent node, children nodes and all parameter ids are provided.

Request parameters

- *documentVersion* (string): File format version.
- *documentFormat* (string): PI_XML (default) or PI_JSON

Response

- Parameters PI-XML or PI-JSON file content.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/parameters/nodes"
```

Example PI-XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<parameterNodes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas
/pi_parameter_nodes.xsd">
    <parameterNode id="subNode">
        <name>Wave Forecast Swell Direction</name>
        <description></description>
        <parameters>
            <parameter>Wave.forecast.swell.dir</parameter>
            <parameter>Wave.forecast.wind.dir</parameter>
        </parameters>
        <parent>Wave Direction</parent>
    </parameterNode>
    <parameterNode id="Wave Direction">
        <name>Wave Direction</name>
        <description>Wave Direction Description</description>
        <parameters>
            <parameter>Wave.obs.swell.dir</parameter>
            <parameter>Wave.obs.wind.dir</parameter>
        </parameters>
        <parent>Parameters</parent>
        <children>
            <child>subNode</child>
        </children>
    </parameterNode>
    <parameterNode id="Constraint">
        <name>Constraint spectral density</name>
        <description></description>
        <parameters>
            <parameter>2d_spectral_density</parameter>
        </parameters>
        <parent>Parameters</parent>
    </parameterNode>
    <parameterNode id="Parameters">
        <name>Tansley</name>
        <description></description>
        <parameters>
            <parameter>TansleyX_PTB</parameter>
            <parameter>TansleyS_PTB</parameter>
            <parameter>TansleyS_XXX</parameter>
            <parameter>Q.meting</parameter>
            <parameter>H.voorspeld</parameter>
            <parameter>Q.voorspeld</parameter>
            <parameter>H.forecast.ensemble</parameter>
            <parameter>frequency</parameter>
            <parameter>direction</parameter>
            <parameter>T-historical</parameter>
            <parameter>Wave.forecast.total.dir</parameter>
            <parameter>Wave.obs.total.dir</parameter>
            <parameter>H.meting</parameter>
            <parameter>H.voorspeld.daily</parameter>
        </parameters>
        <children>
            <child>Wave Direction</child>
            <child>Constraint</child>
        </children>
    </parameterNode>
</parameterNodes>
```

Example PI-JSON response

```
{
  "version" : "1.28",
  "parameterNodes" : [ {
```

```
  "id" : "subNode",
  "name" : "Wave Forecast Swell Direction",
  "parameters" : [ {
    "id" : "Wave.forecast.swell.dir"
  }, {
    "id" : "Wave.forecast.wind.dir"
  } ],
  "parent" : {
    "id" : "Wave Direction"
  }
}, {
  "id" : "Wave Direction",
  "name" : "Wave Direction",
  "description" : "Wave Direction Description",
  "parameters" : [ {
    "id" : "Wave.obs.swell.dir"
  }, {
    "id" : "Wave.obs.wind.dir"
  } ],
  "parent" : {
    "id" : "Parameters"
  },
  "children" : [ {
    "id" : "subNode"
  } ]
}, {
  "id" : "Constraint",
  "name" : "Constraint spectral density",
  "parameters" : [ {
    "id" : "2d_spectral_density"
  } ],
  "parent" : {
    "id" : "Parameters"
  }
}, {
  "id" : "Parameters",
  "name" : "Tansley",
  "parameters" : [ {
    "id" : "TansleyX_PTB"
  }, {
    "id" : "TansleyS_PTB"
  }, {
    "id" : "TansleyS_XXX"
  }, {
    "id" : "Q.meting"
  }, {
    "id" : "H.voorspeld"
  }, {
    "id" : "Q.voorspeld"
  }, {
    "id" : "H.forecast.ensemble"
  }, {
    "id" : "frequency"
  }, {
    "id" : "direction"
  }, {
    "id" : "T-historical"
  }, {
    "id" : "Wave.forecast.total.dir"
  }, {
    "id" : "Wave.obs.total.dir"
  }, {
    "id" : "H.meting"
  }, {
    "id" : "H.voorspeld.daily"
  } ],
  "children" : [ {
    "id" : "Wave Direction"
  }, {
    "id" : "Constraint"
  } ]
```

```
  } ]
}
```

GET topology/nodes (since 2022.01)

Returns the topology config of the fews system. At the moment only JSON is supported as a return type

Request parameters

This function doesn't have request parameters

Response

The response is always in JSON format. Note that only the most important part of the topology config is returned.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/topology/nodes
```

Example response

```
{
  "topologyNodes" : [ {
    "id" : "groupNodeA",
    "name" : "groupNodeA",
    "topologyNodes" : [ {
      "id" : "nodeA",
      "name" : "nodeA",
      "workflowId" : "Import",
      "localRun" : true,
      "displayId": "exampleDisplayId"
    }, {
      "id" : "nodeB",
      "name" : "nodeB",
      "workflowId" : "Import",
      "localRun" : true,
      "displayGroupId", "exampleDisplayGroupId"
    } ]
  }, {
    "id" : "groupNodeB",
    "name" : "groupNodeB",
    "topologyNodes" : [ {
      "id" : "nodeC",
      "name" : "nameNodeC",
      "workflowId" : "Import",
      "localRun" : true,
       "url" : "exampleUrl"
    }, {
      "id" : "nodeD",
      "name" : "nodeD",
      "workflowId" : "ImportSample",
      "localRun" : false
    } ]
  } ]
}
```

POST timeseries/edit

Upload data edits to FEWS

Request parameters

- *locationId* (string): Id of the location for which the edits are requested

- *ensembleId* (string): Ensemble identifier of for time series
- *ensembleMemberId* (string): Ensemble Member identifier of for time series. Only allowed in combination with ensembleId. Since 2022.02.
- *timeSeriesSetIndex* (integer): index of the requested time series set
- *convertDatum* (boolean): Convert values from relative location height to absolute height values.

Body content

The body of the request should contain the time series in pi-json format.

```
{
  "version": "1.23",
  "timeZone": "0.0",
  "timeSeries": [
    {
      "events": [
        {
          "date": "2013-12-30",
          "time": "01:00:00",
          "value": "22.0",
          "comment": "test2",
          "flag": "6"
        }
      ]
    }
  ]
}
```

GET timeseries/history

returns the edit history for the requested time series and times

Request parameters

- *times* (dateTime: yyyy-MM-ddTHH:mm:ssZ) the times for which the edits are requested
- *locationId* (string): Id of the location for which the edits are requested

- *ensembleId* (string): Ensemble identifier of for time series
- *ensembleMemberId* (string): Ensemble Member identifier of for time series. Only allowed in combination with ensembleId. Since 2022.02.
- *timeSeriesSetIndex* (integer): index of the requested time series set.

Response

- PI-JSON

For example:

```
[ {
  "time" : "2013-12-30T01:00:00Z",
  "edits" : [ {
    "workflowOrUserId" : "?",
    "value" : 22.0,
    "flag" : 0,
    "flagSource" : 0,
    "valueSource" : 0,
    "comments" : "test2"
  } ]
}, {
  "time" : "2013-12-30T00:00:00Z",
  "edits" : [ {
    "workflowOrUserId" : "?",
    "value" : 20.0,
    "flag" : 0,
    "flagSource" : 1,
    "valueSource" : 1,
    "comments" : "test"
  } ]
} ]
```


GET timeseries

Returns a pi timeseries xml file containing the time series data filtered by the query parameters.

Request parameters

- *convertDatum* (boolean): Convert values from relative location height to absolute height values.
- *documentVersion* (string, 1.9 or up): File format version (optional). For example: 1.23
- *documentFormat* (string): PI_XML (default), PI_JSON, DD_JSON, NOOS_TEXT, BINARY (only when running embedded tomcat)
- *endCreationTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): End time of search period that looks for creation time of time series. Note: creation time of time series is actually the creation time of the task that produced/imported these time series.
- *endForecastTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): End time of search period that looks for time series produced by forecasts that have their forecast time within this period.
- *endTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): End time of search period that looks for timeseries values that are within this period. If the endTime doesn't match a timestamp of the time series, the closest timestamp after the endTime, will also be returned. Format: yyyy-MM-ddTHH:mm:ssZ. Take note that if no endTime is specified, the end time of the requested period will be set to the current time plus one day and one hour.
- *ensembleId* (string): Ensemble identifier of for time series
- *ensembleMemberId* (string): Ensemble Member identifier of for time series. Only allowed in combination with ensembleId. Since 2022.02.
- exportIdMap (string): Name of the id map that should be used as exportIdMap. Overrules a configured exportIdmapId from the WebServices.xml general section configuration.
- *externalForecastTimes* (dateTime format: yyyy-MM-ddTHH:mm:ssZ): Time value of external forecast time. This parameter has to be duplicated to specify multiple multiple externalForecastTimes.
- *filterId* (string): An existing subfilter of the default filter id. N.B. Can be used in combination with taskRunIds since 2020.01.
- *forecastCount* (integer): Number of forecast runs to return when using start- and end- forecast time. Default is 1.
- *locationIds* (string): Subset of locations for which to retrieve time series. This parameter can be duplicated to use multiple locationIds.
- *moduleInstanceIds* (string): Subset of moduleInstances for which to retrieve time series. This parameter can be duplicated to specify multiple moduleInstanceIds.
- *omitMissing* (boolean): Toggle omitting or returning of missing values in response
- *omitEmptyTimeSeries* (boolean): Toggle omitting or returning headers of empty timeSeries. Default is false. Since 2020.02.
- *onlyHeaders* (boolean): Toggle to return only header information or also data
- *onlyForecasts* (boolean): Toggle to return only forecast time series (Since 2017.02)
- onlyManualEdits (boolean): Toggle to return only manual edits.
- *parameterIds* (string): Subset of parameters for which to retrieve time series. This parameter has to be duplicated to specify multiple parameters.
- *qualifierIds* (string): Subset of qualifiers for which to retrieve time series. This parameter has to be duplicated to specify multiple qualifierIds. To indicate that no qualifier is available, use qualifierIds: "none"
- showProducts (boolean): Toggle to display product information that is assigned to a forecast. (Since 2019.02). See below for an example.
- *showStatistics* (boolean): Toggle to return statistics information about time series. Typically used in combination with onlyHeaders. Returns additional information about data availability of time series (Since 2015.01). These statistics are only provided if there is any data, otherwise they are left out.
    - firstValueTime: First time with a value in the time series
    - lastValueTime: Last time with a value in the time series
    - maxValue: Maximum value in the time series
    - minValue: Minimum value in the time series
    - valueCount: Number of values in the time series
- *showThresholds* (boolean): Option to toggle the returning of threshold information in the headers
- *showEnsembleMemberIds* (boolean): Show ensemble member ids instead of ensemble member indices.

- *timeSeriesType* (string): Explicitly filter on a specific time series type. (Since 2020.01). Possible values are: EXTERNAL_HISTORICAL, EXTERNAL_FORECASTING, SIMULATED_HISTORICAL, SIMULATED_FORECASTING.
- thinning (long): unit ms/pixel. Thinning is used to retrieve the visually interesting time steps of time series. It tries to keep the peaks and gaps and minimizes the number of time steps that have to be retrieved. It is typically used for visualizations. The value to be specified should be equal to the view period in milliseconds of the time series that is visualized divided by the number of pixels that are available for display. For example: visualizing a view period of 5 years (157784760000 milliseconds) on a display of 1024 pixels, the thinning parameter should be set to 157784760000/1024 = 15408668. (Since 2019.02)
- *useMilliseconds* (boolean) Optional argument. Default is false. If it is set to true, the response will contain milliseconds. See example below. Available in 2017.02 and from 2019.02
- *startCreationTime* (dateTime format: yyyy-MM-ddTHH:mm:ssZ): Start time of search period that looks for creation time of time series. Note: creation time of time series is actually the creation time of the task that produced/imported these time series.
- *startForecastTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): Start time of search period that looks for time series produced by forecasts that have their forecast time within this period. Format: yyyy-MM-ddTHH:mm:ssZ. If left empty all forecasts up to the endForecastTime will be used as search period.
- *startTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): Start time of search period that looks for timeseries values that are within this period. If the startTime doesn't match a timestamp of the time series, the closest timestamp before the startTime, will also be returned. Format: yyyy-MM-ddTHH:mm:ssZ. Take note that if no startTime is specified, the start time of the requested period will be set to the current time minus one day.
- *taskRunIds* (string): Subset of task run ids for which to retrieve time series. This parameter has to be duplicated to specify multiple taskRuns. N. B. cannot be used in combination with a filterId or with startTime and/or endTime. Since 2024.01 a response code 206 may be returned if one of the task runs is not completely available to the webservice (either not completed yet, or not available due to index files not up to date). In that case a client should retry the request.
- *useDisplayUnits* (boolean): Export values using display units.
- *importFromExternalDataSource* (boolean, default true): import data from external data source (Archive). (since 2017.02)
- *timeStepId (string)*: filter time series by the timestep that has been configured in the TimeSteps.xml. (since 2018.02). N.B.: It is not required to use the timeStepId's in the filter configurations to be able to use them as long as they have been configured in the TimeSteps.xml.
- *timeStepMillis (integer)*: filter time series by an equidistant timestep with a duration of the given number of milliseconds. This can be used in cases where no timestep Id has been configured in TimeSteps.xml. Cannot be used in combination with timeStepId
- matchAsQualifierSet (boolean): Optional argument. Indicates if the selected time series sets must contain all of the the given qualifiers and are not allowed to have additional qualifier ids.
- defaultRequestParametersId(string): Since 2022.02. If this parameter is set, a defaultRequestParameters element is expected to be configured with the specified id in the piRestService element of the WebServices.xml. All request parameters configured will be used as the default request parameters for the timeseries endpoint. If a parameter is explicitly added to the timeseries endpoint, it will overrule the entry that was set in the defaultRequestParameters configuration.

Response

- PI-XML
- PI-JSON
- BINARY: only supported when running in embedded tomcat in the Operator Client or Stand Alone.

DefaultRequestParameters

Since 2022.02 it is possible to configure default request parameters. In the PiServiceConfigFiles/WebServices.xml configuration file multiple defaultRequestParameters can be configured in the piRestService parameter.

If request parameter is not set in the timeseries endpoint and the defaultRequestParametersId is passed to the timeseries endpoint, the value that is set in defaultRequestParameters, is used. If a request parameter is explicitly set in the timeseries endpoint, the value from the defaultRequestParameters is ignored.

For example:

```
<?xml version="1.0" encoding="utf-8"?>
<webServices xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                        xsi:schemaLocation="http://www.wldelft.nl/fews https://fewsdocs.deltares.nl/schemas
/version1.0/webServices.xsd">
        <general>
        <filters>
            <filterId>All</filterId>
        </filters>
        <readonlyMode>false</readonlyMode>
                <testPageEnabled>false</testPageEnabled>
                <cardinalTimeStep unit="hour" multiplier="1"/>
        </general>
        <piRestService>
                <defaultRequestParameters id="id1">
                        <filterId>filter1</filterId>
                        <locationId>loc1</locationId>
                        <locationId>loc2</locationId>
                        <parameterId>par1</parameterId>
                        <parameterId>par2</parameterId>
                        <moduleInstanceId>mod1</moduleInstanceId>
                        <moduleInstanceId>mod2</moduleInstanceId>
                        <qualifierId>q1</qualifierId>
                        <qualifierId>q2</qualifierId>
                        <ensembleId>1</ensembleId>
```

