

scalardata service

Description

API scalardata allows users to get scalar data.

URL

```
http://data.oceannetworks.ca/api/scalardata
```

Method	Description	Example
getByLocation	Returns scalar data by given location code and device category	method=getByLocation
getByDevice	Returns scalar data by given device code.	method=getByDevice

* getByStation was deprecated as of October 1, 2017 (see bottom of this page for documentation). It has been replaced by the getByLocation method.

getByLocation

Returns scalar data in JSON format by given location code and device category code.

Parameters

Parameter	Type	Description	Example
Required			
token	string	All Web Services require a token. This can be generated at https://data.oceannetworks.ca/Profile . Click on the "Web Services" tab and click "Generate Token".	token=YOUR_TOKEN_HERE
locationCode	string	Return scalar data from a specific Location . <ul style="list-style-type: none">Location Code must be valid.Specific Location Codes can be obtained using the locations service.	locationCode=KVIP.C2
deviceCategoryCode	string	Return scalar data belonging to a specific Device Category Code . <ul style="list-style-type: none">Device Category Code must be valid.Specific Device Category Codes can be obtained using the deviceCategories service.	deviceCategoryCode=BPR
Optional			
propertyCode	string	Return scalar data from device for a comma separated list of Properties . <ul style="list-style-type: none">Specific propertyCode can be obtained using the properties service.propertyCode<ul style="list-style-type: none">must be validexact matchis case sensitive	propertyCode=seawatertemperature, totalpressure
sensorCategoryCodes	string	A comma separated list of sensor code names. If missing, the system will return all search scalar data within the location code, device category pair, and propertyCode (if given). To discover the sensorCategoryCodes, poll this service with returnOptions=excludeScalarData and check the sensorData response.	sensorCategoryCodes= pressure, borehole_pressure1,oxygen
dateFrom	date	Return scalar data that has a timestamp on or after a specific date/time. Accepted DateTime formats: <ul style="list-style-type: none">yyyy-MM-dd'T'HH:mm:ss.SSS'Z' (ISO 8601 Extended)yyyy-MM-dd (ISO 8601 Extended)PnYnMnDTnHnMnS (ISO 8601 Duration) If not specified, the default value is the beginning of time. <ul style="list-style-type: none">DateTime is represented in Coordinated Universal Time (UTC).ISO 8601 Extended format without a time will be assumed to mean midnight (T00:00:00.000Z).Queries with both dateFrom and dateTo in the ISO 8601 Duration format will not be accepted.	dateFrom=2010-07-27T00:00:00.000Z dateFrom=2010-07-27 dateFrom=-P1DT1H <ul style="list-style-type: none">Previous 1 day and 1 hour, relative to the dateTo. Note the '-' before the P.

dateTo	date	<p>Return scalar data that has a timestamp before a specific date/time.</p> <p>Accepted DateTime formats:</p> <ul style="list-style-type: none"> • yyyy-MM-dd'T'HH:mm:ss.SSS'Z' (ISO 8601 Extended) • yyyy-MM-dd (ISO 8601 Extended) • PnYnMnDTnHnMnS (ISO 8601 Duration) <p>If not specified, the default value is the end of time.</p> <ul style="list-style-type: none"> • DateTime is represented in Coordinated Universal Time (UTC). • ISO 8601 Extended format without a time will be assumed to mean midnight (T00:00:00.000Z). • Queries with both dateFrom and dateTo in the ISO 8601 Duration format will not be accepted. 	<p>dateTo=2016-08-01T00:00:00.000Z</p> <p>dateTo=2016-08-01</p> <p>dateTo=PT12H30M</p> <ul style="list-style-type: none"> • Next 12 hours and 30 minutes, relative to the dateFrom.
metadata	string	<p>There are two values:</p> <ul style="list-style-type: none"> • Minimum (default) (<i>case insensitive</i>) <ul style="list-style-type: none"> ◦ provides only basic property information • Full (<i>case insensitive</i>) <ul style="list-style-type: none"> ◦ provides all property information <p>The metadata is for the latest deployment overlapping the time range specified by dateFrom and dateTo.</p>	metadata=Minimum
rowLimit	integer	<p>Limits the number of scalar data rows returned for each sensor code. If rowLimit is missing, is an invalid number, or is over 100,000, the maximum value of 100,000 will be used.</p>	rowLimit=80000
outputFormat	string	<p>There are two values:</p> <ul style="list-style-type: none"> • Array (default) (<i>case insensitive</i>) <ul style="list-style-type: none"> ◦ Data is output as three arrays for a given sensor: a sampleTime array, a values array, and a qaqcFlags array. • Object (<i>case insensitive</i>) <ul style="list-style-type: none"> ◦ Data is output as a list of objects for a given sensor. Inside each data object, there are sampleTime, value, and qaqcFlag lists. 	outputFormat=Object
getLatest	boolean	<p>Specifies whether or not the latest scalar data readings should be returned first.</p> <p>getLatest = false: default</p> <p>getLatest = true: readings are returned and pagination is done from latest to earliest.</p>	getLatest=true
qualityControl	string	<p>accepted values:</p> <ul style="list-style-type: none"> • raw <ul style="list-style-type: none"> ◦ returns raw scalar data • clean <ul style="list-style-type: none"> ◦ returns clean scalar data <p>The default value when no qualityControl has been provided is: qualityControl = clean.</p> <p>To get resampled data values: qualityControl = clean.</p>	qualityControl=clean
resampleType	string	<p>accepted values:</p> <ul style="list-style-type: none"> • avg <ul style="list-style-type: none"> ◦ gets the average value in each resamplePeriod • avgMinMax <ul style="list-style-type: none"> ◦ gets the average, minimum, and maximum values in each resample period • minMax <ul style="list-style-type: none"> ◦ gets the minimum and maximum values in each resample period <p>The resampleType option requires that a resamplePeriod has been added. If there is a resamplePeriod, but no resampleType, resampleType defaults to avgMinMax. The value of 'none' is not accepted, however that is the effective default when both the resampleType and resamplePeriod parameters are left off / not included.</p>	resampleType=avgMinMax

resamplePeriod	integer	<p>To resample clean scalar data the resample period is set in seconds. The resamplePeriod parameter is required for the resampleType parameter.</p> <p>accepted values:</p> <ul style="list-style-type: none"> • 1 (1 Second) • 5 (5 Seconds) • 10 (10 Seconds) • 15 (15 Seconds) • 30 (30 Seconds) • 60 (1 Minute) • 300 (5 Minutes) • 600 (10 Minutes) • 900 (15 Minutes) • 1800 (30 Minutes) • 3600 (1 Hour) • 7200 (2 Hours) • 14400 (4 Hours) • 21600 (6 Hours) • 43200 (12 Hours) • 86400 (1 Day) • 172800 (2 Days) • 259200 (3 Days) • 604800 (1 Week) • 1209600 (2 Weeks) • 2592000 (30 Days) <p>'0' or 'none' are not accepted. To not resample the data, do not include this parameter.</p>	resamplePeriod=3600
fillGaps	boolean	if true fills scalar data gaps with NaN. By default data gaps are filled.	fillGaps=true
sensorsToInclude	string	<p>accepted values:</p> <ul style="list-style-type: none"> • original <ul style="list-style-type: none"> ◦ returns scalar data from original sensors • externallyDerived <ul style="list-style-type: none"> ◦ returns scalar data from externally derived sensors <p>The default value when no sensorsToInclude have been provided is: sensorsToInclude=original. The sensorsToInclude=externallyDerived option requires that qualityControl either not be set or be set to qualityControl=clean</p>	sensorsToInclude=original
returnOptions	string	<p>accepted values:</p> <ul style="list-style-type: none"> • excludeScalarData <p>When set to excludeScalarData the service response will not contain the sensordata. data field. This is useful for users interested in discovering sensorCategoryCodes and other metadata without having to wait for / handle data.</p>	returnOptions=excludeScalarData

Response

Success

```
{
  "metadata": {
    "boundingBox": {
      "maxDepth":<Double>,
      "maxLat":<Double>,
      "maxLon":<Double>,
      "minDepth":<Double>,
      "minLat":<Double>,
      "minLon":<Double>
    },
    "depth":<Double>,
    "deviceCategoryCode":<String>,
    "lat":<Double>,
    "locationName":<String>,
    "lon":<Double>
  },
  "parameters": {
    "metadata":<String>,
    "method": "getByLocation",
    "dateTo":<String (yyyy-MM-dd'T'HH:mm:ss.SSS'Z')>,
    "rowLimit":<String>,
    "locationCode":<String>,
    "deviceCategoryCode":<String>,
    "sensorCategoryCodes":<String>,