```
                        <ensembleMemberId>2</ensembleMemberId>
                        <timeStepId>timeStep1</timeStepId>
                        <exportIdMap>idMap</exportIdMap>
                        <exportUnitConversionId>unitConversionId</exportUnitConversionId>
                        <timeZoneName>GMT</timeZoneName>
                </defaultRequestParameters>
                <defaultRequestParameters id="id2">
                        <parameterId>Q.meting</parameterId>
                </defaultRequestParameters>
        </piRestService>
</webServices>
```

Filter combinations

Not all parameters can be combined. The following combinations are commonly used valid combinations of parameters. The main way to filter timeSeries is by using filterIds or taskRunIds.

	filterId **Only the default filter or one of its subfilters can be applied)**	**taskRunIds** **One or more taskRunIds can be specified**	**startTime, endTime**	**startCreationTime, endCreationTime**	**startForecastTime, endForecastTime**
Apply a filter to the time series. The requested period will be set to the current time minus one day and one hour ago until the current time plus one day and one hour	X(1)				
Get all time series created by one or more taskRuns. All time steps of the matching time series are returned.		**X**			
Get the time series created by a taskrun and apply a filter from the Filters configuration. startTime and endTime cannot be specified. The complete time series will be returned. Since 2020.01.	X	**X**			
Apply a filter to the time series and return time steps that are in the startTime and endTime range. If the startTime or endTime doesn't match a timestamp of the time series, the closest time step before startTime and/or after endTime is returned as well.	X		X		
Apply a filter to the time series. Only time series created during the startCreationTime and endCreationTime period will be returned. All time steps of the matching time series are returned.	X			X	
Apply a filter to the time series. Return only time series created during the creation time period. Only return timesteps in the startTime and endTime range.	X		X	X	
Apply a filter to the time series. Return only time series with external forecast times in the startForecastTime and endForecastTime period. Only return timesteps in the startTime and endTime range.	X		X		X
Apply a filter to the time series. Return only time series with external forecast times in the startForecastTime and endForecastTime period that were created in the creation time period. Only return time steps in the startTime and endTime range.	X		X	X	X
Apply a filter to the time series. Return only time series created during the creation time period. All time steps of the matching time series are returned. (before 2020.01 startTime and endTime had to be specified).	X			X	
Apply a filter to the time series. Return only time series with external forecast times in the startForecastTime and endForecastTime period. All time steps of the matching time series are returned.	X				X
Apply a filter to the time series. Return only time series created during the creation time period and with external forecast times in the startForecastTime and	X			X	X

endForecastTime period. All time steps of the matching time series are returned.					

(1): Take note that if no startTime and endTime are specified, the requested period will be set to the current time minus one day and one hour ago until the current time plus one day and one hour. If only the startTime is specified, the requested period will be set to the startTime until the startTime time plus one day and one hour. If only the endTime is specified, the requested period will be set to the endTime minus one day and one hour until the endTime.

no data vs no time series

If a timeseries query has matching timeseries sets a http 200 code will be returned and the headers of all matching timeseries sets will be returned.
If there is any data for the requested period, the headers will be followed by the actual events that contain the data. So even if no data is available for the requested period,the headers are always returned.

It is also possible that a timeseries query doesn't match any time series sets at all. This is seen as in invalid request and will result in a HTTP 400 response code.
The following are examples of use cases where this might occur:

- query parameters don't occur in filter. For example: the default filter has subfilters: filterA and filterB. filterA contains timeseries sets with module instance id moduleInstanceA and filterB contains timeseries sets with module instance id moduleInstanceB. If a timeseries query is done with parameters: filterId=filterA and moduleInstanceId=moduleInstanceB, this will return in a HTTP 400 response
- no timeseries for creation period. For example: if a query is using startCreationTime and endCreationTime and no time series have been produced during that period, this is seen as an invalid request and a HTTP 400 response is returned.

availability of new timeseries

When new timeseries have been created, it can take some time before they can be found by the WebServices. The web services updates its indexes every second (every five seconds before 2023.01). Once the indexes have been updated, newly created time series can be found. So it typically can take a few seconds before newly created time series can be found.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/timeseries?
filterId=Netherlands&moduleInstanceIds=ImportObserved&startTime=2013-01-01T00:00:00Z&endTime=2014-01-01T00:00:
00Z&convertDatum=false&useDisplayUnits=false&showThresholds=true&omitMissing=true&onlyHeaders=false&showEnsemble
MemberIds=false&documentVersion=1.23&forecastCount=1"
```

Example PI-XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<TimeSeries xmlns="http://www.wldelft.nl/fews/PI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas
/pi_timeseries.xsd" version="1.23" xmlns:fs="http://www.wldelft.nl/fews/fs">
    <timeZone>0.0</timeZone>
    <series>
        <header>
            <type>instantaneous</type>
            <moduleInstanceId>ImportObserved</moduleInstanceId>
            <locationId>63306260000</locationId>
            <parameterId>T.obs.mean</parameterId>
            <timeStep unit="nonequidistant"/>
            <startDate date="2013-01-01" time="00:00:00"/>
            <endDate date="2014-01-01" time="00:00:00"/>
            <missVal>-999.0</missVal>
            <stationName>DE BILT</stationName>
            <lat>52.1</lat>
            <lon>5.18</lon>
            <x>5.18</x>
            <y>52.1</y>
            <z>15.0</z>
            <units>oC</units>
        </header>
        <event date="2013-01-01" time="00:00:00" value="2" flag="0"/>
        <event date="2013-02-01" time="00:00:00" value="1.7" flag="0"/>
        <event date="2013-03-01" time="00:00:00" value="2.5" flag="0"/>
        <event date="2013-04-01" time="00:00:00" value="8.1" flag="0"/>
        <event date="2013-05-01" time="00:00:00" value="11.5" flag="0"/>
        <event date="2013-06-01" time="00:00:00" value="15.3" flag="0"/>
        <event date="2013-07-01" time="00:00:00" value="19.2" flag="0"/>
        <event date="2013-08-01" time="00:00:00" value="18.1" flag="0"/>
        <event date="2013-09-01" time="00:00:00" value="14.4" flag="0"/>
        <event date="2013-10-01" time="00:00:00" value="12.2" flag="0"/>
```

```
        <event date="2013-11-01" time="00:00:00" value="6.7" flag="0"/>
        <event date="2013-12-01" time="00:00:00" value="5.9" flag="0"/>
        <event date="2014-01-01" time="00:00:00" value="5.7" flag="0"/>
    </series>
    <series>
        <header>
            <type>instantaneous</type>
            <moduleInstanceId>ImportObserved</moduleInstanceId>
            <locationId>63306380000</locationId>
            <parameterId>T.obs.mean</parameterId>
            <timeStep unit="nonequidistant"/>
            <startDate date="2013-01-01" time="00:00:00"/>
            <endDate date="2014-01-01" time="00:00:00"/>
            <missVal>-999.0</missVal>
            <stationName>MAASTRICHT AP</stationName>
            <lat>50.92</lat>
            <lon>5.78</lon>
            <x>5.78</x>
            <y>50.92</y>
            <z>116.0</z>
            <units>oC</units>
        </header>
        <event date="2013-01-01" time="00:00:00" value="1.8" flag="0"/>
        <event date="2013-02-01" time="00:00:00" value="0.8" flag="0"/>
        <event date="2013-03-01" time="00:00:00" value="2.4" flag="0"/>
        <event date="2013-04-01" time="00:00:00" value="9" flag="0"/>
        <event date="2013-05-01" time="00:00:00" value="11.5" flag="0"/>
        <event date="2013-06-01" time="00:00:00" value="15.7" flag="0"/>
        <event date="2013-07-01" time="00:00:00" value="20" flag="0"/>
        <event date="2013-08-01" time="00:00:00" value="18.5" flag="0"/>
        <event date="2013-09-01" time="00:00:00" value="14.4" flag="0"/>
        <event date="2013-10-01" time="00:00:00" value="12.5" flag="0"/>
        <event date="2013-11-01" time="00:00:00" value="5.8" flag="0"/>
        <event date="2013-12-01" time="00:00:00" value="5.6" flag="0"/>
        <event date="2014-01-01" time="00:00:00" value="5.7" flag="0"/>
    </series>
</TimeSeries>
```

PI-XML example with useMilliseconds enabled

```
<?xml version="1.0" encoding="UTF-8"?>
<TimeSeries xmlns="http://www.wldelft.nl/fews/PI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas
/pi_timeseries.xsd" version="1.23" xmlns:fs="http://www.wldelft.nl/fews/fs">
    <timeZone>0.0</timeZone>
    <series>
        <header>
            <type>instantaneous</type>
            <moduleInstanceId>ImportObserved</moduleInstanceId>
            <locationId>63306260000</locationId>
            <parameterId>T.obs.mean</parameterId>
            <timeStep unit="nonequidistant"/>
            <startDate date="2013-01-01" time="00:00:00"/>
            <endDate date="2014-01-01" time="00:00:00"/>
            <missVal>-999.0</missVal>
            <stationName>DE BILT</stationName>
            <lat>52.1</lat>
            <lon>5.18</lon>
            <x>5.18</x>
            <y>52.1</y>
            <z>15.0</z>
            <units>oC</units>
        </header>
        <event date="2013-01-01" time="00:00:00.250" value="2" flag="0"/>
        <event date="2013-02-01" time="00:00:00.250" value="1.7" flag="0"/>
        <event date="2013-03-01" time="00:00:00.250" value="2.5" flag="0"/>
        <event date="2013-04-01" time="00:00:00.250" value="8.1" flag="0"/>
        <event date="2013-05-01" time="00:00:00.250" value="11.5" flag="0"/>
        <event date="2013-06-01" time="00:00:00.250" value="15.3" flag="0"/>
```

```
        <event date="2013-07-01" time="00:00:00.250" value="19.2" flag="0"/>
        <event date="2013-08-01" time="00:00:00.250" value="18.1" flag="0"/>
        <event date="2013-09-01" time="00:00:00.250" value="14.4" flag="0"/>
        <event date="2013-10-01" time="00:00:00.250" value="12.2" flag="0"/>
        <event date="2013-11-01" time="00:00:00.250" value="6.7" flag="0"/>
        <event date="2013-12-01" time="00:00:00.250" value="5.9" flag="0"/>
        <event date="2014-01-01" time="00:00:00.250" value="5.7" flag="0"/>
    </series>
</TimeSeries>
```

PI-XML example with showProducts enabled (snippet).

```
<product id="aProduct" name="A Product">
    <productDate date="2019-06-25" time="11:31:59"/>
    <category id="Meteo" name="Meteo"/>
    <productInfo>
        <user>Rudie Ekkelenkamp</user>
        <confidence>MEDIUM</confidence>
        <classification>INTERNAL</classification>
        <comment>My product comment</comment>
    </productInfo>
</product>
```

Example PI-JSON response

```
{
  "version" : "1.23",
  "timeZone" : "0.0",
  "timeSeries" : [ {
    "header" : {
      "type" : "instantaneous",
      "moduleInstanceId" : "ImportObserved",
      "locationId" : "63306260000",
      "parameterId" : "T.obs.mean",
      "timeStep" : {
        "unit" : "nonequidistant"
      },
      "startDate" : {
        "date" : "2013-01-01",
        "time" : "00:00:00"
      },
      "endDate" : {
        "date" : "2014-01-01",
        "time" : "00:00:00"
      },
      "missVal" : "-999.0",
      "stationName" : "DE BILT",
      "lat" : "52.1",
      "lon" : "5.18",
      "x" : "5.18",
      "y" : "52.1",
      "z" : "15.0",
      "units" : "oC"
    },
    "events" : [ {
      "date" : "2013-01-01",
      "time" : "00:00:00",
      "value" : "2",
      "flag" : "0"
    }, {
      "date" : "2013-02-01",
      "time" : "00:00:00",
      "value" : "1.7",
      "flag" : "0"
    }, {
      "date" : "2013-03-01",
```

```
    "time" : "00:00:00",
    "value" : "2.5",
    "flag" : "0"
  }, {
    "date" : "2013-04-01",
    "time" : "00:00:00",
    "value" : "8.1",
    "flag" : "0"
  }, {
    "date" : "2013-05-01",
    "time" : "00:00:00",
    "value" : "11.5",
    "flag" : "0"
  }, {
    "date" : "2013-06-01",
    "time" : "00:00:00",
    "value" : "15.3",
    "flag" : "0"
  }, {
    "date" : "2013-07-01",
    "time" : "00:00:00",
    "value" : "19.2",
    "flag" : "0"
  }, {
    "date" : "2013-08-01",
    "time" : "00:00:00",
    "value" : "18.1",
    "flag" : "0"
  }, {
    "date" : "2013-09-01",
    "time" : "00:00:00",
    "value" : "14.4",
    "flag" : "0"
  }, {
    "date" : "2013-10-01",
    "time" : "00:00:00",
    "value" : "12.2",
    "flag" : "0"
  }, {
    "date" : "2013-11-01",
    "time" : "00:00:00",
    "value" : "6.7",
    "flag" : "0"
  }, {
    "date" : "2013-12-01",
    "time" : "00:00:00",
    "value" : "5.9",
    "flag" : "0"
  }, {
    "date" : "2014-01-01",
    "time" : "00:00:00",
    "value" : "5.7",
    "flag" : "0"
  } ]
}, {
  "header" : {
    "type" : "instantaneous",
    "moduleInstanceId" : "ImportObserved",
    "locationId" : "63306380000",
    "parameterId" : "T.obs.mean",
    "timeStep" : {
      "unit" : "nonequidistant"
    },
    "startDate" : {
      "date" : "2013-01-01",
      "time" : "00:00:00"
    },
    "endDate" : {
      "date" : "2014-01-01",
      "time" : "00:00:00"
    },
```

```
  "missVal" : "-999.0",
  "stationName" : "MAASTRICHT AP",
  "lat" : "50.92",
  "lon" : "5.78",
  "x" : "5.78",
  "y" : "50.92",
  "z" : "116.0",
  "units" : "oC"
},
"events" : [ {
  "date" : "2013-01-01",
  "time" : "00:00:00",
  "value" : "1.8",
  "flag" : "0"
}, {
  "date" : "2013-02-01",
  "time" : "00:00:00",
  "value" : "0.8",
  "flag" : "0"
}, {
  "date" : "2013-03-01",
  "time" : "00:00:00",
  "value" : "2.4",
  "flag" : "0"
}, {
  "date" : "2013-04-01",
  "time" : "00:00:00",
  "value" : "9",
  "flag" : "0"
}, {
  "date" : "2013-05-01",
  "time" : "00:00:00",
  "value" : "11.5",
  "flag" : "0"
}, {
  "date" : "2013-06-01",
  "time" : "00:00:00",
  "value" : "15.7",
  "flag" : "0"
}, {
  "date" : "2013-07-01",
  "time" : "00:00:00",
  "value" : "20",
  "flag" : "0"
}, {
  "date" : "2013-08-01",
  "time" : "00:00:00",
  "value" : "18.5",
  "flag" : "0"
}, {
  "date" : "2013-09-01",
  "time" : "00:00:00",
  "value" : "14.4",
  "flag" : "0"
}, {
  "date" : "2013-10-01",
  "time" : "00:00:00",
  "value" : "12.5",
  "flag" : "0"
}, {
  "date" : "2013-11-01",
  "time" : "00:00:00",
  "value" : "5.8",
  "flag" : "0"
}, {
  "date" : "2013-12-01",
  "time" : "00:00:00",
  "value" : "5.6",
  "flag" : "0"
}, {
  "date" : "2014-01-01",
```

```
      "time" : "00:00:00",
      "value" : "5.7",
      "flag" : "0"
    } ]
  } ]
}
```

Example Json with milliseconds:

```
{
  "version" : "1.23",
  "timeZone" : "0.0",
  "timeSeries" : [ {
    "header" : {
      "type" : "instantaneous",
      "moduleInstanceId" : "ImportObserved",
      "locationId" : "63306260000",
      "parameterId" : "T.obs.mean",
      "timeStep" : {
        "unit" : "nonequidistant"
      },
      "startDate" : {
        "date" : "2013-01-01",
        "time" : "00:00:00"
      },
      "endDate" : {
        "date" : "2014-01-01",
        "time" : "00:00:00"
      },
      "missVal" : "-999.0",
      "stationName" : "DE BILT",
      "lat" : "52.1",
      "lon" : "5.18",
      "x" : "5.18",
      "y" : "52.1",
      "z" : "15.0",
      "units" : "oC"
    },
    "events" : [ {
      "date" : "2013-01-01",
      "time" : "00:00:00.250",
      "value" : "2",
      "flag" : "0"
    }, {
      "date" : "2013-02-01",
      "time" : "00:00:00.250",
      "value" : "1.7",
      "flag" : "0"
    }, {
      "date" : "2013-03-01",
      "time" : "00:00:00.250",
      "value" : "2.5",
      "flag" : "0"
    }, {
      "date" : "2013-04-01",
      "time" : "00:00:00.250",
      "value" : "8.1",
      "flag" : "0"
    }, {
      "date" : "2013-05-01",
      "time" : "00:00:00.250",
      "value" : "11.5",
      "flag" : "0"
    }, {
      "date" : "2013-06-01",
      "time" : "00:00:00.250",
      "value" : "15.3",
      "flag" : "0"
    }, {
      "date" : "2013-07-01",
```

```
      "time" : "00:00:00.250",
      "value" : "19.2",
      "flag" : "0"
    }, {
      "date" : "2013-08-01",
      "time" : "00:00:00.250",
      "value" : "18.1",
      "flag" : "0"
    }, {
      "date" : "2013-09-01",
      "time" : "00:00:00.250",
      "value" : "14.4",
      "flag" : "0"
    }, {
      "date" : "2013-10-01",
      "time" : "00:00:00.250",
      "value" : "12.2",
      "flag" : "0"
    }, {
      "date" : "2013-11-01",
      "time" : "00:00:00.250",
      "value" : "6.7",
      "flag" : "0"
    }, {
      "date" : "2013-12-01",
      "time" : "00:00:00.250",
      "value" : "5.9",
      "flag" : "0"
    }, {
      "date" : "2014-01-01",
      "time" : "00:00:00.250",
      "value" : "5.7",
      "flag" : "0"
    } ]
  }]
 }
```

Example TimeSteps.xml using OGC compliant periods

The following is an example of the TimeSteps.xml from the RegionConfig folder. For example to search for time series with a 3 hourly time step, the parameter timeStepId=PT3H can be added to the request.

```
<?xml version="1.0" encoding="UTF-8"?>
<timeSteps xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews http://fews.wldelft.nl/schemas/version1.0/timeSteps.xsd">
    <!--
      Configure well known timesteps as [ISO 8601:1988(E)] standards.
      See also:
      http://cite.opengeospatial.org/OGCTestData/wms/1.1.1/spec/wms1.1.1.html#date_time.time
    -->
   <!-- Yearly time steps -->
   <timeStep id="P1Y" multiplier="1" unit="year"/>
   <!-- Monthly time steps -->
   <timeStep id="P1M" multiplier="1" unit="month"/>
   <!-- Daily time steps -->
   <timeStep id="PT1D" multiplier="1" unit="day"/>
   <timeStep id="PT5D" multiplier="5" unit="day"/>
   <!-- Hourly time steps -->
   <timeStep id="PT1H" multiplier="1" unit="hour"/>
   <timeStep id="PT2H" multiplier="2" unit="hour"/>
   <timeStep id="PT3H" multiplier="3" unit="hour"/>
   <timeStep id="PT6H" multiplier="6" unit="hour"/>
   <!-- Minute based time steps -->
   <timeStep id="PT1M" multiplier="1" unit="minute"/>
   <timeStep id="PT5M" multiplier="5" unit="minute"/>
```

```
   <timeStep id="PT10M" multiplier="10" unit="minute"/>
   <timeStep id="PT15M" multiplier="15" unit="minute"/>
   <timeStep id="PT30M" multiplier="30" unit="minute"/>
   <timeStep id="PT60M" multiplier="60" unit="minute"/>
   <!-- Second based time steps -->
   <timeStep id="PT1S" multiplier="1" unit="second"/>
   <timeStep id="PT5S" multiplier="5" unit="second"/>
</timeSteps>
```

PI-XML example with detection limit symbols (since 2020.02)

In case limit a detection limit is available in for an event, the detection attribute will be given as part of the event element. Possible values are: <>~

```
<event date="2015-01-07" time="23:00:00" value="0.11" detection="&lt;" flag="0"/>
<event date="2015-01-21" time="23:00:00" value="0.09" detection="~" flag="0"/>
```

PI-JSON example with detection limit symbols (since 2020.02)

In case limit a detection limit is available in for an event, the detection attribute will be given as part of the event element. Possible values are: <>~

```
  "events" : [ {
      "date" : "2015-01-07",
      "time" : "23:00:00",
      "value" : "0.11",
      "detection" : "<",
      "flag" : "0"
    }, {
      "date" : "2015-01-21",
      "time" : "23:00:00",
      "value" : "0.09",
      "detection" : "~",
      "flag" : "0"
    }
]
```

GET timeseries/displaygroups (2019.02)

Returns a pi timeseries xml or json file containing the timeseries data filtered by a plotId of the DisplayGroups.xml configuration in the SystemConfigFiles folder. See also http://fews.wldelft.nl/schemas/version1.0/displayGroups.xsd.

The TimeSeriesSets configured for a plotId will be used to filter the timeSeries. The **line, area** and **bar** elements are used when determining the relevant TimeSeriesSets. In case of forecasts, this means only the current forecast will be retrieved. It is not possible to request older forecasts.

> If no line, area or bar elements are used in the displayGroups.xml configuration, the TimeSeriesSets will not be applied.

Request parameters

- *convertDatum* (boolean): Convert values from relative location height to absolute height values.
- *documentVersion* (string, 1.9 or up): File format version (optional). For example: 1.23
- *documentFormat* (string): PI_XML (default), PI_JSON, DD_JSON, NOOS_TEXT
- *endTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): End time of search period that looks for timeseries values that are within this period. If the endTime doesn't match a timestamp of the time series, the closest timestamp after the endTime, will also be returned. Format: yyyy-MM-ddTHH:mm:ssZ.
- *locationIds* (string): Subset of locations for which to retrieve timeseries. This parameter can be duplicated to use multiple locationIds. At least one location id is required.
- *omitMissing* (boolean): Toggle omitting or returning of missing values in response
- *omitEmptyTimeSeries* (boolean): Toggle omitting or returning headers of empty timeSeries. Default is false. Since 2020.02.
- *onlyHeaders* (boolean): Toggle to return only header information or also data
- *onlyForecasts* (boolean): Toggle to return only forecast timeSeries (Since 2017.02)
- onlyManualEdits (boolean): Toggle to return only manual edits.
- **plotId** (string, required): the plotId as configured in the DisplayGroups.xml for which the configured TimeSeriesSets will be determined from the line, area and bar elements.
- showProducts (boolean): Toggle to display product information that is assigned to a forecast. (Since 2019.02).
- *showStatistics* (boolean): Toggle to return statistics information about timeseries. Typically used in combination with onlyHeaders. Returns additional information about data availability of timeseries (Since 2015.01).
    - firstValueTime: First time with a value in the timeSeries

    - lastValueTime: Last time with a value in the timeSeries
    - maxValue: Maximum value in the timeSeries
    - minValue: Minimum value in the timeSeries
    - valueCount: Number of values in the timeSeries
- *showThresholds* (boolean): Option to toggle the returning of threshold information in the headers
- *showEnsembleMemberIds* (boolean): Show ensemble member ids.
- thinning (long): Thinning is used to retrieve the visually interesting time steps of a timeSeries. It tries to keep the peaks and gaps and minimizes the number of time steps that have to be retrieved. It is typically used for visualizations. The value to be specified should be equal to the view period of the timeSeries that is visualized divided by the number of pixels that are available for display. For example: visualizing a view period of 5 years (157784760000 milliseconds) on a display of 1024, the thinning parameter should be set to 157784760000/1024 = 15408668. (Since 2019.02)
- *useMilliseconds* (boolean) Optional argument. Default is false. If it is set to true, the response will contain milliseconds. See example below
- *startTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): Start time of search period that looks for timeseries values that are within this period. If the startTime doesn't match a timestamp of the time series, the closest timestamp before the startTime, will also be returned. Format: yyyy-MM-ddTHH:mm:ssZ.
- *useDisplayUnits* (boolean): Export values using display units.

Response

- Timeseries PI-XML or PI-JSON file content..

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/timeseries/displaygroups?
plotId=TidalForecasts&locationIds=UKCFF_ABD&startTime=2019-06-10T00:00:00Z&endTime=2019-06-25T00:00:00Z"
```

Example PI-XML and JSON response

See the timeseries endpoint for examples.

GET timeseries/grid/actions (2023.02)

The timeseries/grid endpoint is intended to be used together with the Delft-FEWS WMS service. It returns the details about how to display the timeseries from a grid cell for a request period.

The time series itself can be retrieved using the endpoint timeseries/grid

Request parameters

- **bbox** (string, required): bounding box of map that is viewed in , separated EPSG:3857 format. The order of the coordinates is as follows: bottom left X, bottom left Y, top right X, top right Y. For example: bbox=-1558755.612890017,4979850.04379049,1623657.8112034467,6709422.556884765
- *documentVersion* (string, 1.9 or up): File format version (optional). For example: 1.23
- *documentFormat* (string): PI_XML (default), PI_JSON, DD_JSON, NOOS_TEXT
- elevation (double): used for 3d data, like for example water depth, to get the timeseries of a grid point at a specific elevation. Since 2020.01.
- ***endTime*** (dateTime: yyyy-MM-ddTHH:mm:ssZ, required): End time of search period that looks for timeseries values that are within this period. If the endTime doesn't match a timestamp of the time series, the closest timestamp after the endTime, will also be returned. Format: yyyy-MM-ddTHH:mm:ssZ.
- *externalForecastTime* (dateTime format: yyyy-MM-ddTHH:mm:ssZ): Time value of external forecast time.
- *ensembleId* (String): Used in combination with ensembleMemberId to identify a unique ensemble. Since 2020.01.
- *ensembleMemberId* (String): Used in combination with ensembleId to identify a unique ensemble. Since 2020.01.
- *importFromExternalDataSource* (boolean, default true): import data from external data source (Archive). since 2019.02.
- **layers** (string, required): layerd id (only one layer is supported and required) that matches the gridPlot id as configured in the gridDisplay. Every gridPlot that has been configured in the grid display configuration represents a WMS layer. For more information, see the WMS Service documentation: FEWS Web Mapping Service with time support: WMS-T
- showVerticalProfile (boolean): Show vertical profile in case of 3D data. Since 2020.01.
- ***startTime*** (dateTime: yyyy-MM-ddTHH:mm:ssZ, required): Start time of search period that looks for timeseries values that are within this period. If the startTime doesn't match a timestamp of the time series, the closest timestamp before the startTime, will also be returned. Format: yyyy-MM-ddTHH:mm:ssZ.
- **x** (double, required): x position on the map in EPSG:3857 format.
- **y** (double, required): y position on the map in EPSG:3857 format.

GET timeseries/grid/actions (2023.02)

The timeseries/grid endpoint is intended to be used together with the Delf-FEWS WMS service. Every layer that is provided by the WMS service, can be used with this endpoint. See also FEWS Web Mapping Service with time support: WMS-T.

Returns a pi timeseries xml file containing the timeseries data from a grid cell for a request period. The grid is specified by passing a layerId. The grid cell is determined by specifying a x and y coordinate and a bounding box. Currenly only EPSG:3857 is supported for the x,y, and bounding box coordinates. At least a layer, startTime, endTime, x,y and bounding box are required.

Request parameters

- **bbox** (string, required): bounding box of map that is viewed in , separated EPSG:3857 format. The order of the coordinates is as follows: bottom left X, bottom left Y, top right X, top right Y. For example: bbox=-1558755.612890017,4979850.04379049,1623657.8112034467,6709422.556884765
- *convertDatum* (boolean): Convert values from relative location height to absolute height values.
- *documentVersion* (string, 1.9 or up): File format version (optional). For example: 1.23
- *documentFormat* (string): PI_XML (default), PI_JSON, DD_JSON, NOOS_TEXT
- elevation (double): used for 3d data, like for example water depth, to get the timeseries of a grid point at a specific elevation. Since 2020.01.
- ***endTime*** (dateTime: yyyy-MM-ddTHH:mm:ssZ, required): End time of search period that looks for timeseries values that are within this period. If the endTime doesn't match a timestamp of the time series, the closest timestamp after the endTime, will also be returned. Format: yyyy-MM-ddTHH:mm:ssZ.
- *externalForecastTime* (dateTime format: yyyy-MM-ddTHH:mm:ssZ): Time value of external forecast time.
- *ensembleId* (String): Used in combination with ensembleMemberId to identify a unique ensemble. Since 2020.01.
- *ensembleMemberId* (String): Used in combination with ensembleId to identify a unique ensemble. Since 2020.01.
- *importFromExternalDataSource* (boolean, default true): import data from external data source (Archive). since 2019.02.
- **layers** (string, required): layerd id (only one layer is supported and required) that matches the gridPlot id as configured in the gridDisplay. Every gridPlot that has been configured in the grid display configuration represents a WMS layer. For more information, see the WMS Service documentation: FEWS Web Mapping Service with time support: WMS-T
- *omitMissing* (boolean): Toggle omitting or returning of missing values in response
- *omitEmptyTimeSeries* (boolean): Toggle omitting or returning headers of empty timeSeries. Default is false. Since 2020.02.
- *onlyHeaders* (boolean): Toggle to return only header information or also data
- *onlyForecasts* (boolean): Toggle to return only forecast timeSeries
- onlyManualEdits (boolean): Toggle to return only manual edits.
- showProducts (boolean): Toggle to display product information that is assigned to a forecast. (Since 2019.02). See below for an example.
- *showStatistics* (boolean): Toggle to return statistics information about timeseries. Typically used in combination with onlyHeaders. Returns additional information about data availability of timeseries (Since 2015.01).
    - firstValueTime: First time with a value in the timeSeries
    - lastValueTime: Last time with a value in the timeSeries
    - maxValue: Maximum value in the timeSeries
    - minValue: Minimum value in the timeSeries
    - valueCount: Number of values in the timeSeries
- *showThresholds* (boolean): Option to toggle the returning of threshold information in the headers
- *showEnsembleMemberIds* (boolean): Show ensemble member ids.
- showVerticalProfile (boolean): Show vertical profile in case of 3D data. Since 2020.01.
- ***startTime*** (dateTime: yyyy-MM-ddTHH:mm:ssZ, required): Start time of search period that looks for timeseries values that are within this period. If the startTime doesn't match a timestamp of the time series, the closest timestamp before the startTime, will also be returned. Format: yyyy-MM-ddTHH:mm:ssZ.
- thinning (long): unit ms/pixel. Thinning is used to retrieve the visually interesting time steps of timeSeries. It tries to keep the peaks and gaps and minimizes the number of time steps that have to be retrieved. It is typically used for visualizations. The value to be specified should be equal to the view period in milliseconds of the timeSeries that is visualized divided by the number of pixels that are available for display. For example: visualizing a view period of 5 years (157784760000 milliseconds) on a display of 1024 pixels, the thinning parameter should be set to 157784760000/1024 = 15408668. (Since 2019.02)
- *useMilliseconds* (boolean) Optional argument. Default is false. If it is set to true, the response will contain milliseconds. See example below. Since 2019.02
- *useDisplayUnits* (boolean): Export values using display units.
- *convertDatum* (boolean): Export values using display units. Also use displayUnits in the generated requests
- **x** (double, required): x position on the map in EPSG:3857 format.
- **y** (double, required): y position on the map in EPSG:3857 format.

Response

- Actions in PI-JSON file content.