```

```

        "dateFrom":<String (yyyy-MM-dd'T'HH:mm:ss.SSS'Z')>,
        "outputFormat" : <String>,
        "token":<String (GUID)>,
        "resamplePeriod": <String>,
        "resampleType": <String>,
        "qualityControl": <String>,
        "fillGaps": <Boolean>,
        "sensorsToInclude": <String>
    },
    "next": {
        "parameters": {
            "metadata":<String>,
            "method": "getByLocation",
            "dateTo":<String (yyyy-MM-dd'T'HH:mm:ss.SSS'Z')>,
            "rowLimit":<String>,
            "locationCode":<String>,
            "deviceCategoryCode":<String>,
            "sensorCategoryCodes":<String>,
            "dateFrom":<String (yyyy-MM-dd'T'HH:mm:ss.SSS'Z')>,
            "outputFormat" : <String>,
            "token":<String (GUID)>,
            "resamplePeriod": <String>,
            "resanmpleType": <String>,
            "qualityControl": <String>,
            "fillGaps": <Boolean>
            "sensorsToInclude": <String>
        },
        "url": <String (URL)>
    },
    "queryUrl": <String (URL)>,
    "sensorData": [
        {
            "data": {
                "qaqcFlags": <Array of integers>,
                "sampleTimes": <Array of String (yyyy-MM-dd'T'HH:mm:ss.SSS'Z')>,
                "values": <Array of doubles>,
                "min": <Array of doubles>,
                "max": <Array of doubles>,
                "counts": <Array of integers>
            },
            "sensorCategoryCode":<String>,
            "sensorCode":<String>,
            "sensorName":<String>,
            "unitOfMeasure":<String>
            "actualSamples":<Integer>
        },
        ...
    ]
}

```

Property	Type	Description	Example
<ul style="list-style-type: none"> parameters 	Object	An object containing the set of parameters making up the query to get the next set of data in the requested date range. The parameters will include all parameters passed into the original URL.	
<ul style="list-style-type: none"> <ul style="list-style-type: none"> parameters.metadata 	String	As passed into the original call	"minimum"

<ul style="list-style-type: none"> parameters.method 	String	getByLocation, as passed into the original call.	"getByLocation"
<ul style="list-style-type: none"> parameters.dateTo 	String	As passed into the original call	"2015-02-25T00:00:01.000Z"
<ul style="list-style-type: none"> parameters.roleLimit 	String	As passed into the original call	"10"
<ul style="list-style-type: none"> parameters.locationCode 	String	As passed into the original call	"NC27"
<ul style="list-style-type: none"> parameters.deviceCategoryCode 	String	As passed into the original call	"CORK"
<ul style="list-style-type: none"> parameters.sensorCategoryCodes 	String	As passed into the original call	"pressure"

◦ parameters. date From	String	This is set to 1 millisecond more than the date of the last reading sent in the current call.	"2015-02-25T00:00:01.189Z"
◦ parameters. output Format	String	As passed into the original call	"object"
◦ parameters. token	String	As passed into the original call	"ffffff-ffff-ffff-ffffff"
• parameters. resamplePeriod	String	As passed into the original call	15
• parameters. resampleType	String	As passed into the original call	avg
• parameters. qualityControl	String	As passed into the original call	clean
• parameters. dataGaps	Boolean	As passed into the original call	false
• parameters. sensors ToInclude	String	As passed into the original call	original
next	Object	An object containing information on how to make a rawdata call if the current call was unable to return all the data as a limit was exceeded. This is null if there is no more data in the requested range.	
• next. parameters	Object	An object containing the set of parameters making up the query to get the next set of data in the requested date range. The parameters will include all parameters passed into the original URL.	

<ul style="list-style-type: none"> ◦ next parameters metadata 	String	As passed into the original call	"minimum"
<ul style="list-style-type: none"> ◦ next parameters method 	String	getByLocation, as passed into the original call.	"getByLocation"
<ul style="list-style-type: none"> ◦ next parameters date To 	String	As passed into the original call	"2015-02-25T00:00:01.000Z"
<ul style="list-style-type: none"> ◦ next parameters rolimit 	String	As passed into the original call	"10"
<ul style="list-style-type: none"> ◦ next parameters location Code 	String	As passed into the original call	"NC27"

<ul style="list-style-type: none"> next parameters deviceCategoryCode 	String	As passed into the original call	"CORK"
<ul style="list-style-type: none"> next parameters sensorCategoryCodes 	String	As passed into the original call	"pressure"
<ul style="list-style-type: none"> next parameters dateFrom 	String	This is set to 1 millisecond more than the date of the last reading sent in the current call.	"2015-02-25T00:00:01.189Z"
<ul style="list-style-type: none"> next parameters outputFormat 	String	As passed into the original call	"object"
<ul style="list-style-type: none"> next parameter token 	String	As passed into the original call	"YOUR_TOKEN_HERE"

<ul style="list-style-type: none"> next.parameters.sensorsToInclude 	String	As passed into the original call	"original"
<ul style="list-style-type: none"> next.url 	String	The complete URL that can be used to get the next set of data in the requested range.	"https://data.oceannetworks.ca/api/scalardata?method=getByLocation&dateTo=2015-02-25T00%3A00%3A01.000Z&locationCode=NC27&deviceCategoryCode=CORK&dateFrom=2015-02-25T00%3A00%3A01.189Z&token=YOUR_TOKEN_HERE"
queryUrl	String	URL of current query	"https://data.oceannetworks.ca/api/scalardata?method=getByLocation&dateTo=2015-02-25T00%3A00%3A01.000Z&locationCode=NC27&deviceCategoryCode=CORK&dateFrom=2015-02-25T00%3A00%3A00.000Z&token=YOUR_TOKEN_HERE"
sensorData	Array of objects	Data sets returned for all applicable sensors	
<ul style="list-style-type: none"> sensorData.data 	Object	Data sets returned for a single sensor	
<ul style="list-style-type: none"> <ul style="list-style-type: none"> sensorsOrData.data.qcFlags 	Array of integers	<p>QAQC test results of each data point—guidelines for what each number means:</p> <ul style="list-style-type: none"> 0 No quality control on data 1 Data passed all tests 2 Data probably good 3 Data probably bad. Failed minor tests 4 Data bad. Failed major tests 7 Averaged value 8 Interpolated value 9 Missing data 	[1]
<ul style="list-style-type: none"> <ul style="list-style-type: none"> sensorsOrData.data.sampleTimes 	Array of strings	Sampling times of sensor data returned, formatted as <code>YYYY-MM-dd'T'HH:mm:ss.SSS'Z'</code>	["2015-02-25T00:00:00.188Z"]
<ul style="list-style-type: none"> <ul style="list-style-type: none"> sensorsOrData.data.values 	Array of doubles	Values of sensor data returned, units of measurement are given in <code>sensorData.unitOfMeasure</code>	[2706.1145305467917]
<ul style="list-style-type: none"> sensorData.data.min 	Array of doubles	only in the response if min values are part of the <code>resampleType</code> . The array contains the minimum scalar data value in each resample period. Units of measurement are given in <code>sensorData.unitOfMeasure</code> .	[2706.1145305467917]
<ul style="list-style-type: none"> sensorData.data.max 	Array of doubles	only in the response if max values are part of the <code>resampleType</code> . The array contains the maximum scalar data value in each resample period. Units of measurement are given in <code>sensorData.unitOfMeasure</code> .	[2706.1145305467917]

<ul style="list-style-type: none"> • sensorData.data.counts 	Array of Integers	only in the response if a resample period is selected. contains the count of samples in each period	[3600]
<ul style="list-style-type: none"> • sensorData.actualSamples 	String	the number of scalar data values returned excluding any NaN values	48
<ul style="list-style-type: none"> • sensorData.sensorCode 	String	Code of sensor, can be null	"sigmaT"
<ul style="list-style-type: none"> • sensorData.sensorName 	String	Name that denotes what the sensor measures	"Uncompensated Borehole Pressure"
<ul style="list-style-type: none"> • sensorData.unitOfMeasure 	String	Unit of measurement or "NA" if none apply; temperature units of measure are "C" for degrees Celsius	"Gregorian days since 19700101T000000Z"
metadata	Object	Optional; only returned if the metadata parameter is set to full	
<ul style="list-style-type: none"> • metadata.boundingBox 	Object	Bounding limits statistics	
<ul style="list-style-type: none"> ◦ metadata.boundingBox.maxDepth 	Double	Maximum depth in meters below water surface (negative numbers denote above)	982.0
<ul style="list-style-type: none"> ◦ metadata.boundingBox.maxLat 	Double	Maximum latitude in degrees north of the equator (negative numbers denote south)	48.316583