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/timeseries/grid/actions?
documentFormat=PI_JSON&layers=Temp_forecast_nwp&x=-121480.85450867479&y=6072939.872066073&startTime=2020-01-
08T01:00:00.000Z&endTime=2020-01-
08T01:00:00.000Z&bbox=-1558755.612890017,4979850.04379049,1623657.8112034467,6709422.556884765"
```

GET timeseries/grid (2018.02)

The timeseries/grid endpoint is intended to be used together with the Delf-FEWS WMS service. Every layer that is provided by the WMS service, can be used with this endpoint. See also FEWS Web Mapping Service with time support: WMS-T.

Returns a pi timeseries xml file containing the timeseries data from a grid cell for a request period. The grid is specified by passing a layerId. The grid cell is determined by specifying a x and y coordinate and a bounding box. Currenly only EPSG:3857 is supported for the x,y, and bounding box coordinates. At least a layer, startTime, endTime, x,y and bounding box are required.

Request parameters

- **bbox** (string, required): bounding box of map that is viewed in , separated EPSG:3857 format. The order of the coordinates is as follows: bottom left X, bottom left Y, top right X, top right Y. For example: bbox=-1558755.612890017,4979850.04379049,1623657.8112034467,6709422.556884765
- *convertDatum* (boolean): Convert values from relative location height to absolute height values.
- *documentVersion* (string, 1.9 or up): File format version (optional). For example: 1.23
- *documentFormat* (string): PI_XML (default), PI_JSON, DD_JSON, NOOS_TEXT
- elevation (double): used for 3d data, like for example water depth, to get the timeseries of a grid point at a specific elevation. Since 2020.01.
- ***endTime*** (dateTime: yyyy-MM-ddTHH:mm:ssZ, required): End time of search period that looks for timeseries values that are within this period. If the endTime doesn't match a timestamp of the time series, the closest timestamp after the endTime, will also be returned. Format: yyyy-MM-ddTHH:mm:ssZ.
- *externalForecastTime* (dateTime format: yyyy-MM-ddTHH:mm:ssZ): Time value of external forecast time.
- *ensembleId* (String): Used in combination with ensembleMemberId to identify a unique ensemble. Since 2020.01.
- *ensembleMemberId* (String): Used in combination with ensembleId to identify a unique ensemble. Since 2020.01.
- *importFromExternalDataSource* (boolean, default true): import data from external data source (Archive). since 2019.02.
- **layers** (string, required): layerd id (only one layer is supported and required) that matches the gridPlot id as configured in the gridDisplay. Every gridPlot that has been configured in the grid display configuration represents a WMS layer. For more information, see the WMS Service documentation: FEWS Web Mapping Service with time support: WMS-T
- *omitMissing* (boolean): Toggle omitting or returning of missing values in response
- *omitEmptyTimeSeries* (boolean): Toggle omitting or returning headers of empty timeSeries. Default is false. Since 2020.02.
- *onlyHeaders* (boolean): Toggle to return only header information or also data
- *onlyForecasts* (boolean): Toggle to return only forecast timeSeries
- onlyManualEdits (boolean): Toggle to return only manual edits.
- showProducts (boolean): Toggle to display product information that is assigned to a forecast. (Since 2019.02). See below for an example.
- *showStatistics* (boolean): Toggle to return statistics information about timeseries. Typically used in combination with onlyHeaders. Returns additional information about data availability of timeseries (Since 2015.01).
    - firstValueTime: First time with a value in the timeSeries
    - lastValueTime: Last time with a value in the timeSeries
    - maxValue: Maximum value in the timeSeries
    - minValue: Minimum value in the timeSeries
    - valueCount: Number of values in the timeSeries
- *showThresholds* (boolean): Option to toggle the returning of threshold information in the headers
- *showEnsembleMemberIds* (boolean): Show ensemble member ids.
- showVerticalProfile (boolean): Show vertical profile in case of 3D data. Since 2020.01.
- ***startTime*** (dateTime: yyyy-MM-ddTHH:mm:ssZ, required): Start time of search period that looks for timeseries values that are within this period. If the startTime doesn't match a timestamp of the time series, the closest timestamp before the startTime, will also be returned. Format: yyyy-MM-ddTHH:mm:ssZ.
- thinning (long): unit ms/pixel. Thinning is used to retrieve the visually interesting time steps of timeSeries. It tries to keep the peaks and gaps and minimizes the number of time steps that have to be retrieved. It is typically used for visualizations. The value to be specified should be equal to the view period in milliseconds of the timeSeries that is visualized divided by the number of pixels that are available for display. For example: visualizing a view period of 5 years (157784760000 milliseconds) on a display of 1024 pixels, the thinning parameter should be set to 157784760000/1024 = 15408668. (Since 2019.02)
- *useMilliseconds* (boolean) Optional argument. Default is false. If it is set to true, the response will contain milliseconds. See example below. Since 2019.02
- *useDisplayUnits* (boolean): Export values using display units.
- **x** (double, required): x position on the map in EPSG:3857 format.
- **y** (double, required): y position on the map in EPSG:3857 format.

Response

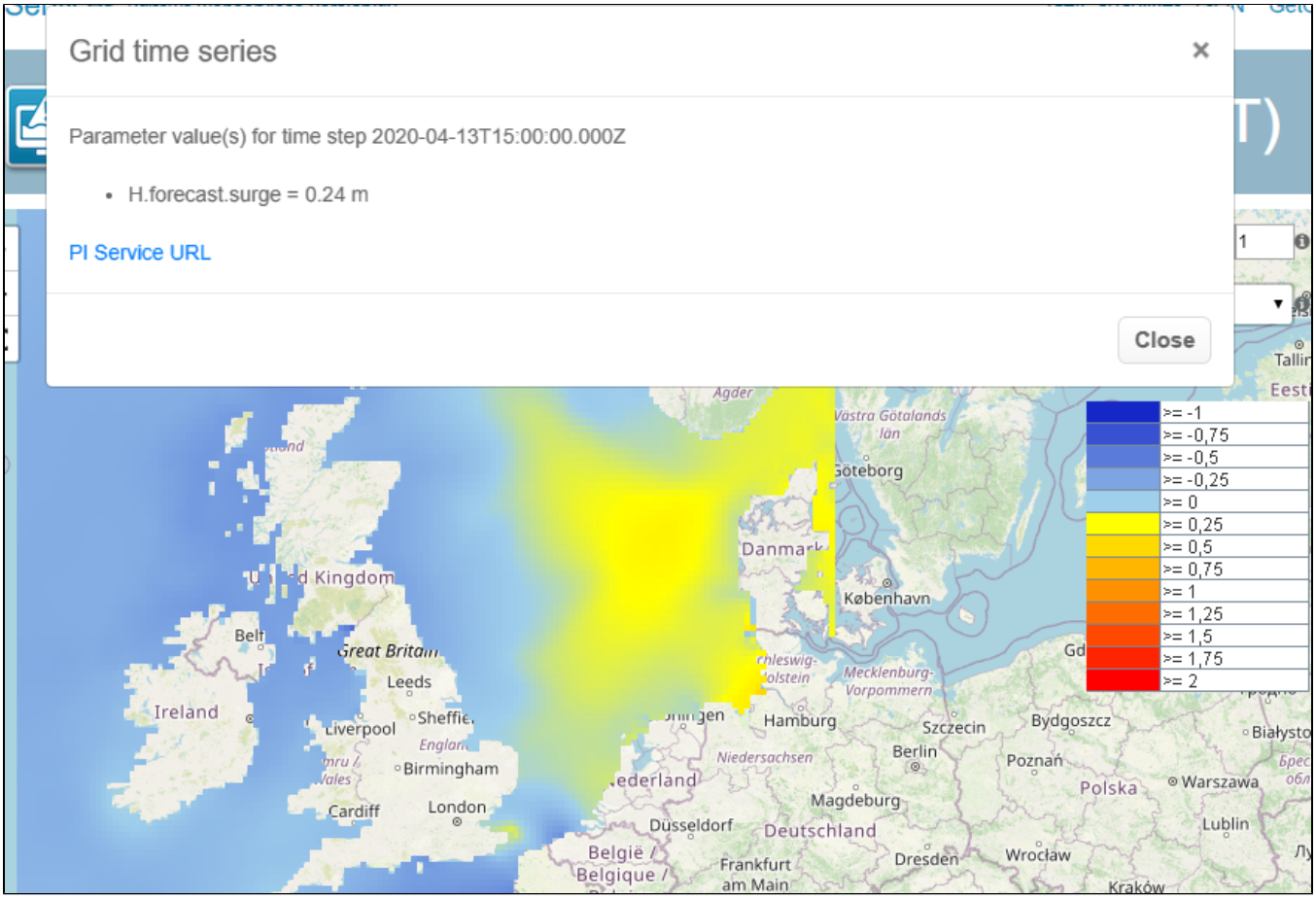
- Timeseries PI-XML or PI-JSON file content.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/timeseries/grid?
documentFormat=PI_JSON&layers=Temp_forecast_nwp&x=-121480.85450867479&y=6072939.872066073&startTime=2020-01-
08T01:00:00.000Z&endTime=2020-01-
08T01:00:00.000Z&bbox=-1558755.612890017,4979850.04379049,1623657.8112034467,6709422.556884765"
```

WMS Test page

This timeSeries grid functionality can also be tested using the WMS service test page. Right clicking on a grid cell, will show a popup that will show the value of the grid cell for the current time step. When clicking on the PI Service URL the PI Request that was used to determine the current value, will be opened.

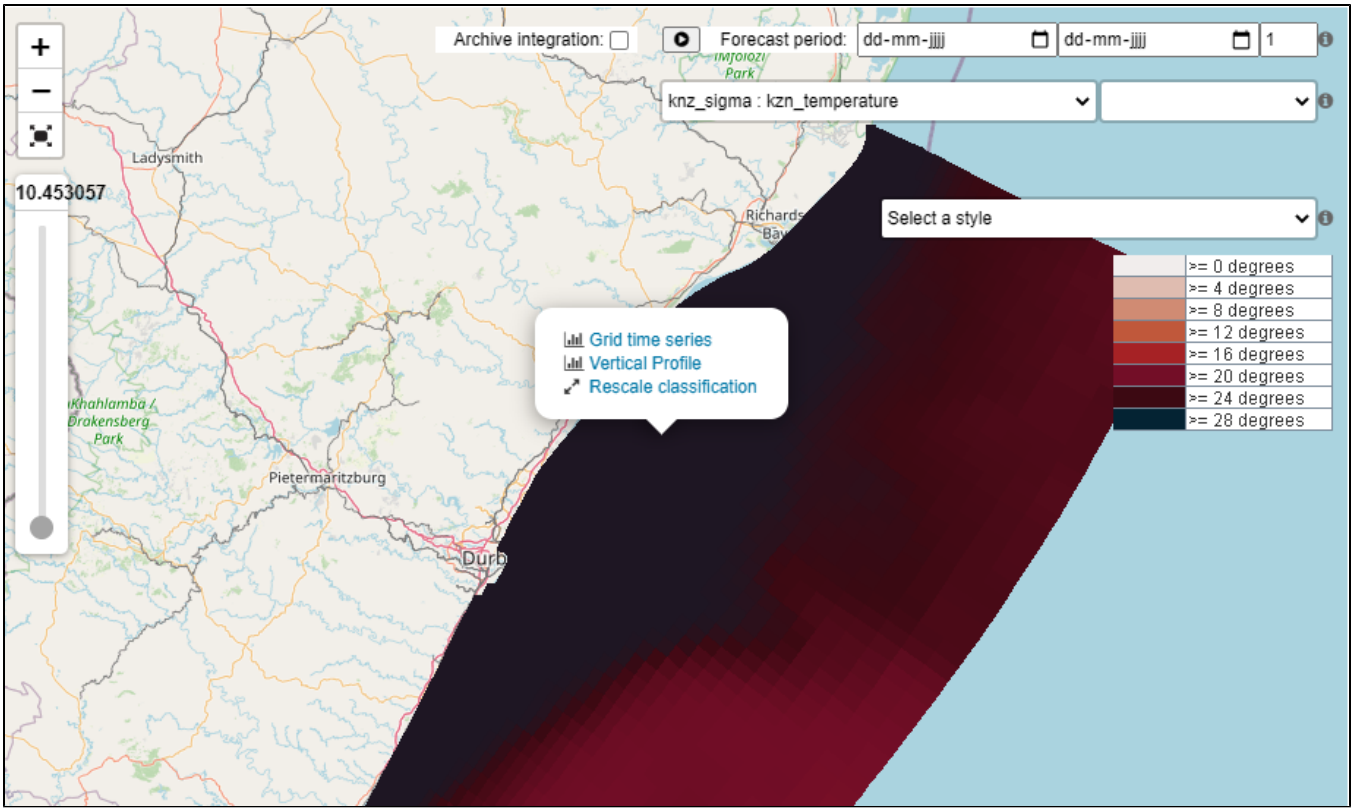


Example PI-XML and JSON response

See the timeseries endpoint for examples.

3D grids elevation and vertical profiles

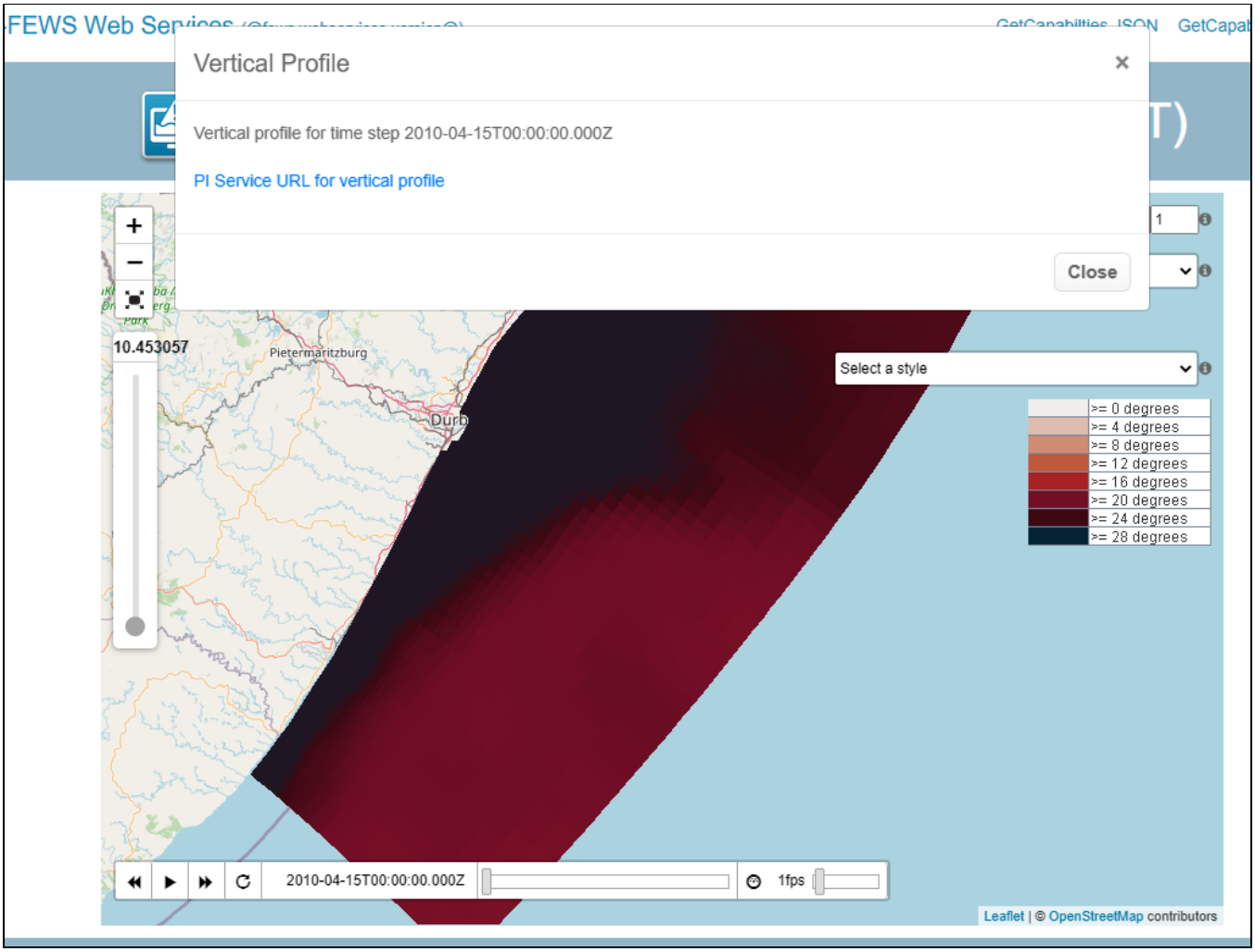
For 3D grids the elevation parameter can be used to get the timeseries of a grid cell at a specified elevation (for example to get the water temperature at a specific water depth). On the WMS testing page this functionality can be tested by right clicking on a grid cell. The vertical slider on the left can be used to specify the elevation that should be used. When selecting the "Grid time series" option, a popup with the PI service URL will be generated that will get the timeseries for a grid cell at the selected elevation.



An example request with elevation:

```
http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/timeseries/grid?
documentFormat=PI_JSON&layers=kzn_temperature&x=3496535.421877103&y=-3484305.4973514746&startTime=2010-04-15T00:
00:00.000Z&endTime=2010-04-15T00:00:00.000Z&bbox=3272727.803058107,-3739299.4237108226,3835304.3312370046,
-3372401.6879419754&_=1602068760367&importFromExternalDataSource=false&elevation=10.453057
```

It is also possible to get a vertical profile for 3D data. On the WMS testing page this functionality can be tested by right clicking on a grid cell. When selecting the "Vertical Profile" option, a popup with the PI service URL will be generated that will get the timeseries of all elevations for a grid cell at the selected timestep. This is achieved by passing the showVerticalProfile=true parameter. It is possible to request vertical profiles for more than one time step by using the startTime and endTime.