◦ metadata.boundingBoxMaximumLon	Double	Maximum longitude in degrees east of the prime meridian (negative numbers denote west)	-126.050796
◦ metadata.boundingBoxMinimumDepth	Double	Minimum depth in meters below water surface (negative numbers denote above)	982.0
◦ metadata.boundingBoxMinimumLat	Double	Minimum latitude in degrees north of the equator (negative numbers denote south)	48.316583
◦ metadata.boundingBoxMinimumLon	Double	Minimum longitude in degrees east of the prime meridian (negative numbers denote west)	-126.050796
• metadata.depth	Double	Depth in meters below water surface (negative numbers denote above)	2660.5
• metadata.deviceCategoryCode	String	Should be the same as next.parameters.deviceCategoryCode	"CORK"
• metadata.lat	Double	Latitude in degrees north of the equator (negative numbers denote south)	47.7626

<ul style="list-style-type: none"> • metadata.a.locationName 	String	Name of location	"ODP 1026"
<ul style="list-style-type: none"> • metadata.a.lon 	Double	Longitude in degrees east of the prime meridian (negative numbers denote west)	-127.759212

Note: Not all the properties above may be included in the resulting JSON, especially parameters that were not in the original call.

Example

[https://data.oceannetworks.ca/api/scalardata?](https://data.oceannetworks.ca/api/scalardata?method=getByLocation&token=YOUR_TOKEN_HERE&locationCode=NC27&deviceCategoryCode=CORK&dateFrom=2015-02-25T00:00:00.000Z&dateTo=2015-02-25T00:00:01.000Z&propertyCode=totalpressure&sensorCategoryCodes=pressure)

[method=getByLocation&token=YOUR_TOKEN_HERE&locationCode=NC27&deviceCategoryCode=CORK&dateFrom=2015-02-25T00:00:00.000Z&dateTo=2015-02-25T00:00:01.000Z&propertyCode=totalpressure&sensorCategoryCodes=pressure](https://data.oceannetworks.ca/api/scalardata?method=getByLocation&token=YOUR_TOKEN_HERE&locationCode=NC27&deviceCategoryCode=CORK&dateFrom=2015-02-25T00:00:00.000Z&dateTo=2015-02-25T00:00:01.000Z&propertyCode=totalpressure&sensorCategoryCodes=pressure)

```
{
  "next": null,
  "parameters": {
    "dateFrom": "2015-02-25T00:00:00.000Z",
    "dateTo": "2015-02-25T00:00:01.000Z",
    "deviceCategoryCode": "CORK",
    "fillGaps": true,
    "getLatest": false,
    "locationCode": "NC27",
    "metaData": "Minimum",
    "method": "getByLocation",
    "outputFormat": "Array",
    "propertyCode": ["totalpressure"],
    "qualityControl": "clean",
    "resamplePeriod": null,
    "resampleType": null,
    "rowLimit": 100000,
    "sensorCategoryCodes": "pressure",
    "sensorsToInclude": "original",
    "token": "TOKEN"
  },
  "queryUrl": "https://data.oceannetworks.ca/api/scalardata?method=getByLocation&token=TOKEN&locationCode=NC27&deviceCategoryCode=CORK&dateFrom=2015-02-25T00:00:00.000Z&dateTo=2015-02-25T00:00:01.000Z&propertyCode=totalpressure&sensorCategoryCodes=pressure",
  "sensorData": [{
    "actualSamples": 1,
    "data": {
      "qaqcFlags": [1],
      "sampleTimes": ["2015-02-25T00:00:00.968Z"],
      "values": [2707.312585039748]
    },
    "sensorCategoryCode": "pressure",
    "sensorCode": "Pressure",
    "sensorName": "Uncompensated Seafloor Pressure",
    "unitOfMeasure": "decibar"
  }]
}
```

Example with outputFormat option

[https://data.oceannetworks.ca/api/scalardata?](https://data.oceannetworks.ca/api/scalardata?method=getByLocation&token=YOUR_TOKEN_HERE&locationCode=NC27&deviceCategoryCode=CORK&dateFrom=2015-02-25T00:00:00.000Z&dateTo=2015-02-25T00:00:02.000Z&propertyCode=totalpressure&outputFormat=object)

[method=getByLocation&token=YOUR_TOKEN_HERE&locationCode=NC27&deviceCategoryCode=CORK&dateFrom=2015-02-25T00:00:00.000Z&dateTo=2015-02-25T00:00:02.000Z&propertyCode=totalpressure&outputFormat=object](https://data.oceannetworks.ca/api/scalardata?method=getByLocation&token=YOUR_TOKEN_HERE&locationCode=NC27&deviceCategoryCode=CORK&dateFrom=2015-02-25T00:00:00.000Z&dateTo=2015-02-25T00:00:02.000Z&propertyCode=totalpressure&outputFormat=object)

```

{
  "next": null,
  "parameters": {
    "dateFrom": "2015-02-25T00:00:00.000Z",
    "dateTo": "2015-02-25T00:00:02.000Z",
    "deviceCategoryCode": "CORK",
    "fillGaps": true,
    "getLatest": false,
    "locationCode": "NC27",
    "metaData": "Minimum",
    "method": "getByLocation",
    "outputFormat": "object",
    "propertyCode": ["totalpressure"],
    "qualityControl": "clean",
    "resamplePeriod": null,
    "resampleType": null,
    "rowLimit": 100000,
    "sensorCategoryCodes": null,
    "sensorsToInclude": "original",
    "token": "TOKEN"
  },
  "queryUrl": "https://data.oceannetworks.ca/api/scalardata?method=getByLocation&token=TOKEN&locationCode=NC27&deviceCategoryCode=CORK&dateFrom=2015-02-25T00:00:00.000Z&dateTo=2015-02-25T00:00:02.000Z&propertyCode=totalpressure&outputFormat=object",
  "sensorData": [{
    "actualSamples": 2,
    "data": [{
      "qaqcFlag": 1,
      "sampleTime": "2015-02-25T00:00:00.968Z",
      "value": 2706.2404888151023
    }, {
      "qaqcFlag": 1,
      "sampleTime": "2015-02-25T00:00:01.969Z",
      "value": 2706.238440580094
    }
  ],
  "sensorCategoryCode": "borehole_pressure1",
  "sensorCode": "borehole_pressure1",
  "sensorName": "Uncompensated Borehole Pressure",
  "unitOfMeasure": "decibar"
}, {
  "actualSamples": 2,
  "data": [{
    "qaqcFlag": 1,
    "sampleTime": "2015-02-25T00:00:00.968Z",
    "value": 2707.312585039748
  }, {
    "qaqcFlag": 1,
    "sampleTime": "2015-02-25T00:00:01.969Z",
    "value": 2707.3135371260782
  }
  ],
  "sensorCategoryCode": "pressure",
  "sensorCode": "Pressure",
  "sensorName": "Uncompensated Seafloor Pressure",
  "unitOfMeasure": "decibar"
}]
}