An example request to get the vertical profile for one time step:

```
http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/timeseries/grid?
documentFormat=PI_JSON&layers=kzn_temperature&x=3516714.7973443894&y=-3506319.361497606&startTime=2010-04-15T00:
00:00.000Z&endTime=2010-04-15T00:00:00.000Z&bbox=3272727.803058107,-3739299.4237108226,3835304.3312370046,
-3372401.6879419754&_=1602068935896&importFromExternalDataSource=false&showVerticalProfile=true
```

The PI response of profiles will contain domain axes. In this example the water_depth is the domain axis and the different elevations are reported in the domainAxisValues.

Typical profile PI_JSON response


```
{
  "version" : "1.28",
  "timeZone" : "0.0",
  "timeSeries" : [ {
    "header" : {
      "type" : "instantaneous",
      "moduleInstanceId" : "kzn",
      "locationId" : "kzn",
      "parameterId" : "water_temperature",
      "timeStep" : {
        "unit" : "nonequidistant"
      },
      "startDate" : {
        "date" : "2010-04-15",
        "time" : "00:00:00"
      },
      "endDate" : {
```

```
        "date" : "2010-04-15",
        "time" : "00:00:00"
      },
      "forecastDate" : {
        "date" : "2019-10-10",
        "time" : "18:00:00"
      },
      "missVal" : "-999.0",
      "stationName" : "kzn",
      "units" : "degrees",
      "domainAxis" : [ {
        "parameterId" : "water_depth",
        "units" : "m"
      } ],
      "creationDate" : "2020-06-24",
      "creationTime" : "06:19:05",
      "approvedDate" : {
        "date" : "2020-06-24",
        "time" : "06:19:09"
      }
    },
    "domains" : [ {
      "domainAxisValues" : [ {
        "parameterId" : "water_depth",
        "values" : [ [ "36.49779" ], [ "109.4925" ], [ "182.48721" ], [ "255.48279" ], [ "328.4775" ], [
"401.4722" ], [ "474.4678" ], [ "547.4625" ], [ "620.4572" ], [ "693.4528" ], [ "766.4475" ], [ "839.4422" ] ]
      } ],
      "events" : [ {
        "date" : "2010-04-15",
        "time" : "00:00:00",
        "flag" : "2",
        "values" : [ [ "26.1" ], [ "21.0" ], [ "17.7" ], [ "15.2" ], [ "13.3" ], [ "11.7" ], [ "10.6" ], [
"10.0" ], [ "9.5" ], [ "8.8" ], [ "8.1" ], [ "7.4" ] ]
      } ]
    } ]
  } ]
}
```


GET timeseries/intervalstatistics (2023.02)

see https://fewsdocs.deltares.nl/webservices/rest-api/v1/#get-/timeseries/intervalstatistics

GET ensembles/members (2022.02)

Get all ensemble member ids for one or more ensemble Ids. The available member ids depend on an up-to-date index. This is run once a day on a forecasting shell server.

Request parameters

- *ensembleIds* (string, required): one or more ensembleIds for which to get the current ensembleMemberIds.

Example query

```
http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/ensembles/members?ensembleIds=id1&ensembleIds=id2
```

Example Response:


```
{
  "ensembles" : [ {
    "ensembleId" : "id1",
    "ensembleMemberIds" : [ "01", "02" ]
  }, {
    "ensembleId" : "id2",
    "ensembleMemberIds" : [ "01" ]
  } ]
}
```

GET archive/areas (2022.02)

Returns a list of areas that are available in the archive. The list consists of all the areas available in the open archive, netcdf-storages and the archive database.

Request parameters

None

Example query

```
http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/archive/areas
```

Example response

```
{
  "areas" : [ {
    "id" : "area52",
    "name" : "area52"
  }, {
    "id" : "area51",
    "name" : "area51"
  }]
}
```

GET archive/sources (2022.02)

Returns a list of sources that are available in the archive. The list consists of all the sources available in the open archive, netcdf-storages and the archive database.

Request parameters

None

Example query

```
http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/archive/sources
```

Example response

```
{
  "sources" : [ {
    "id" : "Preprocess_NDFD",
    "name" : "Forecast (National Digital Forecast Database)"
  }, {
    "id" : "BrownsFerry_Forecast",
    "name" : "Simulated (HTMS, BrownsFerry Forecast)"
  }, {
    "id" : "Preprocess_US_HTMS",
    "name" : "Simulated (HTMS, Preprocess US)"
  } ]
}
```

GET archive/parameters (2020.01)

Returns a list of parameters that are available in the archive. Optionally the parameters can be filtered by one or more location ids or archive attributes.

Request parameters

- *locationIds* (string): Subset of locations for which to retrieve parameters. This parameter has to be duplicated to specify multiple locations.
- *moduleInstanceIds* (string): Subset of module instances for which to retrieve parameters. This parameter has to be duplicated to specify multiple module instances,
- *areaIds* (string): Area id for which to retrieve parameters. This parameter has to be duplicated to specify multiple areas,
- *sourceIds* (string): Source id for which to retrieve parameters. This parameter has to be duplicated to specify multiple sources,
- *attribute(key)=value* (string): on ore more attributes that have to match the archive attribute. Attributes are passed by passing the key as an argument to the attribute() paramteter and the value as parameter value. See the example where only locations will be returned when an archive attribute storageId was set with the value storageA.
- *documentVersion* (string, 1.9 or up): File format version (optional). For example: 1.23
- *documentFormat* (string): PI_XML (default), PI_JSON, NAME_LIST (since 2021.02) or DD_JSON (since 2021.02).

Example query

```
http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/archive/parameters?attribute(storageId)=storageA
```

Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<timeseriesparameters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews
/PI" xsi:schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas
/pi_timeseriesparameters.xsd" version="1.9">
    <parameter id="QR_Adj">
        <name>QR_Adj</name>
        <parameterType>instantaneous</parameterType>
        <unit></unit>
    </parameter>
</timeseriesparameters>
```

Example documentFormat=NAME_LIST

```
air_pressure
air_temperature
area_of_flow
chloride
chlorophyll-a
discharge
discharge_diurnal
discharge_hourly
discharge_in
discharge_net
discharge_out
eastward_wind
fpptot
northward_wind
number_chutes
opening_height
oxy
precipitation
```

GET archive/moduleinstances (2022.02)

Returns a list of module instances that are available in the netcdf storage.

Request parameters

- *parameterIds* (string): Subset of parameters for which to retrieve module instances. This parameter has to be duplicated to specify multiple parameters,
- *locationIds* (string): Subset of location ids for which to retrieve module instances. This parameter has to be duplicated to specify multiple locations.
- *areaIds* (string): Subset of areas for which to retrieve module instances. This parameter has to be duplicated to specify multiple areas.
- *sourceIds* (string): Subset of sources for which to retrieve module instances. This parameter has to be duplicated to specify multiple sources.

Example query

```
http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/archive/moduleinstances?
locationIds=locA&parameterIds=parA
```

Example response

```
{
  "moduleInstances" : [ {
    "id" : "chasm"
  }, {
    "id" : "import_dcsm_v5_hirlam"
  } ]
}
```

GET archive/locations (2020.01)

Returns a list of locations that are available in the archive. Optionally the locations can be filtered by one or more parameter ids or archive attributes.

Request parameters

- *parameterIds* (string): Subset of parameters for which to retrieve locations. This parameter has to be duplicated to specify multiple parameters,
- *moduleInstanceIds* (string): Subset of module instance ids for which to retrieve locations. This parameter has to be duplicated to specify multiple module instances.
- *areaIds* (string): Subset of areas for which to retrieve locations. This parameter has to be duplicated to specify multiple areas.
- *sourceIds* (string): Subset of sources for which to retrieve locations. This parameter has to be duplicated to specify multiple sources.
- attribute(key)=value (string): on ore more attributes that have to match the archive attribute. Attributes are passed by passing the key as an argument to the attribute() paramteter and the value as parameter value. See the example where only locations will be returned when an archive attribute storageId was set with the value storageA.
- *documentVersion* (string, 1.9 or up): File format version (optional). For example: 1.23
- *documentFormat* (string): PI_XML (default), PI_JSON, GEO_JSON (since 2021.02), NAME_LIST (since 2021.02) or DD_JSON (since 2021.02).

Example query

```
http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/archive/locations?attribute(storageId)=storageA
```

Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<Locations xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas/pi_locations.
xsd" version="1.9">
    <geoDatum>LOCAL</geoDatum>
    <location locationId="BCRA1">
        <shortName>Bear Creek Dam</shortName>
        <lat>34.399166107177734</lat>
        <lon>-87.98777770996094</lon>
    </location>
</Locations>
```

GET archive/attributes (2020.01)

Returns a list of attributes with their values that are available in the archive. Optionally the attributes can be filtered by one or more attribute keys, one or more parameter ids or one or more location ids.

Request parameters

- *parameterIds* (string): Subset of parameters for which to retrieve atributes. This parameter has to be duplicated to specify multiple parameters. (not available yet)
- *locationIds* (string): Subset of locations for which to retrieve attributes. This parameter has to be duplicated to specify multiple locations. (not available yet)
- *attributes* (string): Subset of archive attribute keys. Only attributes with this key will be returned. This parameter has to be duplicated to specify multiple attributes. This parameter is usefull to get all values for a specific attribute.
- *moduleInstanceIds* (string): Subset of module instances for which to retrieve attributes. This parameters has to be duplicated to specify multiple parameters
- *areaIds* (string): Subset of areas for which to retrieve attributes. This parameters has to be duplicated to specify multiple parameters
- *sourceIds* (string): Subset of sources for which to retrieve attributes. This parameters has to be duplicated to specify multiple parameters

- *documentVersion* (string, 1.9 or up): File format version (optional). For example: 1.23
- *documentFormat* (string): PI_XML (default) or PI_JSON

Example query

```
http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/archive/attributes?attributes=storageId
```

Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<archiveAttributes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI"
xsi:schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas
/pi_archive_attributes.xsd">
    <attribute>
        <name>storageId</name>
        <value>storageA</value>
    </attribute>
</archiveAttributes>
```

GET archive/netcdfstorageforecasts (2020.02)

Returns a list of forecasts from the external netcdf storage from the archive.

Request parameters

- *startTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): Start time of the archive search period
- attribute(key)=value (string): on ore more attributes that have to match the archive attribute. Attributes are passed by passing the key as an argument to the attribute() paramteter and the value as parameter value. See the example where only forecasts will be returned were the long_name is equal to parameterA
- *endTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): End time of the archive search period
- *requestedAttributes(string):* The attributes of the forecast which should be included in the response. Repeat the parameter to specify multiple attributes.
- forecastCount (integer): The maximum number of forecasts to be returned from archive. If you only want to download the most recent forecast in the requested period then use forecastCount=1

documentFormat (string): PI_XML (default) or PI_JSON

Example query

```
http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/archive/netcdfstorageforecasts?attribute(long_name)
=parameterA&requestedAttributes=long_name&startTime=2019-01-01T00:00:00Z&endTime=2020-01-01T00:00:
00Z&documentFormat=PI_JSON
```

Example response

```
{
  "externalNetCDFStorageForecasts" : [ {
    "forecastTime" : "2019-09-27T12:00:00+0000",
    "forecastAvailableTime" : "2020-06-23T13:45:53+0000",
    "attributes" : [ {
      "name" : "long_name",
      "name" : "sep"
    }, {
      "name" : "source",
```

```
      "value" : "Export NETCDF-CF_GRID_MATROOS from Delft-FEWS"
    }, {
      "name" : "source",
      "value" : "RWsOS-Noordzee"
    } ]
  }, {
    "forecastTime" : "2019-09-27T06:00:00+0000",
    "forecastAvailableTime" : "2020-06-23T13:45:53+0000",
    "attributes" : [ {
      "name" : "long_name",
      "value" : "sep"
    }, {
      "name" : "source",
      "value" : "Export NETCDF-CF_GRID_MATROOS from Delft-FEWS"
    }, {
      "name" : "source",
      "value" : "RWsOS-Noordzee"
    } ]
  } ]
}
```


GET archive/products/id (2022.02)

This endpoint can be used to return a specific product from the archive. The relative path in the archive will be used as the id.

Request parameters

- relativePath (string): the relative path of the product in the archive

```
http://localhost:63185/FewsWebServices/rest/fewspiservice/v1/archive/products/id?relativePath=products-rws/2022
/05/rivieren/10/product/weerbeeld_maas/2022_05_10_T_09_00_00/KNMI_20220510085242/weerbeeld_maas.txt
```

POST archive/products (2022.02)

This endpoint can be used to upload new products to the archive. The metaData.xml will be automaticly generated. It is only possible to upload a single product file each time this endpoint is used.

Request parameters

- timeZero (dateTime: yyyy-MM-ddTHH:mm:ssZ): time zero of the uploaded product
- areaId (string): area id of the uploaded product
- sourceId (string): source id of the uploaded product
- subFolder(string): the sub folder in which the product will be stored. This can be used as an folder to identify the product easier.

Form data

- file (file): a reference to the file which should be uploaded

```
http://localhost:63525/FewsWebServices/rest/fewspiservice/v1/archive/products/?
areaId=areaId&sourceId=source&timeZero=2020-01-01T00:00:00Z&subFolder=aaabbb
```

GET archive/products (2019.02)

Returns a zip-file with the requested products from the archive. Note that the results of this response are by default not cached.

Request parameters

- *areaId* (string): the area id of the requested products
- *sourceId* (string): the source id of the requested products
- *startTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): Start time of the archive search period
- *endTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): End time of the archive search period
- startForecastTime (dateTime: yyyy-MM-ddTHH:mm:ssZ): Start time of the forecast search period
- endForecastTime (dateTime: yyyy-MM-ddTHH:mm:ssZ): End Time of the forecast search period
- productFileName (string): if a productFileName is provided then only the products which match the provided file name will be returned,
- documentFormat (string): BINARY and BINARY_ZIP are supported. BINARY can only be used if the selection consists of a single product file.
- *productCount* (integer): the maximum number of products to be returned from the archive. If you only the download the most recent product in the requested period then use productCount=1

```
http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/archive/products?
areaId=myAreaId&sourceId=mySourceId&startTime=2011-10-10T12:00:00Z&endTime=2021-10-10T12:00:00Z&productCount=1
```

POST timeseries (2017.02)

Write timeseries data to the FEWS system using the timeseries sets defined by the filters. The timeSeries are stored in the piTimeSeriesXmlContent.

The timeseries you post to the rest service should match one of the time series sets in the default filter or one of its sub filters. To make sure you only write time series for a specific filter, you can pass a filterId with the POST request. Only timeseries that have timeseries sets that are configured in that filter (or one of its sub filters) will be accepted. If no filterId is used, all time series will be accepted that are configured in the default filter. Writing the time series works similar to importing time series using the pi.xml format using the "PI" import type. See also: Delft-Fews Published Interface timeseries Format (PI) Import

The 'convertDatum' argument is to allow timeseries that support a datum to have their values converted to a value relative to the location height. If values are already relative to location then enter FALSE or omit.

In case a time series already exists in the database, the time series will be overwritten by the ones that are posted. For forecast time series with different forecastDates a new time series will be added. The latter can be achieved by providing a forecastDate element in the POST request, e.g. *<forecastDate date="2013-01-01" time="00:00:00"/>.*

N.B.: by default POST operations are disabled in the Delft-FEWS WebServices and have to be explicitly enabled by setting the READONLY_MODE to false in the FewsPiService.properties. See also: FEWS Web Services Configuration FewsPiService.properties (deprecated since 2022.02).

Request parameters

- *filterId* (string, optional): Filter id for when the input timeseries maps to multiple internal timeseries. Within the scope of the filter the input timeseries should only map to one internal timeseries. When no filterId is specified, the default filter applies.
- *convertDatum* (boolean, optional): Convert values from relative location height to absolute height values

Body parameters

- *piTimeSeriesXmlContent (xml string, required):* PiTimeseries xml file encoded in the 'application/x-www-form-urlencoded Content-Type. Example of a *piTimeSeriesXmlContent (unencoded):*

```
<?xml version="1.0" encoding="UTF-8"?>
<TimeSeries xmlns="http://www.wldelft.nl/fews/PI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                        xsi:schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas
/version1.0/pi-schemas/pi_timeseries.xsd" version="1.12">
        <timeZone>0.0</timeZone>
        <series>
                <header>
                        <type>instantaneous</type>
                        <locationId>63306260000</locationId>
                        <parameterId>T.obs.mean</parameterId>
                        <timeStep unit="nonequidistant"/>
                        <startDate date="2013-01-01" time="00:40:00"/>
                        <endDate date="2013-01-01" time="00:50:00"/>
                        <missVal>-999.0</missVal>
                        <stationName>DE BILT</stationName>
                        <lat>51.76391247583424</lat>
                        <lon>4.329717456157946</lon>
                        <x>82000.0</x>
                        <y>420000.0</y>
                        <z>0.0</z>
```

```
                        <units>m</units>
                </header>
                <event date="2013-01-01" time="00:40:00" value="13.0" flag="0"/>
                <event date="2013-01-01" time="00:50:00" value="13.0" flag="0"/>
        </series>
</TimeSeries>
```

Since 2018.02 the service also supports milliseconds. The parser automatically recognises which format is used.