```

Example with resamplePeriod option

https://data.oceannetworks.ca/api/scalardata?locationCode=NCBC&deviceCategoryCode=BPR&dateFrom=2019-11-23T23%3A15%3A51.109Z&dateTo=2019-11-23T23%3A45%3A51.109Z&sensorCategoryCodes=pressure&method=getByLocation&token=YOUR_TOKEN_HERE&resamplePeriod=600&fillGaps=false&resampleType=avgminmax

```

{
  "next": null,
  "parameters": {
    "dateFrom": "2019-11-23T23:15:51.109Z",
    "dateTo": "2019-11-23T23:45:51.109Z",
    "deviceCategoryCode": "BPR",
    "fillGaps": false,
    "getLatest": false,
    "locationCode": "NCBC",
    "metaData": "Minimum",
    "method": "getByLocation",
    "outputFormat": "Array",
    "propertyCode": null,
    "qualityControl": "clean",
    "resamplePeriod": 600,
    "resampleType": "avgminmax",
    "rowLimit": 100000,
    "sensorCategoryCodes": "pressure",
    "sensorsToInclude": "original",
    "token": "TOKEN"
  },
  "queryUrl": "https://data.oceannetworks.ca/api/scalardata?locationCode=NCBC&deviceCategoryCode=BPR&dateFrom=2019-11-23T23%3A15%3A51.109Z&dateTo=2019-11-23T23%3A45%3A51.109Z&sensorCategoryCodes=pressure&method=getByLocation&token=TOKEN&resamplePeriod=600&fillGaps=false&resampleType=avgminmax",
  "sensorData": [{
    "actualSamples": 2,
    "data": {
      "counts": [600, 600],
      "max": [410.2151224819547, 410.17906942902374],
      "min": [410.1460181119412, 410.1308777147202],
      "qaqcFlags": [7, 7],
      "sampleTimes": ["2019-11-23T23:20:00.000Z", "2019-11-23T23:30:00.000Z"],
      "values": [410.18493704546466, 410.15440282136035]
    },
    "sensorCategoryCode": "pressure",
    "sensorCode": "Pressure",
    "sensorName": "Seafloor Pressure",
    "unitOfMeasure": "decibar"
  }]
}

```

Example with multiple property codes option

https://data.oceannetworks.ca/api/scalardata?locationCode=NCBC&deviceCategoryCode=BPR&dateFrom=2019-11-23T23%3A59%3A51.109Z&dateTo=2019-11-23T23%3A59%3A55.109Z&method=getByLocation&token=TOKEN_HERE&propertyCode=totalpressure,seawatertemperature

```

{
  "next": null,
  "parameters": {
    "dateFrom": "2019-11-23T23:59:51.109Z",
    "dateTo": "2019-11-23T23:59:55.109Z",
    "deviceCategoryCode": "BPR",
    "fillGaps": true,
    "getLatest": false,
    "locationCode": "NCBC",
    "metaData": "Minimum",
    "method": "getByLocation",
    "outputFormat": "Array",
    "propertyCode": ["totalpressure", "seawatertemperature"],
    "qualityControl": "clean",
    "resamplePeriod": null,
    "resampleType": null,
    "rowLimit": 100000,
    "sensorCategoryCodes": null,
    "sensorsToInclude": "original",
    "token": "TOKEN_HERE"
  },
  "queryUrl": "https://data.oceannetworks.ca/api/scalardata?
locationCode=NCBC&deviceCategoryCode=BPR&dateFrom=2019-11-23T23%3A59%3A51.109Z&dateTo=2019-11-23T23%3A59%3A55.
109Z&method=getByLocation&token=TOKEN_HERE&propertyCode=totalpressure,seawatertemperature",
  "sensorData": [{
    "actualSamples": 4,
    "data": {
      "qaqcFlags": [1, 1, 1, 1],
      "sampleTimes": ["2019-11-23T23:59:51.446Z", "2019-11-23T23:59:52.446Z", "2019-11-23T23:
59:53.446Z", "2019-11-23T23:59:54.447Z"],
      "values": [410.1160345002942, 410.1155502316816, 410.12320424260855, 410.1133151942577]
    },
    "sensorCategoryCode": "pressure",
    "sensorCode": "Pressure",
    "sensorName": "Seafloor Pressure",
    "unitOfMeasure": "decibar"
  }, {
    "actualSamples": 4,
    "data": {
      "qaqcFlags": [1, 1, 1, 1],
      "sampleTimes": ["2019-11-23T23:59:51.446Z", "2019-11-23T23:59:52.446Z", "2019-11-23T23:
59:53.446Z", "2019-11-23T23:59:54.447Z"],
      "values": [5.7701782267100015, 5.770633494259997, 5.770480759340003, 5.770225222070003]
    },
    "sensorCategoryCode": "temperature",
    "sensorCode": "Temperature",
    "sensorName": "Housing Temperature",
    "unitOfMeasure": "C"
  }, {
    "actualSamples": 4,
    "data": {
      "qaqcFlags": [1, 1, 1, 1],
      "sampleTimes": ["2019-11-23T23:59:51.446Z", "2019-11-23T23:59:52.446Z", "2019-11-23T23:
59:53.446Z", "2019-11-23T23:59:54.447Z"],
      "values": [5.8714500185348575, 5.871427558955646, 5.871432050871547, 5.8714500185348575]
    },
    "sensorCategoryCode": "temperature2",
    "sensorCode": "temperature2",
    "sensorName": "P-Sensor Temperature",
    "unitOfMeasure": "C"
  }
  ]
}

```

getByDevice

Returns scalar data in JSON format by given device code.

Parameters

Parameter	Type	Description	Example
Required			
token	String	All Web Services require a token. This can be generated at https://data.oceannetworks.ca/Profile . Click on the "Web Services" tab and click "Generate Token".	token=YOUR_TO KEN_HERE
deviceCode	String	Return raw data of a specific Device Code . <ul style="list-style-type: none"> Device Code must be valid. Specific Device Codes can be found by simply running the service without this parameter to get a list of all devices. 	deviceCode=ASL ZAP1006
Optional			
sensorCategoryCodes	String	A comma separated list of sensor code names. If missing, the system will return all search scalar data within the location code, device category pair, and propertyCode (if given). To discover the sensorCategoryCodes available, poll this service with returnOptions=excludeScalarData and check the sensorData response.	sensorCategoryCodes= pressure, borehole_pressur e1,oxygen
dateFrom	String	Return raw data that has a timestamp on or after a specific date/time. Accepted DateTime formats: <ul style="list-style-type: none"> yyyy-MM-dd'T'HH:mm:ss.SSS'Z' (ISO 8601 Extended) yyyy-MM-dd (ISO 8601 Extended) PnYnMnDTnHnMnS (ISO 8601 Duration) If not specified, the default value is the beginning of time. <ul style="list-style-type: none"> DateTime is represented in Coordinated Universal Time (UTC). ISO 8601 Extended format without a time will be assumed to mean midnight (T00:00:000.000Z). Queries with both dateFrom and dateTo in the ISO 8601 Duration format will not be accepted. 	dateFrom=2017- 06-08T00:00: 00.000Z dateFrom=-P2Y1 1MT15H30M5S
dateTo	String	Return scalar data that has a timestamp before a specific date/time. Accepted DateTime formats: <ul style="list-style-type: none"> yyyy-MM-dd'T'HH:mm:ss.SSS'Z' (ISO 8601 Extended) yyyy-MM-dd (ISO 8601 Extended) PnYnMnDTnHnMnS (ISO 8601 Duration) If not specified, the default value is the end of time. <ul style="list-style-type: none"> DateTime is represented in Coordinated Universal Time (UTC). ISO 8601 Extended format without a time will be assumed to mean midnight (T00:00:000.000Z). Queries with both dateFrom and dateTo in the ISO 8601 Duration format will not be accepted. 	dateTo=2017-06- 08T01:00:00.000 Z dateTo=P1Y2M1 0DT2H30M
rowLimit	Integer	The limit on the number of scalar data readings to return. If not specified the row limit is 100,000.	rowLimit=300
outputFormat	String	There are two values: <ul style="list-style-type: none"> Array (default or if missing) (<i>case insensitive</i>) <ul style="list-style-type: none"> Data will be output as three arrays for a given device: a lineTypes array, a readings array, and a times array. Object (case insensitive) <ul style="list-style-type: none"> Data will be output as a list of objects for a given device. Inside each data object, there are lineType, scalarData, and sampleTime lists. 	outputFormat=obj ect
getLatest	boolean	Specifies whether or not the latest scalar data readings should be returned first. getLatest = false: default getLatest = true: readings are returned and pagination is done from latest to earliest.	getLatest=true
qualityControl	string	accepted values: <ul style="list-style-type: none"> raw <ul style="list-style-type: none"> returns raw scalar data clean <ul style="list-style-type: none"> returns clean scalar data The default value when no qualityControl has been provided is: qualityControl = clean. To get resampled data values: qualityControl = clean.	qualityControl=cl ean

resampleType	string	<p>accepted values:</p> <ul style="list-style-type: none"> avg <ul style="list-style-type: none"> gets the average value in each resamplePeriod avgMinMax <ul style="list-style-type: none"> gets the average, minimum, and maximum values in each resample period minMax <ul style="list-style-type: none"> gets the minimum and maximum values in each resample period <p>The default value when no resampleType has been provided is: resampleType=avgMinMax.</p> <p>The resmapleType option requires that a samplePeriod has been added</p>	resampleType=avgMinMax
resamplePeriod	integer	<p>To resample clean scalar data the resample period is set in seconds. The resamplePeriod parameter is required for the resampleType parameter.</p> <p>accepted values:</p> <ul style="list-style-type: none"> <ul style="list-style-type: none"> 1 (1 Second) 5 (5 Seconds) 10 (10 Seconds) 15 (15 Seconds) 30 (30 Seconds) 60 (1 Minute) 300 (5 Minutes) 600 (10 Minutes) 900 (15 Minutes) 1800 (30 Minutes) 3600 (1 Hour) 7200 (2 Hours) 14400 (4 Hours) 21600 (6 Hours) 43200 (12 Hours) 86400 (1 Day) 172800 (2 Days) 259200 (3 Days) 604800 (1 Week) 1209600 (2 Weeks) 2592000 (30 Days) <p>'0' or 'none' are not accepted. To not resample the data, do not include this parameter.</p>	resamplePeriod=3600
fillGaps	boolean	if true fills scalar data gaps with NaN. By default data gaps are filled.	fillGaps=true
sensorsToInclude	string	<p>accepted values:</p> <ul style="list-style-type: none"> original <ul style="list-style-type: none"> returns scalar data from original sensors externallyDerived <ul style="list-style-type: none"> returns scalar data from externally derived sensors <p>The default value when no sensorsToInclude have been provided is: sensorsToInclude=original.</p> <p>The sensorsToInclude=externallyDerived option requires that qualityControl either not be set or be set to qualityControl=clean, also the device supplied in with the deviceCode parameter must have externally derived sensors.</p>	sensorsToInclude=original
returnOptions	string	<p>accepted values:</p> <ul style="list-style-type: none"> excludeScalarData <p>When set to excludeScalarData the service response will not contain the sensordata.data field. This is useful for users interested in discovering sensorCategoryCodes and other metadata without having to wait for / handle data.</p>	returnOptions=excludeScalarData

Response

Success (outputFormat = Array)

Sample Response

```
{
  "next": {
    "parameters": {
      "method": "getByDevice",
      "dateTo": "2019-02-16T02:00:00.010Z",
      "deviceCode": "SBECTD19p7027",
      "rowLimit": "2",
      "dateFrom": "2019-02-16T00:40:41.643Z",
      "token": "TOKEN"
    },
    "url": "https://data.oceannetworks.ca/api/scalardata?method=getByDevice&dateTo=2019-02-16T02%3A00%3A00.010Z&deviceCode=SBECTD19p7027&rowLimit=2&dateFrom=2019-02-16T00%3A40%3A41.643Z&token=TOKEN"
  }
}
```

```

    },
    "parameters": {
        "dateFrom": "2019-02-16T00:40:39.319Z",
        "dateTo": "2019-02-16T02:00:00.010Z",
        "deviceCode": "SBECTD19p7027",
        "fillGaps": true,
        "getLatest": false,
        "metaData": "Minimum",
        "method": "getByDevice",
        "outputFormat": "Array",
        "qualityControl": "clean",
        "resamplePeriod": null,
        "resampleType": null,
        "rowLimit": 2,
        "sensorsToInclude": "original",
        "token": "TOKEN"
    },
    "queryURL": "https://data.oceannetworks.ca/api/scalardata?method=getByDevice&dateTo=2019-02-16T02%3A00%3A00.010Z&deviceCode=SBECTD19p7027&rowLimit=2&dateFrom=2019-02-16T00%3A40%3A39.319Z&token=TOKEN",
    "sensorData": [{
        "actualSamples": 3,
        "data": {
            "qaqcFlags": [1, 1],
            "sampleTimes": ["2019-02-16T00:40:39.644Z", "2019-02-16T00:40:40.643Z"],
            "values": [3.28515, 3.28523]
        },
        "sensorCategoryCode": "conductivity",
        "sensorCode": "cond",
        "sensorName": "Conductivity",
        "unitOfMeasure": "S/m"
    }, {
        "actualSamples": 3,
        "data": {
            "qaqcFlags": [1, 1],
            "sampleTimes": ["2019-02-16T00:40:39.644Z", "2019-02-16T00:40:40.643Z"],
            "values": [1030.0017535948343, 1030.0013900365639]
        },
        "sensorCategoryCode": "density",
        "sensorCode": "density",
        "sensorName": "Density",
        "unitOfMeasure": "kg/m3"
    }, {
        "actualSamples": 3,
        "data": {
            "qaqcFlags": [1, 1],
            "sampleTimes": ["2019-02-16T00:40:39.644Z", "2019-02-16T00:40:40.643Z"],
            "values": [649.54, 649.543]
        },
        "sensorCategoryCode": "pressure",
        "sensorCode": "Pressure",
        "sensorName": "Pressure",
        "unitOfMeasure": "decibar"
    }, {
        "actualSamples": 3,
        "data": {
            "qaqcFlags": [1, 1],
            "sampleTimes": ["2019-02-16T00:40:39.644Z", "2019-02-16T00:40:40.643Z"],
            "values": [34.1337, 34.1334]
        },
        "sensorCategoryCode": "salinity",
        "sensorCode": "salinity",
        "sensorName": "Practical Salinity",
        "unitOfMeasure": "psu"
    }, {
        "actualSamples": 3,
        "data": {
            "qaqcFlags": [1, 1],
            "sampleTimes": ["2019-02-16T00:40:39.644Z", "2019-02-16T00:40:40.643Z"],
            "values": [27.00702584186911, 27.006664020993867]
        },
        "sensorCategoryCode": "sigma_t",

```

```

        "sensorCode": "sigmaT",
        "sensorName": "Sigma-t",
        "unitOfMeasure": "kg/m3"
    }, {
        "actualSamples": 3,
        "data": {
            "qaqcFlags": [1, 1],
            "sampleTimes": ["2019-02-16T00:40:39.644Z", "2019-02-16T00:40:40.643Z"],
            "values": [27.01278012868579, 27.012419543747]
        },
        "sensorCategoryCode": "sigma_theta",
        "sensorCode": "SIGMA_THETA",
        "sensorName": "Sigma-theta (0 dbar)",
        "unitOfMeasure": "kg/m3"
    }, {
        "actualSamples": 3,
        "data": {
            "qaqcFlags": [1, 1],
            "sampleTimes": ["2019-02-16T00:40:39.644Z", "2019-02-16T00:40:40.643Z"],
            "values": [1479.624, 1479.628]
        },
        "sensorCategoryCode": "sound_speed",
        "sensorCode": "Sound_Speed",
        "sensorName": "Sound Speed",
        "unitOfMeasure": "m/s"
    }, {
        "actualSamples": 3,
        "data": {
            "qaqcFlags": [1, 1],
            "sampleTimes": ["2019-02-16T00:40:39.644Z", "2019-02-16T00:40:40.643Z"],
            "values": [4.8362, 4.8373]
        },
        "sensorCategoryCode": "temperature",
        "sensorCode": "Temperature",
        "sensorName": "Temperature",
        "unitOfMeasure": "C"
    }
}

```

Property	Type	Description	Example
<ul style="list-style-type: none"> parameters 	Object	An object containing the set of parameters making up the query to get the next set of data in the requested date range. The parameters will include all parameters passed into the original URL.	
<ul style="list-style-type: none"> <ul style="list-style-type: none"> parameters.metadata 	String	As passed into the original call	"minimum"
<ul style="list-style-type: none"> <ul style="list-style-type: none"> parameters.method 	String	getByLocation, as passed into the original call.	"getByDevice"

◦ parameters. date To	String	As passed into the original call	"2015-02-25T00:00:01.000Z"
◦ parameters. row Limit	String	As passed into the original call	"10"
◦ parameters. location Code	String	As passed into the original call	"NC27"
◦ parameters. device Category Code	String	As passed into the original call	"CORK"
◦ parameters. sensor Category Codes	String	As passed into the original call	"pressure"

◦ parameters.dateFrom	String	This is set to 1 millisecond more than the date of the last reading sent in the current call.	"2015-02-25T00:00:01.189Z"
◦ parameters.outputFormat	String	As passed into the original call	"object"
◦ parameters.token	String	As passed into the original call	"ffffff-ffff-ffff-ffffff"
• parameters.resamplePeriod	String	As passed into the original call	15
• parameters.resampleType	String	As passed into the original call	avg
• parameters.qualityControl	String	As passed into the original call	clean
• parameters.dataGaps	Boolean	As passed into the original call	false
• parameters.sensorsToInclude	String	As passed into the original call	original
next	Object	An object containing information on how to make a scalar data call if the current call was unable to return all the data as a limit was exceeded. This is null if there is no more data in the requested range.	
• next.parameters	Object	An object containing the set of parameters making up the query to get the next set of data in the requested date range. The parameters will include all parameters passed into the original URL.	