- ```
  <?xml version="1.0" encoding="UTF-8"?>
  <TimeSeries xmlns="http://www.wldelft.nl/fews/PI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                          xsi:schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas
  /version1.0/pi-schemas/pi_timeseries.xsd" version="1.12">
          <timeZone>0.0</timeZone>
          <series>
                  <header>
                          <type>instantaneous</type>
                          <locationId>63306260000</locationId>
                          <parameterId>T.obs.mean</parameterId>
                          <timeStep unit="nonequidistant"/>
                          <startDate date="2013-01-01" time="00:40:00.750"/>
                          <endDate date="2013-01-01" time="00:50:00.250"/>
                          <missVal>-999.0</missVal>
                          <stationName>DE BILT</stationName>
                          <lat>51.76391247583424</lat>
                          <lon>4.329717456157946</lon>
                          <x>82000.0</x>
                          <y>420000.0</y>
                          <z>0.0</z>
                          <units>m</units>
                  </header>
                  <event date="2013-01-01" time="00:40:00.750" value="13.0" flag="0"/>
                  <event date="2013-01-01" time="00:50:00.250" value="13.0" flag="0"/>
          </series>
  </TimeSeries>
  ```

- N.B. The SOAP Web Service has support for posting timeSeries as binary data using the *piTimeSeriesBinaryContent*. This is NOT support in this REST service of the PI Service.
- When sending in a set of timeseries that correspond to (e.g.) a completed forecast over multiple locations, it is not necessary to POST each location separately, the entire set be sent in one TimeSeries XML document with multiple series.
- There is no option in the REST Web Service to do any aggregation or transformation.

Response

- Diag PI xml response with all diagnostic output.

Example request

The following example shows how timeSeries can created using the POST method. Take note that the TimeSeries have to be posted in PI XML Format in the body of the POST request using the piTimeSeriesXmlContent key. The value is the content of the PI XML Timeseries in application/x-www-form-urlencoded format.

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/timeseries/" -X POST -H "Content-Type:
application/x-www-form-urlencoded" -d piTimeSeriesXmlContent=%3C%3Fxml%20version%3D%221.0%22%20encoding%3D%
22UTF-8%22%3F%3E%0D%0A%3CTimeSeries%20xmlns%3D%22http%3A%2F%2Fwww.wldelft.nl%2Ffews%2FPI%22%20xmlns%3Axsi%3D%
22http%3A%2F%2Fwww.w3.org%2F2001%2FXMLSchema-instance%22%0D%0A%09%09%09xsi%3AschemaLocation%3D%22http%3A%2F%
2Fwww.wldelft.nl%2Ffews%2FPI%20http%3A%2F%2Ffews.wldelft.nl%2Fschemas%2Fversion1.0%2Fpi-schemas%2Fpi_timeseries.
xsd%22%20version%3D%221.12%22%3E%0D%0A%09%3CtimeZone%3E0.0%3C%2FtimeZone%3E%0D%0A%09%3Cseries%3E%0D%0A%09%09%
3Cheader%3E%0D%0A%09%09%09%3Ctype%3Einstantaneous%3C%2Ftype%3E%0D%0A%09%09%09%3ClocationId%3E63306260000%3C%
2FlocationId%3E%0D%0A%09%09%09%3CparameterId%3ET.obs.mean%3C%2FparameterId%3E%0D%0A%09%09%09%3CtimeStep%20unit%
3D%22nonequidistant%22%2F%3E%0D%0A%09%09%09%3CstartDate%20date%3D%222013-01-01%22%20time%3D%2200%3A40%3A00%22%
2F%3E%0D%0A%09%09%09%3CendDate%20date%3D%222013-01-01%22%20time%3D%2200%3A50%3A00%22%2F%3E%0D%0A%09%09%09%
3CmissVal%3E-999.0%3C%2FmissVal%3E%0D%0A%09%09%09%3CstationName%3EDE%20BILT%3C%2FstationName%3E%0D%0A%09%09%09%
3Clat%3E51.76391247583424%3C%2Flat%3E%0D%0A%09%09%09%3Clon%3E4.329717456157946%3C%2Flon%3E%0D%0A%09%09%09%3Cx%
3E82000.0%3C%2Fx%3E%0D%0A%09%09%09%3Cy%3E420000.0%3C%2Fy%3E%0D%0A%09%09%09%3Cz%3E0.0%3C%2Fz%3E%0D%0A%09%09%09%
3Cunits%3Em%3C%2Funits%3E%0D%0A%09%09%3C%2Fheader%3E%0D%0A%09%09%3Cevent%20date%3D%222013-01-01%22%20time%3D%
2200%3A40%3A00%22%20value%3D%2213.0%22%20flag%3D%220%22%2F%3E%0D%0A%09%09%3Cevent%20date%3D%222013-01-01%22%
20time%3D%2200%3A50%3A00%22%20value%3D%2213.0%22%20flag%3D%220%22%2F%3E%0D%0A%09%3C%2Fseries%3E%0D%0A%3C%
2FTimeSeries%3E
```
```

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<Diag xmlns="http://www.wldelft.nl/fews/PI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.
wldelft.nl/schemas/version1.0/pi-schemas/pi_diag.xsd" version="1.2">
    <line level="2" description="Multiple time series sets found for parameter=T.obs.mean location=63306260000"
/>
    <line level="3" description="Import.Info: External time series successfully mapped to FEWS time series
63306260000 T.obs.mean   (m) nonequidistant Tue Jan 0
1 01:40:00 CET 2013 z=0.0"/>
    <line level="3" description="Import.info: 1 time series imported, 0 time series rejected"/>
    <line level="3" description="Import.info: The following locations-parameter combination imported
63306260000:T.obs.mean"/
</Diag>
```

Example response with non-matching time series.

It is possible that the posted time series do not match any of the time series sets that have been configured in the default filter or in a passed filterId. In that case, the response will report that some time series couldn't be mapped. In the following eexample 8 time series were posted, but only one time series was actually writte, 7 time series were rejected:

```
<?xml version="1.0" encoding="UTF-8"?>
<Diag xmlns="http://www.wldelft.nl/fews/PI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas/pi_diag.xsd"
version="1.2">
    <line level="3" description="Import.Info: External time series successfully mapped to FEWS time series
amerongen_beneden H.voorspeld   (m) 10 minutes Tue Jan 01 01:00:00 CET 2013 z=0.0"/>
    <line level="3" description="Import.Info: 1 time series imported, 7 time series rejected"/>
    <line level="3" description="Import.Info: The following locations-parameter combination imported
amerongen_beneden:H.voorspeld"/>
</Diag>
```

Example forecast time series.

By specifying a forecastDate it is possible to write external forecasts using the PI service.

```
<?xml version="1.0" encoding="UTF-8"?>
<TimeSeries xmlns="http://www.wldelft.nl/fews/PI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas
/pi_timeseries.xsd" version="1.12">
    <timeZone>0.0</timeZone>
    <series>
        <header>
            <type>instantaneous</type>
            <locationId>amerongen_beneden</locationId>
            <parameterId>H.voorspeld</parameterId>
            <timeStep unit="minute" multiplier="10"/>
            <startDate date="2013-01-01" time="00:00:00"/>
            <endDate date="2013-01-01" time="00:30:00"/>
            <forecastDate date="2013-01-01" time="00:00:00"/>
            <missVal>-999.0</missVal>
            <units>m</units>
        </header>
        <event date="2013-01-01" time="00:00:00" value="10.0" flag="0" />
        <event date="2013-01-01" time="00:10:00" value="12.0" flag="0"/>
        <event date="2013-01-01" time="00:20:00" value="11.8" flag="0"/>
        <event date="2013-01-01" time="00:30:00" value="9.0" flag="0"/>
    </series>
</TimeSeries>
```

Example code

To post timeSeries to the Delft-FEWS WebServices, the following is example JAVA code:

```
String piTimeSeries = "PI XML Content";
URL url = new URL("http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/timeseries");
HttpURLConnection connection = (HttpURLConnection) url.openConnection();
connection.setDoInput(true);
connection.setRequestMethod("POST");
connection.setDoOutput(true);
connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
connection.setRequestProperty("charset", "utf-8");
String postData = "piTimeSeriesXmlContent=" + URLEncoder.encode(piTimeSeries, StandardCharsets.UTF_8);
connection.setRequestProperty("Content-Length", Integer.toString(postData.length()));
connection.setUseCaches(false);
try (DataOutputStream wr = new DataOutputStream(connection.getOutputStream())) {
    wr.write(postData.getBytes());

}
```

GET taskruns

Get the taskRuns that comply to the filters. By default only forecasts are returned. Note that the results of this response are by default not cached.

Request parameters

- *onlyForecasts* (boolean): option to toggle if only forecasts should be returned. Default true.
- *onlyCurrent* (boolean): option to toggle if only current forecasts should be returned. Default false.
- *startDispatchTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): Start time of search period that looks for taskruns that have their dispatch time within this period.
- *endDispatchTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): End time of search period that looks for taskruns that have their dispatch time within this period.
- *startForecastTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): Start time of search period that looks for taskruns that have their forecast time within this period.
- *endForecastTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): End time of search period that looks for taskruns that have their forecast time within this period.
- *workflowId* (string): Filter by an existing workflow id.
- *scenarioId* (string): Filter by an existing whatsif scenario id.
- mcId (string): Filter by mcId. Since 2021.01.
- *taskRunStatusIds* (string): Filter taskruns using the taskrunstatus code value. For valid status codes see the section 'Get taskrunstatus' below.
- *documentVersion* (string, 1.9 or up): File format version (optional). For example: 1.23

Response

- TaskRuns PI XML file content.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/taskruns?
workflowId=ImportObserved&onlyForecasts=false"
```

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<TaskRuns xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas/pi_taskruns.
xsd" version="1.23">
    <timeZone>0.0</timeZone>
    <taskRun taskRunId="0_0">
        <forecast>false</forecast>
        <status>completed fully successful</status>
        <workflowId>ImportObserved</workflowId>
        <dispatchTime date="2017-08-13" time="09:55:18"/>
        <completionTime date="2017-08-13" time="10:21:35"/>
        <time0 date="2017-08-13" time="09:45:00"/>
        <outputStartTime date="1743-11-01" time="00:00:00"/>
        <outputEndTime date="2017-08-01" time="00:00:00"/>
```

```
        <user>rudie</user>
    </taskRun>
</TaskRuns>
```

GET moduleruntimes (2021.02)

Get all expected runtimes for workflows per module instance id. The list can optionally be filtered by workflowId. Only workflows of scheduled tasks that contain module instance descriptors that have been configured with updateModuleRunTimesOnCompletion enabled, will be available in this end point.

The expected start time of a module is calculated based on the scheduled next due time and the expected pending duration time. The expected end time of a module is calculated based on the scheduled next due time, the expected pending duration time and the expected running time. For triggered tasks, the expected start time en end time won't be available until the task is actually started.

Request parameters

- *workflowId* (string): optional, allow filtering for a specific worklowId.
- *documentFormat* (string): PI_XML (default)

Response

- ModuleRunTime PI XML or PI JSON content.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/moduleruntimes?
workflowId=Meuse_DWD_COSMO_LEPS"
```

Example response PI_XML

```
<?xml version="1.0" encoding="UTF-8"?>
<ModuleRunTimes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas
/pi_moduleruntimes.xsd">
    <moduleRunTime>
        <mcId>nldefedmc00</mcId>
        <workflowId>Meuse_DWD_COSMO_LEPS</workflowId>
        <moduleInstanceId>Dummy</moduleInstanceId>
        <expectedStartTime date="2021-10-13" time="13:15:01"/>
        <expectedCompletionTime date="2021-10-13" time="13:16:54"/>
        <expectedPendingDuration>1465</expectedPendingDuration>
        <expectedRunningDuration>112634</expectedRunningDuration>
    </moduleRunTime>
</ModuleRunTimes>
```

Example response PI_JSON

```
{
  "moduleRunTimes" : [ {
    "workflowId" : "Meuse_DWD_COSMO_LEPS",
    "moduleInstanceId" : "Dummy",
    "mcId" : "nldefedmc00",
    "expectedStartTime" : 1414882800000,
    "expectedCompletionTime" : 1414886400000,
    "expectedPendingDuration" : 1465,
    "expectedRunningDuration" : 112634
  } ]
}
```

GET taskrunstatus (2017.02)

Track the status of a workflow using the tasId which is provided when running a workflow with the runTask endpoint.

Request parameters

- *taskId* (string, required): task Id of a workflow.
- *maxWaitMillis* (integer) time in milliseconds to wait for response
- *documentVersion* (string, 1.9 or up): Since 2022.02. File format version (optional). For example: 1.29
- *documentFormat* (string): PI_JSON. Since 2022.02. If not specified, plain text is assumed.

Response

Status of the workflow task. Possible response codes are:

- `I = Invalid,`
- `P = Pending,`
- `T = Terminated,`
- `R = running,`
- `F = Failed,`
- `C = Completed fully successful,`
- `D = Completed partly successful,`
- `A = Approved,`
- `B = Approved partly successful`
- `null= No status available (produces when method call times-out)`

Example request

```
curl "localhost:8080/FewsWebServices/rest/fewspiservice/v1/taskrunstatus?taskId=1_0"
```

Example response

```
C
```

Since 2022.02 the taskrun status response supports PI_JSON as well. Since 2024.01 the taskRunId is available as well. For example:

```
{
  "version" : "1.29",
  "code" : "C",
  "description" : "Completed fully successful",
  "taskRunId" : "1234"
}
```

POST runtask (2017.02)

Runs a workflow task for a given workflowId. Returns a handle to the task in the form of a taskid. This taskId can be used to track the status of the workflow using the taskrunstatus method. Since 2018.02 it is possible to use a workflow descriptor attribute: waitWhenAlreadyRunning. This will allow running a task that hasn't been scheduled to wait when another task of that workflow is already running.

Request parameters

- *workflowId* (string, required): Identifier of the task to run
- startTime (dateTime: yyyy-MM-ddTHH:mm:ssZ): Start of run period. Used for state selection period.
- *timeZero* (dateTime: yyyy-MM-ddTHH:mm:ssZ): Forecast time zero. If missing System time is used (optional)
- *endTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): End of run period. Used to define forecast length.
- *coldStateId* (string): Id of a coldstate. Can be used to force state selection (optional).
- *scenarioId* (string): Id of a predefined WhatIf scenario. Can be used to alter run parameters (optional).
- *userId* (string) User id of the user that runs the task.
- *description* (string): Descriptive text to identify run.
- runOption (string, optional, since 2022.02): Run option can be any of: all, alloneatatime or allmostrecentonly. If not set, the default is used: all.
    1. all: Multiple instances of this workflow can run simultaneously.
    2. alloneatatime: Running (and queued) instances of this workflow prevail.
    3. allmostrecentonly: A running instance of this workflow prevails. Queued instances of this workflow will be replaced by a recent one.
- optional: Since 2022.02 properties can be included in the url. These will be used as taskRunProperties, and override global or workflow properties. Each property has to be aded to the url seperately.
  Example:
  &property(fileName)=exportFile&property(outputValue)=9.0
  Where the name of the property is fileName, the value is exportFile. The name of the second property is outputValue, the value is 9.0.
- optional since 2023.02: (boolean) runLocallyAndPromoteToServer. Default is false. If it's set to true, the task will be run locally first, and if the run is succesful, it will be promoted to the server.

Body parameters

- *piModelParametersXmlContent* (pi XML url encoded): Contents of a Pi ModelParameters XML file. PI ModelParameters can be exported by the General Adapter to provide information to external models being run by FEWS. The xml file content has to be encoded in the 'application/x-www-form-urlencoded Content-Type.

Response

- taskId String with the identifier of the task that is run.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/runtask/?
workflowId=ImportObserved&startTime=2014-01-01T00:00:00Z+0000&timeZero=2014-01-01T00:00:00Z+0000&endTime=2014-
01-01T00:30:00Z+0000&property(fileName)=exportFile&property(outputValue)=9.0" -X POST -H "Content-Type:
application/x-www-form-urlencoded"  -d ""
```

Example response

```
1_0
```

GET timeseriesmodifiers (2017.02)

Get a list of all timeSeries modifiers

Request parameters

- *locationIds* (string):
- *moduleInstanceIds* (string):
- *startTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): start time of modifiers search period
- *endTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): end time of modifiers search period.
- *userId* (string):
- *modifierTypeId* (string):
- *active* (boolean, default true):
- *userDefinedModifierDescriptionKeyValuePair* (string):

If no startTime and endTime are given, the search period is any time.