<ul style="list-style-type: none"> next.parameters.method 	String	getByDevice, as passed into the original call.	"getByDevice"
<ul style="list-style-type: none"> next.parameters.dateTo 	String	As passed into the original call	"2019-02-16T02:00:00.010Z"
<ul style="list-style-type: none"> next.parameters.deviceCode 	String	As passed into the original call	"SBECTD19p7027"
<ul style="list-style-type: none"> next.parameters.rowLimit 	String	As passed into the original call	"2"
<ul style="list-style-type: none"> next.parameters.dateFrom 	String	This is set to 1 millisecond more than the date of the last reading sent in the current call.	"2019-02-16T00:40:39.319Z"
<ul style="list-style-type: none"> next.parameters.token 	String	As passed into the original call	"ffffffff-ffff-ffff-ffffffff"
<ul style="list-style-type: none"> next.parameters.sensorsToInclude 	String	As passed into the original call	original
<ul style="list-style-type: none"> next.url 	String	The complete URL that can be used to get the next set of data in the requested range.	"https://data.oceannetworks.ca/api/scalardata?method=getByDevice&dateTo=2019-02-16T02%3A00%3A00.010Z&deviceCode=SBECTD19p7027&rowLimit=2&dateFrom=2019-02-16T00%3A40%3A39.319Z&token=ffffffff-ffff-ffff-ffff-ffffffffffff"
queryUrl	String	URL of current query	"https://data.oceannetworks.ca/api/scalardata?method=getByDevice&token=ffffffff-ffff-ffff-ffff-ffffffffffff&deviceCode=SBECTD19p7027&dateFrom=2019-02-16T00:06:26.028Z&dateTo=2019-02-16T02:00:00.010Z&rowLimit=2"
sensorData	Array of objects	Data sets returned for all applicable sensors	
<ul style="list-style-type: none"> sensorData.data 	Object	Data sets returned for a single sensor	
<ul style="list-style-type: none"> sensorData.data.qaqcFlags 	Array of integers	QAQC test results of each data point—guidelines for what each number means: <ul style="list-style-type: none"> 0 No quality control on data 1 Data passed all tests 2 Data probably good 3 Data probably bad. Failed minor tests 4 Data bad. Failed major tests 7 Averaged value 8 Interpolated value 9 Missing data 	[1,1]

<ul style="list-style-type: none"> • sensor Data. data. sample Times 	Array of strings	Sampling times of sensor data returned, formatted as yyyy-MM-dd'T'HH:mm:ss.SSS'Z'	"2019-02-16T00:40:37.316Z", "2019-02-16T00:40:38.323Z"]
<ul style="list-style-type: none"> • sensor Data. data. values 	Array of doubles	Values of sensor data returned, units of measurement are given in sensorData.unitOfMeasure	[3.15891, 3.15889]
<ul style="list-style-type: none"> • sensor Data. data. min 	Array of doubles	only in the response if min values are part of the resampleType. The array contains the minimum scalar data value in each resample period. Units of measurement are given in sensorData.unitOfMeasure.	[2706.1145305467917]
<ul style="list-style-type: none"> • sensor Data. data. max 	Array of doubles	only in the response if max values are part of the resampleType. The array contains the maximum scalar data value in each resample period. Units of measurement are given in sensorData.unitOfMeasure.	[2706.1145305467917]
<ul style="list-style-type: none"> • sensor Data. data. counts 	Array of Integers	only in the response if a resample period is selected. contains the count of samples in each period	[3600]
<ul style="list-style-type: none"> • sensor Data. actualSamples 	String	the number of scalar data values returned excluding any NaN values	48
<ul style="list-style-type: none"> • sensor Data. sensor CategoryCode 	String	Code of sensor, can be null	"conductivity"
<ul style="list-style-type: none"> • sensor Data. sensor Code 	String	Code of sensor, can be null <ul style="list-style-type: none"> • deprecated in favour of sensorCategoryCode 	"cond"
<ul style="list-style-type: none"> • sensor Data. sensor Name 	String	Name that denotes what the sensor measures	"Conductivity"
<ul style="list-style-type: none"> • sensor Data. unitOfMeasure 	String	Unit of measurement or "NA" if none apply; temperature units of measure are "C" for degrees Celsius	"S/m"

Failure

See [External Web Services Standards: Error Messaging](#).

Example (outputFormat object)

<https://data.oceannetworks.ca/api/scalardata?method=getByDevice&token=TOKEN&deviceCode=SBECTD19p7027&dateFrom=2019-02-16T00:05:26.028Z&dateTo=2019-02-16T00:05:29.010Z&outputFormat=object>

```
{
  "next": null,
  "parameters": {
    "dateFrom": "2019-02-16T00:05:26.028Z",
```

```

        "dateTo": "2019-02-16T00:05:29.010Z",
        "deviceCode": "SBECTD19p7027",
        "fillGaps": true,
        "getLatest": false,
        "metaData": "Minimum",
        "method": "getByDevice",
        "outputFormat": "object",
        "qualityControl": "clean",
        "resamplePeriod": null,
        "resampleType": null,
        "rowLimit": 100000,
        "sensorsToInclude": "original",
        "token": "TOKEN"
    },
    "queryURL": "https://data.oceannetworks.ca/api/scalardata?
method=getByDevice&token=TOKEN&deviceCode=SBECTD19p7027&dateFrom=2019-02-16T00:05:26.028Z&dateTo=2019-02-16T00:
05:29.010Z&outputFormat=object",
    "sensorData": [{
        "actualSamples": 3,
        "data": [{
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:26.610Z",
            "value": 3.28074
        }, {
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:27.616Z",
            "value": 3.28092
        }, {
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:28.611Z",
            "value": 3.28085
        }
    ],
    "sensorCategoryCode": "conductivity",
    "sensorCode": "cond",
    "sensorName": "Conductivity",
    "unitOfMeasure": "S/m"
}, {
    "actualSamples": 3,
    "data": [{
        "qaqcFlag": 1,
        "sampleTime": "2019-02-16T00:05:26.610Z",
        "value": 1030.0199905691406
    }, {
        "qaqcFlag": 1,
        "sampleTime": "2019-02-16T00:05:27.616Z",
        "value": 1030.0181011515497
    }, {
        "qaqcFlag": 1,
        "sampleTime": "2019-02-16T00:05:28.611Z",
        "value": 1030.0167458438395
    }
    ],
    "sensorCategoryCode": "density",
    "sensorCode": "density",
    "sensorName": "Density",
    "unitOfMeasure": "kg/m3"
}, {
    "actualSamples": 3,
    "data": [{
        "qaqcFlag": 1,
        "sampleTime": "2019-02-16T00:05:26.610Z",
        "value": 649.402
    }, {
        "qaqcFlag": 1,
        "sampleTime": "2019-02-16T00:05:27.616Z",
        "value": 649.405
    }, {
        "qaqcFlag": 1,
        "sampleTime": "2019-02-16T00:05:28.611Z",
        "value": 649.404
    }
    ],
    "sensorCategoryCode": "pressure",

```



```

        "sensorCode": "Pressure",
        "sensorName": "Pressure",
        "unitOfMeasure": "decibar"
    }, {
        "actualSamples": 3,
        "data": [{
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:26.610Z",
            "value": 34.1474
        }, {
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:27.616Z",
            "value": 34.1456
        }, {
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:28.611Z",
            "value": 34.144
        }
    ],
        "sensorCategoryCode": "salinity",
        "sensorCode": "salinity",
        "sensorName": "Practical Salinity",
        "unitOfMeasure": "psu"
    }, {
        "actualSamples": 3,
        "data": [{
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:26.610Z",
            "value": 27.02498738577401
        }, {
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:27.616Z",
            "value": 27.02313407261704
        }, {
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:28.611Z",
            "value": 27.021786323347214
        }
    ],
        "sensorCategoryCode": "sigma_t",
        "sensorCode": "sigmaT",
        "sensorName": "Sigma-t",
        "unitOfMeasure": "kg/m3"
    }, {
        "actualSamples": 3,
        "data": [{
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:26.610Z",
            "value": 27.03067026862732
        }, {
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:27.616Z",
            "value": 27.02882094005463
        }, {
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:28.611Z",
            "value": 27.027473586958195
        }
    ],
        "sensorCategoryCode": "sigma_theta",
        "sensorCode": "SIGMA_THETA",
        "sensorName": "Sigma-theta (0 dbar)",
        "unitOfMeasure": "kg/m3"
    }, {
        "actualSamples": 3,
        "data": [{
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:26.610Z",
            "value": 1479.38
        }, {
            "qaqcFlag": 1,
            "sampleTime": "2019-02-16T00:05:27.616Z",
            "value": 1479.393
        }, {

```

```

        "qaqcFlag": 1,
        "sampleTime": "2019-02-16T00:05:28.611Z",
        "value": 1479.394
    }],
    "sensorCategoryCode": "sound_speed",
    "sensorCode": "Sound_Speed",
    "sensorName": "Sound Speed",
    "unitOfMeasure": "m/s"
}, {
    "actualSamples": 3,
    "data": [{
        "qaqcFlag": 1,
        "sampleTime": "2019-02-16T00:05:26.610Z",
        "value": 4.773
    }, {
        "qaqcFlag": 1,
        "sampleTime": "2019-02-16T00:05:27.616Z",
        "value": 4.7768
    }, {
        "qaqcFlag": 1,
        "sampleTime": "2019-02-16T00:05:28.611Z",
        "value": 4.7775
    }],
    "sensorCategoryCode": "temperature",
    "sensorCode": "Temperature",
    "sensorName": "Temperature",
    "unitOfMeasure": "C"
}]
}

```

Example (getLatest)

```

{
    "next": {
        "parameters": {
            "getLatest": "true",
            "method": "getByDevice",
            "dateTo": "2019-02-16T00:06:26.612Z",
            "deviceCode": "SBECTD19p7027",
            "rowLimit": "5",
            "dateFrom": "2019-02-16T00:06:26.028Z",
            "token": "TOKEN"
        },
        "url": "https://data.oceannetworks.ca/api/scalardata?
getLatest=true&method=getByDevice&dateTo=2019-02-16T00%3A06%3A26.
612Z&deviceCode=SBECTD19p7027&rowLimit=5&dateFrom=2019-02-16T00%3A06%3A26.028Z&token=TOKEN"
    },
    "parameters": {
        "dateFrom": "2019-02-16T00:06:26.028Z",
        "dateTo": "2019-02-16T00:59:54.321Z",
        "deviceCode": "SBECTD19p7027",
        "fillGaps": true,
        "getLatest": true,
        "metaData": "Minimum",
        "method": "getByDevice",
        "outputFormat": "Array",
        "qualityControl": "clean",
        "resamplePeriod": null,
        "resampleType": null,
        "rowLimit": 5,
        "sensorsToInclude": "original",
        "token": "TOKEN"
    },
    "queryURL": "https://data.oceannetworks.ca/api/scalardata?getLatest=true&method=getByDevice&dateTo=2019-
02-16T00%3A59%3A54.321Z&deviceCode=SBECTD19p7027&rowLimit=5&dateFrom=2019-02-16T00%3A06%3A26.028Z&token=TOKEN",
    "sensorData": [{
        "actualSamples": 6,

```

```

        "data": {
            "qaqcFlags": [1, 1, 1, 1, 1],
            "sampleTimes": ["2019-02-16T00:06:31.611Z", "2019-02-16T00:06:30.611Z", "2019-02-16T00:06:29.611Z", "2019-02-16T00:06:28.616Z", "2019-02-16T00:06:27.611Z"],
            "values": [3.28094, 3.28063, 3.28071, 3.2808, 3.2809]
        },
        "sensorCategoryCode": "conductivity",
        "sensorCode": "cond",
        "sensorName": "Conductivity",
        "unitOfMeasure": "S/m"
    }, {
        "actualSamples": 6,
        "data": {
            "qaqcFlags": [1, 1, 1, 1, 1],
            "sampleTimes": ["2019-02-16T00:06:31.611Z", "2019-02-16T00:06:30.611Z", "2019-02-16T00:06:29.611Z", "2019-02-16T00:06:28.616Z", "2019-02-16T00:06:27.611Z"],
            "values": [1030.0203515299559, 1030.0177887041705, 1030.0175815192931, 1030.0170878468855, 1030.0174763727805]
        },
        "sensorCategoryCode": "density",
        "sensorCode": "density",
        "sensorName": "Density",
        "unitOfMeasure": "kg/m3"
    }, {
        "actualSamples": 6,
        "data": {
            "qaqcFlags": [1, 1, 1, 1, 1],
            "sampleTimes": ["2019-02-16T00:06:31.611Z", "2019-02-16T00:06:30.611Z", "2019-02-16T00:06:29.611Z", "2019-02-16T00:06:28.616Z", "2019-02-16T00:06:27.611Z"],
            "values": [649.402, 649.405, 649.402, 649.402, 649.403]
        },
        "sensorCategoryCode": "pressure",
        "sensorCode": "Pressure",
        "sensorName": "Pressure",
        "unitOfMeasure": "decibar"
    }, {
        "actualSamples": 6,
        "data": {
            "qaqcFlags": [1, 1, 1, 1, 1],
            "sampleTimes": ["2019-02-16T00:06:31.611Z", "2019-02-16T00:06:30.611Z", "2019-02-16T00:06:29.611Z", "2019-02-16T00:06:28.616Z", "2019-02-16T00:06:27.611Z"],
            "values": [34.1481, 34.1448, 34.1447, 34.1443, 34.1449]
        },
        "sensorCategoryCode": "salinity",
        "sensorCode": "salinity",
        "sensorName": "Practical Salinity",
        "unitOfMeasure": "psu"
    }, {
        "actualSamples": 6,
        "data": {
            "qaqcFlags": [1, 1, 1, 1, 1],
            "sampleTimes": ["2019-02-16T00:06:31.611Z", "2019-02-16T00:06:30.611Z", "2019-02-16T00:06:29.611Z", "2019-02-16T00:06:28.616Z", "2019-02-16T00:06:27.611Z"],
            "values": [27.025374920940294, 27.022779129870287, 27.02259908664587, 27.022125064394686, 27.022522746984805]
        },
        "sensorCategoryCode": "sigma_t",
        "sensorCode": "sigmaT",
        "sensorName": "Sigma-t",
        "unitOfMeasure": "kg/m3"
    }, {
        "actualSamples": 6,
        "data": {
            "qaqcFlags": [1, 1, 1, 1, 1],
            "sampleTimes": ["2019-02-16T00:06:31.611Z", "2019-02-16T00:06:30.611Z", "2019-02-16T00:06:29.611Z", "2019-02-16T00:06:28.616Z", "2019-02-16T00:06:27.611Z"],
            "values": [27.03105972530443, 27.028462887967407, 27.028283837275467, 27.0278113399836, 27.02820999845312]
        },
        "sensorCategoryCode": "sigma_theta",
        "sensorCode": "SIGMA_THETA",

```

```

        "sensorName": "Sigma-theta (0 dbar)",
        "unitOfMeasure": "kg/m3"
    }, {
        "actualSamples": 6,
        "data": {
            "qaqcFlags": [1, 1, 1, 1, 1],
            "sampleTimes": ["2019-02-16T00:06:31.611Z", "2019-02-16T00:06:30.611Z", "2019-02-16T00:06:29.611Z", "2019-02-16T00:06:28.616Z", "2019-02-16T00:06:27.611Z"],
            "values": [1479.387, 1479.382, 1479.386, 1479.391, 1479.395]
        },
        "sensorCategoryCode": "sound_speed",
        "sensorCode": "Sound_Speed",
        "sensorName": "Sound Speed",
        "unitOfMeasure": "m/s"
    }, {
        "actualSamples": 6,
        "data": {
            "qaqcFlags": [1, 1, 1, 1, 1],
            "sampleTimes": ["2019-02-16T00:06:31.611Z", "2019-02-16T00:06:30.611Z", "2019-02-16T00:06:29.611Z", "2019-02-16T00:06:28.616Z", "2019-02-16T00:06:27.611Z"],
            "values": [4.7745, 4.7743, 4.7752, 4.7766, 4.7773]
        },
        "sensorCategoryCode": "temperature",
        "sensorCode": "Temperature",
        "sensorName": "Temperature",
        "unitOfMeasure": "C"
    }
}