Response

- timeSeriesModifiers PI XML

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/timeseriesmodifiers"
```

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<Modifiers xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas/pi_modifiers.
xsd">
    <modifier>
        <name>T.obs.mean_DE BILT</name>
        <modifierId>0</modifierId>
        <systemActivityDescriptorId>34</systemActivityDescriptorId>
        <enabled>true</enabled>
        <modifierType>MISSING_VALUE_MODIFIER</modifierType>
        <userId>Andre Grijze</userId>
        <startTime date="1900-10-17" time="00:00:00"/>
        <endTime date="2017-11-23" time="00:00:00"/>
        <validTime date="3000-01-01" time="00:00:00"/>
        <creationTime date="2017-10-24" time="11:43:11"/>
        <constantValueTimeSeriesModifier>
            <timeSeriesSet>
                <moduleInstanceId>ImportObserved</moduleInstanceId>
                <parameterId>T.obs.mean</parameterId>
                <locationId>63306260000</locationId>
                <timeSeriesType>external historical</timeSeriesType>
                <timeStep unit="nonequidistant"/>
```

```
                <ensembleId>main</ensembleId>
            </timeSeriesSet>
            <startTime date="1900-10-17" time="00:00:00"/>
            <endTime date="2017-11-23" time="00:00:00"/>
            <value>NaN</value>
        </constantValueTimeSeriesModifier>
    </modifier>
</Modifiers>
```

GET modifiers

Get a list of all modifiers.

Request parameters

- *startTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): start time of modifiers search period
- *endTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): end time of modifiers search period.
- *modifierTypeId* (string): filter on modifiers by modifierTypeId as specified in the ModifierTypes.xml configuration. If modifier type cannot be found, not filtering on type is done.

If no startTime and endTime are given, the search period is any time.

Response

- Modifiers PI XML

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/modifiers?
modifierTypeId=MISSING_VALUE_MODIFIER"
```

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<Modifiers xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas/pi_modifiers.
xsd">
    <modifier>
        <name>T.obs.mean_DE BILT</name>
        <modifierId>0</modifierId>
        <systemActivityDescriptorId>34</systemActivityDescriptorId>
        <enabled>true</enabled>
        <modifierType>MISSING_VALUE_MODIFIER</modifierType>
        <userId>Rudie Ekkelenkamp</userId>
        <startTime date="1900-10-17" time="00:00:00"/>
        <endTime date="2017-11-23" time="00:00:00"/>
        <validTime date="3000-01-01" time="00:00:00"/>
        <creationTime date="2017-10-24" time="11:43:11"/>
        <constantValueTimeSeriesModifier>
            <timeSeriesSet>
                <moduleInstanceId>ImportObserved</moduleInstanceId>
                <parameterId>T.obs.mean</parameterId>
                <locationId>63306260000</locationId>
                <timeSeriesType>external historical</timeSeriesType>
                <timeStep unit="nonequidistant"/>
                <ensembleId>main</ensembleId>
            </timeSeriesSet>
            <startTime date="1900-10-17" time="00:00:00"/>
            <endTime date="2017-11-23" time="00:00:00"/>
            <value>NaN</value>
        </constantValueTimeSeriesModifier>
    </modifier>
</Modifiers>
```

POST modifiers (2017.02)

Create new modifiers.

Request parameters

- *commitModiifers* (boolean): Indicates if the modifiers should be committed after the uploaded. This option defaults to true.
- *deleteAllModifiers* (boolean): Indicates if all the existing modifiers should be deleted prior to inserting the new modifiers. This option defaults to false.

Body parameters

- *piModifiersXmlContent* (pi Modifiers XML url encoded): Contents of a modifiers file. The xml file content has to be encoded in the 'application/x-www-form-urlencoded Content-Type.

Response

- Diag PI xml response with all diagnostic output.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/modifiers" -X POST -H "Content-Type:
application/x-www-form-urlencoded"  -d "piModifiersXmlContent="%3C%3Fxml%20version%3D%221.0%22%20encoding%3D%
22UTF-8%22%3F%3E%0D%0A%3CModifiers%20xmlns%3Axsi%3D%22http%3A%2F%2Fwww.w3.org%2F2001%2FXMLSchema-instance%22%
20xmlns%3D%22http%3A%2F%2Fwww.wldelft.nl%2Ffews%2FPI%22%20xsi%3AschemaLocation%3D%22http%3A%2F%2Fwww.wldelft.nl%
2Ffews%2FPI%20http%3A%2F%2Ffews.wldelft.nl%2Fschemas%2Fversion1.0%2Fpi-schemas%2Fpi_modifiers.xsd%22%3E%0D%0A%
20%20%20%20%3Cmodifier%3E%0D%0A%20%20%20%20%20%20%20%20%3Cname%3ET.obs.mean_DE%20BILT%3C%2Fname%3E%0D%0A%20%20%
20%20%20%20%20%20%3Cenabled%3Etrue%3C%2Fenabled%3E%0D%0A%20%20%20%20%20%20%20%20%3CmodifierType%
3EMODIFY_TIMESERIES%3C%2FmodifierType%3E%0D%0A%20%20%20%20%20%20%20%20%3CuserId%3ERudie%20Ekkelenkamp%3C%
2FuserId%3E%0D%0A%20%20%20%20%20%20%20%20%3CstartTime%20date%3D%221900-10-24%22%20time%3D%2200%3A00%3A00%22%2F%
3E%0D%0A%20%20%20%20%20%20%20%20%3CendTime%20date%3D%222017-10-25%22%20time%3D%2200%3A00%3A00%22%2F%3E%0D%0A%20%
20%20%20%20%20%20%20%3CvalidTime%20date%3D%222017-10-24%22%20time%3D%2200%3A00%3A00%22%2F%3E%0D%0A%20%20%20%20%
20%20%20%20%3CcreationTime%20date%3D%222017-10-24%22%20time%3D%2211%3A15%3A43%22%2F%3E%0D%0A%20%20%20%20%20%20%
20%20%3CtransformationTimeSeriesModifier%3E%0D%0A%20%20%20%20%20%20%20%20%20%20%20%20%3CtimeSeriesSet%3E%0D%0A%
20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3CmoduleInstanceId%3EImportObserved%3C%2FmoduleInstanceId%3E%0D%
0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3CparameterId%3ET.obs.mean%3C%2FparameterId%3E%0D%0A%20%20%
20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3ClocationId%3E63306260000%3C%2FlocationId%3E%0D%0A%20%20%20%20%20%
20%20%20%20%20%20%20%20%20%20%20%20%3CtimeSeriesType%3Eexternal%20historical%3C%2FtimeSeriesType%3E%0D%0A%20%20%20%20%
20%20%20%20%20%20%20%20%20%20%20%20%3CtimeStep%20unit%3D%22nonequidistant%22%2F%3E%0D%0A%20%20%20%20%20%20%20%
20%20%20%20%20%20%20%20%20%20%3CensembleId%3Emain%3C%2FensembleId%3E%0D%0A%20%20%20%20%20%20%20%20%20%20%20%3C%
2FtimeSeriesSet%3E%0D%0A%20%20%20%20%20%20%20%20%20%20%20%3Cmultiplier%3E1.0%3C%2Fmultiplier%3E%0D%0A%20%20%
20%20%20%20%20%20%20%20%20%20%3Cdivider%3E1.1%3C%2Fdivider%3E%0D%0A%20%20%20%20%20%20%20%20%20%20%20%20%
3Cincrementer%3E0.0%3C%2Fincrementer%3E%0D%0A%20%20%20%20%20%20%20%20%20%20%20%20%3Cdelay%3E0%3C%2Fdelay%3E%0D%
0A%20%20%20%20%20%20%20%20%3C%2FtransformationTimeSeriesModifier%3E%0D%0A%20%20%20%20%3C%2Fmodifier%3E%0D%0A%3C%
2FModifiers%3E"
```

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<Diag xmlns="http://www.wldelft.nl/fews/PI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.
wldelft.nl/schemas/version1.0/pi-schemas/pi_diag.xsd" version="1.2">
    <line level="3" description="Successfully imported modifier:none MODIFY_TIMESERIES -1 T.obs.mean_DE BILT
none"/>
</Diag>
```

GET workflows (2017.02)

Request parameters

- *documentVersion* (string, 1.9 or up): File format version (optional). For example: 1.23

Response

- xml workflows.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/workflows"
```

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<workflows xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://
fews.wldelft.nl/schemas/version1.0/pi-schemas/pi_workflows.xsd" version="1.23">
    <workflow id="ImportObserved">
        <name>Import Observed Data</name>
        <description>Import observed scalar data from external sources and process the imported data<
/description>
    </workflow>
</workflows>
```

GET samples (2017.02)

Request parameters

- *documentVersion* (string, 1.21 or up): File format version (optional). For example: 1.23
- *endCreationTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): End time of search period that looks for creation time of samples. Note: creation time of samples is actually the creation time of the task that produced/imported these samples.
- *endTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): End time of search period that looks for sample values that lie within this period.
- *filterId* (string): Filter id.
- *locationIds* (string): Subset of locations for which to retrieve samples. Repeat parameter for multiple location ids.
- *omitMissing* (boolean): Toggle omitting or returning of missing values in response
- *onlyHeaders* (boolean): Toggle to return only header information or also data
- *filterTimeSeriesWithinSample* (boolean): Since 2023.02 Toggle to filter time series data within sample. For example on parameter id's and / or qualifier id's
- *parameterIds* (string): Subset of parameters for which to retrieve samples.
- *qualifierIds* (string): Subset of qualifiers for which to retrieve samples.
- *sampleIds* (string): Subset of sample id's for which to retrieve samples. Repeat parameter for multiple samples ids.
- *moduleInstanceIds* (string): Subset of module instance id's for which to retrieve samples.
- *startCreationTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): Start time of search period that looks for creation time of samples. Note: creation time of samples is actually the creation time of the task that produced/imported these samples.
- *startTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): Start time of search period that looks for sample values that lie within this period.

Since all parameters are optional, some defaults are chosen for the search period in case no search period or creation period was specified. The following rules apply:

- If no startCreationTime and endCreationTime have been set, the startTime and endTime are used to determine the search period.
- If no startTime and endTime have been specified, the search period will be set to the current system time minus one day and one hour until the current system time plus one day and one hour.

Response

- Samples PI XML file content

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/samples?
sampleIds=713&locationIds=MBP012&startTime=1970-01-01T00:00:00Z+0000&endTime=2017-01-01T00:00:00Z+0000"
```

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<Samples xmlns="http://www.wldelft.nl/fews/PI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas/pi_samples.
xsd" version="1.23" xmlns:fs="http://www.wldelft.nl/fews/fs" xmlns:qualifierId="http://www.wldelft.nl/fews
/qualifierId">
    <timeZone>0.0</timeZone>
    <sample id="713">
        <header>
            <locationId>MBP012</locationId>
            <sampleDate date="1980-12-31" time="23:00:00"/>
            <lat>52.18084820135932</lat>
            <lon>5.083729510982581</lon>
            <x>134244.0</x>
            <y>465900.0</y>
```

```
        </header>
        <properties>
            <string key="externereferentie" value="PUT-01\SYS\0000000713"/>
            <string key="xcoormonster" value="134240"/>
            <string key="ycoormonster" value="465900"/>
            <string key="monsternemer" value="IMP"/>
            <string key="analist" value="IMP"/>
            <string key="bron" value="TABHIST"/>
        </properties>
        <table>
            <row parameterId="VSA_n" qualifierId:q1="AC_18" qualifierId:q2="CO_19" qualifierId:q3="QU_1"
qualifierId:q4="BN_13934" qualifierId:q5="LC_1" qualifierId:q6="GE_1" flag="0" value="346.97" unit="n"/>
            <row parameterId="VSA_n" qualifierId:q1="AC_18" qualifierId:q2="CO_19" qualifierId:q3="QU_1"
qualifierId:q4="BN_13934" qualifierId:q5="LC_2" qualifierId:q6="GE_1" flag="0" value="34.72" unit="n"/>
        </table>
    </sample>
</Samples>
```

Samples PI-XML example with detection limit symbols (since 2020.02)

In case limit a detection limit is available in for an event, the detection attribute will be given as part of the row element. Possible values are: <>~

```
<table>
    <row parameterId="Ntot" qualifierId:q1="CO_19" qualifierId:q2="QU_27" qualifierId:q3="UE_27" flag="0"
value="3.58" detection="&gt;" unit="mg/l"/>
    <row parameterId="Ptot" qualifierId:q1="CO_19" qualifierId:q2="QU_27" qualifierId:q3="UE_34" flag="0"
value="0.11" detection="&lt;" unit="mg/l"/>
</table>
```

Samples PI-XML example with value properties (since 2023.02 pi version 1.34)

When there are value properties for individual values they are shown before the row values

Samples PI-XML example with value properties

```
<?xml version="1.0" encoding="UTF-8"?>
<Samples xmlns="http://www.wldelft.nl/fews/PI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas/pi_samples.
xsd" version="1.34" xmlns:fs="http://www.wldelft.nl/fews/fs" xmlns:qualifierId="http://www.wldelft.nl/fews
/qualifierId">
    <timeZone>0.0</timeZone>
    <sample id="713">
        <header>
            <locationId>MBP012</locationId>
            <sampleDate date="1980-12-31" time="23:00:00"/>
            <lat>52.18084820135932</lat>
            <lon>5.083729510982581</lon>
            <x>134244.0</x>
            <y>465900.0</y>
        </header>
        <properties>
            <string key="externereferentie" value="PUT-01\SYS\0000000713"/>
            <string key="xcoormonster" value="134240"/>
            <string key="ycoormonster" value="465900"/>
            <string key="monsternemer" value="IMP"/>
            <string key="analist" value="IMP"/>
            <string key="bron" value="TABHIST"/>
        </properties>
        <table>
            <properties>
                <string key="valuePropertyA" value="XXX"/>
                <string key="valuePropertyB" value="XYZ"/>
            </properties>
            <row parameterId="VSA_n" qualifierId:q1="AC_18" qualifierId:q2="CO_19" qualifierId:q3="QU_1"
qualifierId:q4="BN_13934" qualifierId:q5="LC_1" qualifierId:q6="GE_1" flag="0" value="346.97" unit="n"/>
            <properties>
                <string key="valuePropertyA" value="YYY"/>
```

```
                <string key="valuePropertyB" value="ABC"/>
            </properties>
            <row parameterId="VSA_n" qualifierId:q1="AC_18" qualifierId:q2="CO_19" qualifierId:q3="QU_1"
qualifierId:q4="BN_13934" qualifierId:q5="LC_2" qualifierId:q6="GE_1" flag="0" value="34.72" unit="n"/>
        </table>
    </sample>
</Samples>
```

Streaming sample response (since 2023.02)

In order to handle larger requests the get samples processes its response in smaller steps. This means that it will first read 100,000 time series headers and all samples they are part of, then those full samples are read and written to the xml response. Then the next 100,000 headers are processed, this continues until all headers are processed. If there are less than 100,000 headers all samples will be processed at the same time. All samples that are fully written to the response are kept track of so that subsequent sets of 100,000 headers will not include the previously written samples again.