```

Example (resamplePeriod)

```

{
    "next": null,
    "parameters": {
        "dateFrom": "2019-11-23T23:50:00.000Z",
        "dateTo": "2019-11-24T00:00:00.109Z",
        "deviceCode": "BPR-Folger-59",
        "fillGaps": true,
        "getLatest": false,
        "metaData": "Minimum",
        "method": "getByDevice",
        "outputFormat": "Array",
        "qualityControl": "clean",
        "resamplePeriod": 600,
        "resampleType": "avgMinMax",
        "rowLimit": 100000,
        "sensorsToInclude": "original",
        "token": "TOKEN"
    },
    "queryURL": "https://data.oceannetworks.ca/api/scalardata?deviceCode=BPR-Folger-59&dateFrom=2019-11-23T23%3A50%3A00.000Z&dateTo=2019-11-24T00%3A00%3A00.109Z&method=getByDevice&token=TOKEN&resamplePeriod=600",
    "sensorData": [{
        "actualSamples": 1,
        "data": {
            "counts": [600],
            "max": [18224.00060185185],
            "min": [18223.993668981482],
            "qaqcFlags": [7],
            "sampleTimes": ["2019-11-23T23:50:00.000Z"],
            "values": [18223.99713541667]
        },
        "sensorCategoryCode": "clock",
        "sensorCode": "clock",
        "sensorName": "Instrument Clock",
        "unitOfMeasure": "Gregorian days since 19700101T000000Z"
    }, {
        "actualSamples": 1,

```

```

    "data": {
      "counts": [600],
      "max": [0.016661775226793907],
      "min": [-0.018115118449031797],
      "qaqcFlags": [7],
      "sampleTimes": ["2019-11-23T23:50:00.000Z"],
      "values": [7.484659961238549E-4]
    },
    "sensorCategoryCode": "dart_pressure_residual",
    "sensorCode": "dart_pressure_residual",
    "sensorName": "DART Pressure Residual",
    "unitOfMeasure": "decibar"
  }, {
    "actualSamples": 1,
    "data": {
      "counts": [600],
      "max": [0.0],
      "min": [0.0],
      "qaqcFlags": [7],
      "sampleTimes": ["2019-11-23T23:50:00.000Z"],
      "values": [0.0]
    },
    "sensorCategoryCode": "direct_detider_weighted_threshold",
    "sensorCode": "Direct_Detider_Weighted_Threshold",
    "sensorName": "Direct Detider Weighted Threshold",
    "unitOfMeasure": "NA"
  }, {
    "actualSamples": 1,
    "data": {
      "counts": [600],
      "max": [0.0],
      "min": [0.0],
      "qaqcFlags": [7],
      "sampleTimes": ["2019-11-23T23:50:00.000Z"],
      "values": [0.0]
    },
    "sensorCategoryCode": "kurtosis_weighted_threshold",
    "sensorCode": "Kurtosis_Weighted_Threshold",
    "sensorName": "Kurtosis Weighted Threshold",
    "unitOfMeasure": "NA"
  }, {
    "actualSamples": 1,
    "data": {
      "counts": [600],
      "max": [410.1353110066574],
      "min": [410.1012234744143],
      "qaqcFlags": [7],
      "sampleTimes": ["2019-11-23T23:50:00.000Z"],
      "values": [410.1182347632783]
    },
    "sensorCategoryCode": "pressure",
    "sensorCode": "Pressure",
    "sensorName": "Seafloor Pressure",
    "unitOfMeasure": "decibar"
  }, {
    "actualSamples": 1,
    "data": {
      "counts": [600],
      "max": [0.004869102302980243],
      "min": [-0.006169526778020364],
      "qaqcFlags": [7],
      "sampleTimes": ["2019-11-23T23:50:00.000Z"],
      "values": [-5.749410881530408E-7]
    },
    "sensorCategoryCode": "rayleigh_detector",
    "sensorCode": "Rayleigh_Detector",
    "sensorName": "Seismic Detector",
    "unitOfMeasure": "NA"
  }, {
    "actualSamples": 1,
    "data": {

```

```

        "counts": [600],
        "max": [1.2691588039586816],
        "min": [0.8568423943177923],
        "qaqcFlags": [7],
        "sampleTimes": ["2019-11-23T23:50:00.000Z"],
        "values": [1.0578961478264164]
    },
    "sensorCategoryCode": "rayleigh_signal_to_noise_ratio",
    "sensorCode": "Rayleigh_Signal_To_Noise_Ratio",
    "sensorName": "Seismic Signal to Noise Ratio",
    "unitOfMeasure": "NA"
}, {
    "actualSamples": 1,
    "data": {
        "counts": [600],
        "max": [0.0],
        "min": [0.0],
        "qaqcFlags": [7],
        "sampleTimes": ["2019-11-23T23:50:00.000Z"],
        "values": [0.0]
    },
    "sensorCategoryCode": "rayleigh_weighted_threshold",
    "sensorCode": "Rayleigh_Weighted_Threshold",
    "sensorName": "Seismic Threshold",
    "unitOfMeasure": "NA"
}, {
    "actualSamples": 1,
    "data": {
        "counts": [600],
        "max": [1.0617238224810472],
        "min": [0.7387132597606645],
        "qaqcFlags": [7],
        "sampleTimes": ["2019-11-23T23:50:00.000Z"],
        "values": [0.8750975631784089]
    },
    "sensorCategoryCode": "sta_lta_ratio",
    "sensorCode": "STA_LTA_Ratio",
    "sensorName": "Signal to Noise Ratio",
    "unitOfMeasure": "NA"
}, {
    "actualSamples": 1,
    "data": {
        "counts": [600],
        "max": [0.0],
        "min": [0.0],
        "qaqcFlags": [7],
        "sampleTimes": ["2019-11-23T23:50:00.000Z"],
        "values": [0.0]
    },
    "sensorCategoryCode": "sta_lta_ratio_weighted_threshold",
    "sensorCode": "STA_LTA_Ratio_Weighted_Threshold",
    "sensorName": "Signal to Noise Ratio Weighted Threshold",
    "unitOfMeasure": "NA"
}, {
    "actualSamples": 1,
    "data": {
        "counts": [600],
        "max": [5.772419317939999],
        "min": [5.769990245270002],
        "qaqcFlags": [7],
        "sampleTimes": ["2019-11-23T23:50:00.000Z"],
        "values": [5.771207405512597]
    },
    "sensorCategoryCode": "temperature",
    "sensorCode": "Temperature",
    "sensorName": "Housing Temperature",
    "unitOfMeasure": "C"
}, {
    "actualSamples": 1,
    "data": {
        "counts": [600],

```

```

        "max": [5.872855986721493],
        "min": [5.871409591291745],
        "qaqcFlags": [7],
        "sampleTimes": ["2019-11-23T23:50:00.000Z"],
        "values": [5.872147522737963]
    },
    "sensorCategoryCode": "temperature2",
    "sensorCode": "temperature2",
    "sensorName": "P-Sensor Temperature",
    "unitOfMeasure": "C"
}, {
    "actualSamples": 1,
    "data": {
        "counts": [600],
        "max": [3.1184035054379855],
        "min": [2.9657060692477186],
        "qaqcFlags": [7],
        "sampleTimes": ["2019-11-23T23:50:00.000Z"],
        "values": [3.0259129890276735]
    },
    "sensorCategoryCode": "warn_kurtosis",
    "sensorCode": "WARN_Kurtosis",
    "sensorName": "WARN Kurtosis",
    "unitOfMeasure": "NA"
}, {
    "actualSamples": 1,
    "data": {
        "counts": [600],
        "max": [0.0],
        "min": [0.0],
        "qaqcFlags": [7],
        "sampleTimes": ["2019-11-23T23:50:00.000Z"],
        "values": [0.0]
    },
    "sensorCategoryCode": "warn_watcher",
    "sensorCode": "WARN_Watcher",
    "sensorName": "WARN Watcher",
    "unitOfMeasure": "NA"
}, {
    "actualSamples": 1,
    "data": {
        "counts": [600],
        "max": [0.0],
        "min": [0.0],
        "qaqcFlags": [7],
        "sampleTimes": ["2019-11-23T23:50:00.000Z"],
        "values": [0.0]
    },
    "sensorCategoryCode": "warn_watcher_weighted_threshold",
    "sensorCode": "WARN_Watcher_Weighted_Threshold",
    "sensorName": "WARN Watcher Weighted Threshold",
    "unitOfMeasure": "NA"
}
}]
}

```