Because the time series headers are processed step by step it is not known what the headers and samples for the next step are going to be. Therefore the samples will not appear in a predictable order.

GET processdata (2017.02)

Request parameters

- *startTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): start time of the requested period
- *endTime* (dateTime: yyyy-MM-ddTHH:mm:ssZ): end time of the requested period
- *workflowId* (string): workflow which should be run. This workflow should generate the requested data file
- *xMin* (string): left x coordinate of the bounding box for which data should be generated
- *xMax* (string): right x coordinate of the bounding box for which data should be generated
- *yMin* (string): lower y coordinate of the bounding box for which data should be generated
- *yMax* (string): upper y coordinate of the bounding box for which data should be generated
- *xCellSize* (string): width of the cell size
- *yCellSize* (string): height of the cell size
- optional since 2023.02: (boolean) runLocallyAndPromoteToServer. Default is false. If it's set to true, the task will be run locally first, and if the run is succesful, it will be promoted to the server.

Response

- The workflow indicated by the workflowId should generate a data-file (usually a netcdf-file). if the workfow has succesfully generated a netcdf-file, the response will be binary stream containing the generated file. If no file is generated an 500 error will be thrown.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/processdata/?
workflowId=Transformation_ASA_Grid_Wind&xMin=52.58&yMin=24.38&xMax=54.62&yMax=26.10&xCellSize=0.01&yCellSize=0.
01&startTime=2017-11-18T00:00:00Z&endTime=2017-11-19T00:00:00Z"
```

Configuration

A property **EXPORT_FOLDER_PROCESS_DATA** should be configured in the global properties or in the webservice properties. The workflow indicated by the workflow id should export data to a folder **EXPORT_FOLDER_PROCESS_DATA**.

GET qualifiers (2019.02)

Retrieve all configured qualifiers.

Request parameters

- *documentFormat* (string): only PI_XML is supported.
- *showAttributes* (boolean): toggle to show qualifier attributes. Default is false.

Response

- Qualifiers in PI-XML.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/qualifiers?
showAttributes=true&documentVersion=1.24"
```

Example PI-XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<qualifiers xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas
/pi_qualifiers.xsd" version="1.24">
    <qualifier id="1">
        <name>Alisma plantago-aquatica</name>
        <attribute id="ExternalQualifierId" name="ExternalQualifierId">
            <text>PAR_REF_Alisma plantago-aquatica</text>
        </attribute>
        <attribute id="Qualifier_TAXA" name="Qualifier_TAXA">
            <description>Taxon naam</description>
            <text>Alisma plantago-aquatica</text>
        </attribute>
        <attribute id="Type" name="Type">
            <description>type</description>
            <text>aquaticaType</text>
        </attribute>
        <attribute id="Groep" name="Groep">
            <description>Familie</description>
            <text>aquaticaGroup</text>
        </attribute>
    </qualifier>
    <qualifier id="2">
        <name>Butomus umbellatus</name>
        <attribute id="ExternalQualifierId" name="ExternalQualifierId">
            <text>PAR_REF_Butomus umbellatus</text>
        </attribute>
        <attribute id="Qualifier_TAXA" name="Qualifier_TAXA">
            <description>Taxon naam</description>
            <text>Butomus umbellatus</text>
        </attribute>
        <attribute id="Type" name="Type">
            <description>type</description>
            <text>umbellatusype</text>
        </attribute>
        <attribute id="Groep" name="Groep">
            <description>Familie</description>
            <text>umbellatusGroup</text>
        </attribute>
    </qualifier>
</qualifiers>
```

GET moduleruntables/current (2020.02)

Retrieve csv file from the current module run table, based on moduleInstanceId or taskRunId. Either moduleInstanceId or taskRunId has to be specified.

Request parameters

- *moduleInstanceId* (string): an existing module instance id.
- *taskRunId* (string): an existing taskRunId.

Response

- Content of the csv file of the module run table. Dates are in GMT-0 in the format: yyyyMMddHHmmss
- The csv file name is returned in the Content-Disposition header: attachment;filename=filename.csv

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/moduleruntables/current?
moduleInstanceId=myModuleInstance"
```

Example CSV response: content type: text/csv

```
datum,status,Yes or no,int,double,string
20030222000000,Ok,F,14,1.45,Overview 1 description 2
20030223000000,Not Ok,T,24,2.45,Overview 1 description 2
```

GET version (2021.02)

GET version information of the current installed Web Services.

Request parameters

- *documentFormat* (string): PI_XML (default) or PI_JSON

Response

- PI-XML or PI-JSON file content.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/version"
```

Example PI-XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<Version xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas/pi_version.
xsd">
    <implementation>2017.02</implementation>
    <buildType>stable</buildType>
    <buildNumber>12345</buildNumber>
    <buildTime>2017-10-31T23:00:00Z</buildTime>
</Version>
```

GET oas (2022.01)

GET open api specification of the REST API.

Response

- JSON response with the open api specification.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/oas"
```

Example PI-XML response

```
{
  "openapi" : "3.0.3",
  "info" : {
    "title" : "Delft-FEWS Web Services - Schematic Status Display (SSD) Web Service",
    "description" : "Delft-FEWS Web Services - Schematic Status Display (SSD) Web Service. For more information
see: https://publicwiki.deltares.nl/x/OwXkC",
    "version" : ""
  },
  "servers" : [ {
    "url" : "/FewsWebServices",
    "description" : "API server"
  } ]
}
```

GET ratingcurves (2022.02)

GET all configured rating curves from the region config and optionally filter by locationIds.

Request parameters

- *documentVersion* (string): PI_XML or PI_JSON is supported.
- *locationIds* (string, optional): one or more locationIds for which to retrieve the rating curves.
- *onlyHeaders* (boolean): Toggle to return only header information or also data

Response

- PI_XML response with rating curves.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/ratingcurves?locationIds=ALKA2"
```

Example PI-XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<RatingCurves xmlns="http://www.wldelft.nl/fews/PI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-schemas
/pi_ratingcurves.xsd">
    <timeZone>0.0</timeZone>
    <ratingCurve>
        <header>
            <locationId>ALKA2</locationId>
            <startDate date="2012-01-01" time="00:00:00"/>
            <endDate date="2012-11-21" time="00:00:00"/>
            <stationName>ALKA2</stationName>
            <stageUnit>FT</stageUnit>
            <dischargeUnit>CFS</dischargeUnit>
            <comment>Sep07 88.88 199 kcfs</comment>
        </header>
        <table>
            <interpolationMethod>logarithmic</interpolationMethod>
            <minStage>64.9</minStage>
            <maxStage>INF</maxStage>
            <row stage="64.9" discharge="2990.3328"/>
            <row stage="65.1" discharge="3280.365"/>
            <row stage="65.4" discharge="3740.4165"/>
            <row stage="65.700004" discharge="4240.472"/>
        </table>
    </ratingCurve>
</RatingCurves>
```

POST ratingcurves/stagetodischarge(2022.02)

POST Converts stage values to discharge values using a rating curve at a specific location. The stageValues should be passed in the body of the POST request as a key value pair where the key is dischargeValue and the value is URL Encoded JSON that conforms to the pi_rest_ratingcurves_stage.json schema: https://fewsdocs.deltares.nl/webservices/rest-api/v1/schemas/pirest/pi_rest_ratingcurves_stage.json

Request parameters

- *documentVersion* (string): only PI_XML is supported.
- *locationId* (string, required): Location id for which to calculate the discharge.

Body parameters

- *stageValues (json string, required): Stage values in JSON format that conforms to the schema: https://fewsdocs.deltares.nl/webservices/rest-api/v1/schemas/pirest/pi_rest_ratingcurves_stage.json. Example: { "stageValues" : ["22.1", "22.2"] }:*

```
{ "stageValues" : [ "22.1", "22.2" ] }
```

Response

- PI_JSON response with rating curves.

Example request

```
curl http://localhost:8100/FewsWebServices/rest/fewspiservice/v1/ratingcurves/stagetodischarge?locationId=CMKT2
-d 'stageValues={ "stageValues" : [ "22.1", "22.2" ] }'
```

Example PI-XML response

```
{ "dischargeValues" : [ "9920.0", "10200.0" ] }
```

POST ratingcurves/dischargetostage (2022.02)

Converts stage values to discharge values using a rating curve at a specific location. The stageValues should be passed in the body of the POST request as a key value pair where the key is dischargeValue and the value is URL Encoded JSON that conforms to the pi_rest_ratingcurves_stage.json schema: https://fewsdocs.deltares.nl/webservices/rest-api/v1/schemas/pirest/pi_rest_ratingcurves_stage.json.

Request parameters

- *documentVersion* (string): only PI_XML is supported.
- *locationId* (string, required): Location id for which to calculate the discharge.

Body parameters

- *stageValues (json string, required): Stage values in JSON format that conforms to the schema: https://fewsdocs.deltares.nl/webservices/rest-api/v1/schemas/pirest/pi_rest_ratingcurves_stage.json. Example: { "stageValues" : ["22.1", "22.2"] }:*

  ```
  { "dischargeValues" : [ "9920.0", "10200.0" ] }
  ```

Response

- PI_JSON response with rating curves.

Example request

```
curl http://localhost:8100/FewsWebServices/rest/fewspiservice/v1/ratingcurves/dischargetostage?locationId=CMKT2
-d 'dischargeValues={ "dischargeValues" : [ "66062.04", "66546.8" ] }'
```

Example PI-XML response

```
{ "stageValues" : [ "22.1", "22.2" ] }
```

GET warmstates/times (2022.02)

Get all the available warm state times for the specified module instance ids. The id is used to retrieve the current taskrun for the module instance..

- moduleInstanceIds(string): one or more module instance ids.

Response

- PI_JSON response with state times per module instance ids

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/warmstates/times?
moduleInstanceIds=moduleInstanceId1"
```

Example PI-XML response

```
{
  "warmStateTimes" : [ {
    "moduleInstanceId" : "moduleInstanceId1",
    "times" : [ "1970-01-01T00:57:04Z", "1970-01-01T01:12:32Z" ]
  } ]
}
```

GET warmstates (2022.02)

Get the warm state file for the specified module instance id and state time.

- moduleInstanceId (string, required): .
- stateTime (string, required): state time. For example: 2020-03-18T15:00:00Z

Response

- binary zip file.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/warmstates?
moduleInstanceId=moduleInstanceId1&stateTime=2020-03-18T15:00:00Z"
```

GET resources/static/{path}/{resourceId} (2023.02)

Get a static web resource from the WebResourceFiles folder. Typically used to provide static resources to the Web OC.

Path parameters

- path (string, optional): One or more subfolders.
- resourceId (string, required): name of a resource in the WebResourceFiles config folder.

Response

- The content of the resource.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/resources/static/style.css"
```

GET displaygroups/plot (2023.01)

Get get a plot from a display group in png format: https://fewsdocs.deltares.nl/webservices/rest-api/v1/#get-/displaygroups/plot

To get all available plots per displayGroupId, the displaygroups/nodes or topology/nodes endpoints can be used.

GET displaygroups/nodes (2023.01)

Get all displaygroups nodes: https://fewsdocs.deltares.nl/webservices/rest-api/v1/#get-/displaygroups/nodes

GET lastrefreshtime(2023.02)

GET the last refresh time of the Web Service. Can be used to make sure indexes have been updated before making any requests. Typical example may be to retrieve a new forecast for which you are sure it is already available in the Delft-FEWS database. Before retrieving the new forecast keep requesting

the last refresh time until you see a change in the last refresh time. This will make sure the indexes have been updated and the new forecast can be found with the WebServices. For this endpoint to work reliable in a setup with multiple web services and a load balancer, it is required that the client that connects to this endpoint is always directed to the same Web Service instance.

Request parameters

- *documentFormat* (string): TEXT (default) or PI_JSON

Response

  - text response with the actual system time. Format: yyyy-MM-ddTHH:mm:ssZ

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/lastrefreshtime"
```

Example text response

```
2020-03-18T15:00:00Z
```

FEWS PI REST Web Service API - Embedded endpoints

The following endpoints are only supported when running the Delft-FEWS Web Services embedded from the Operator Client or Stand Alone.

GET systemtime (2022.02)

GET the system time as configured in the Operator Client of Stand Alone

Request parameters

- *documentFormat* (string): TEXT (default) or PI_JSON

Response

  - text response with the actual system time. Format: yyyy-MM-ddTHH:mm:ssZ

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/systemtime"
```

Example text response

```
2020-03-18T15:00:00Z
```

GET lastupdatetime (2022.02)

Get the last time the timeseries were updated in the Operator Client or a Stand Alone. It returns the last time the main time series dialog loaded time series. Time series dialog will only reload the time series when displayed time series are changed in the database or the user selected something different. The method was implemented for a very specific use case and is not recommended to be used anymore.

Request parameters

- *documentFormat* (string): TEXT (default) or PI_JSON

Response

  - text response with the last update time. Format: yyyy-MM-ddTHH:mm:ssZ

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/lastupdatetime"
```

Example text response

```
2020-03-18T15:00:00Z
```

GET topology/selected (2022.02)

Get the currently selected topology node id in the Operator Client or a Stand Alone

Request parameters

- *documentFormat* (string): PI_JSON

Response

- PI_JSON response

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/topology/selected"
```

Example text response

```
{
"topologyNodeId": "string"
}
```

GET parameters/selected (2022.02)

Get the currently selected parameter ids in the Operator Client or a Stand Alone

Request parameters

- *documentFormat* (string): PI_JSON

Response

- PI_JSON response

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/parameters/selected"
```

Example text response

```
{
  "parameterIds": [
    "string"
  ]
}
```

GET filters/selected (2022.02)

GET the system time as configured in the Operator Client of Stand Alone

Request parameters

- *documentFormat* (string): PI_JSON

Response

- PI_JSON response

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/filters/selected"
```

Example text response

```
{
  "filterIds": [
    "string"
  ]
}
```

GET locations/selected (2022.02)

Get the currently selected location ids in the Operator Client or a Stand Alone

Request parameters

- *documentFormat* (string): PI_JSON

Response

- PI_JSON

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/locations/selected"
```

Example text response

```
{
  "locationIds": [
    "string"
  ]
}
```

GET timeseries/topology/node (2022.02)

Get the time series for the selected topology node

Request parameters

- *documentFormat* (string): PI_XML (default) or PI_JSON
- *topologyNodeId (string): Id of the topology node for which the time series will be returned.*
- *timeZero (date): time zero in format: yyyy-MM-ddTHH:mm:ssZ*
- *startTime (date): Optional startTime for the requested period in format: yyyy-MM-ddTHH:mm:ssZ. If not set, the start time of the zoom period in the first time series dialog will be used.*
- *endTime (date): Optional endTime for the requested period in format: yyyy-MM-ddTHH:mm:ssZ. If not set, the end time of the zoom period in the first time series dialog will be used.*
- *thresholdsVisible (boolean): Set to true to make thresholds visible.*
- *omitMissing (boolean): Toggle omitting or returning of missing values in response. Default is true*
- *useDisplayUnits (boolean): Set to true to use display units*
- *convertDatum (boolean): Set to true to convert datum*

- *documentVersion (string): Document Version*

Response

- Timeseries in PI_XML or PI_JSON format

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/timeseries/topology/node"
```

Example text response

FEWS PI REST Web Service API - Web Operator Client endpoints

The following endpoints are meant to be used by the Web Operator client.

GET weboc/configuration (2023.01)

GET the web OC Configuration

Response

- JSON response with the Web OC configuration.

Example request

```
curl "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/weboc/configuration"
```

Example text response

```
{
  "components": [
    {
      "type": "string",
      "title": "string"
    }
  ],
  "general": {
    "title": "string",
    "icons": {
      "logo": "string",
      "favicon": "string"
    }
  }
}
```