CBIBS API Cookbook

July 2014

Overview

The CBIBS system has a program-to-program API available for use. This will take data from a client and return the data in either XML or JSON using the xml-rpc and json-rpc libraries.

Here are the functions available to use.

- **ListStationswithParam**
  - **parameters** (constellation[req], parameter[req], api_key[req])
    This will list all of the stations that have data for a given key.
- **RetrieveCurrentReadings**
  - **parameters** (constellation[req], stationid[req], api_key[req])
    This will get all of the current readings for a buoy.
- **QueryData**
  - **parameters** (constellation, stationid, measurement, beg_date, end_date, api_key[req])
    This will query a station for a parameter over a date range.
- **ListPlatformsinConstellation**
  - **parameters** (constellation[req], api_key[req])
    This will list all of the stations
- **ListParamsonPlatform**
  - **parameters**(constellation[req], stationid[req], api_key[req])
    This will list all of the parameters for a given station
- **GetStationStatus**
  - **parameters** (constellation[req], stationid[req], api_key[req])
    This will return the operating status of a station

Example Data

The constellation and api key will not change. There are currently 11 available stations to query. The getstationstatus function will return if the station is currently active. Some sample data is listed below:

constellation: cbibs
api_key: a1655be60c484b273ffec1595f3661b2b9f15c1a
stationid: SN, PL, J, SR, S, N, AN, UP, GR, FL, RC
beg_date: yyyy-mm-dd hh:mm:ss

Example JSON Clients in the php Language

You can download a library to help speed up your development. http://jsonrpcphp.org/

ListStationswithParam

```
<?php
/**
* CDRH JSON-RPC listStationswithParam Function (constellation[req], parameter[req], api_key)
* @param string $p1
* @param string $p2
* @param string $p3
* @param int $debug when 1 (or 2) will enable debugging of the underlying jsonrpc call (defaults to 0)
* @return array (or an jsonrpcresp obj instance if call fails)
*/
function jsonrpc_ListStationsWithParam ($p1, $p2, $p3, $debug=0) {
$client =& new jsonrpc_client('/studs/studs_cdrh/jsonrpc_cdrh/server.php', 'mw.buoybay.org', 0);
$client->return_type = 'jsonrpcvals';
$client->setDebug($debug);
$msg =& new jsonrpcmsg('ListStationsWithParam');
$p1 =& new jsonrpcval($p1, 'string');
$msg->addparam($p1);
$p2 =& new jsonrpcval($p2, 'string');
$msg->addparam($p2);
$p3 =& new jsonrpcval($p3, 'string');
$msg->addparam($p3);
$res =& $client->send($msg, 0, '');
if ($res->faultcode()) return $res; else return php_jsonrpc_decode($res->value());
}
?>
```

retrieveCurrentReadings

```
<?php
/**
* CDRH JSON-RPC retrieveCurrentReadings Function (constellation[req], stationid[req], api_key)
* @param string $p1
* @param string $p2
* @param string $p3
* @param int $debug when 1 (or 2) will enable debugging of the underlying jsonrpc call (defaults to 0)
* @return string (or an jsonrpcresp obj instance if call fails)
*/
function jsonrpc_RetrieveCurrentReadings ($p1, $p2, $p3, $debug=0) {
$client =& new jsonrpc_client('/studs/studs_cdrh/jsonrpc_cdrh/server.php', 'mw.buoybay.org', 0);
$client->return_type = 'jsonrpcvals';
$client->setDebug($debug);
$msg =& new jsonrpcmsg('RetrieveCurrentReadings');
$p1 =& new jsonrpcval($p1, 'string');
$msg->addparam($p1);
$p2 =& new jsonrpcval($p2, 'string');
$msg->addparam($p2);
```

```
$p3 =& new jsonrpcval($p3, 'string');
$msg->addparam($p3);
$res =& $client->send($msg, 0, '');
if ($res->faultcode()) return $res; else return php_jsonrpc_decode($res->value());
}
?>
```

queryData

```
<?php
/**
* CDRH JSON-RPC Query Data function (constellation[req], stationid[req], measurement[req], beg_date[opt],
end_date[opt], api_key[req])
* @param string $p1
* @param string $p2
* @param string $p3
* @param string $p4
* @param string $p5
* @param string $p6
* @param int $debug when 1 (or 2) will enable debugging of the underlying jsonrpc call (defaults to 0)
* @return string (or an jsonrpcresp obj instance if call fails)
*/
function jsonrpc_QueryData ($p1, $p2, $p3, $p4, $p5, $p6, $debug=0) {
$client =& new jsonrpc_client('/studs/studs_cdrh/jsonrpc_cdrh/server.php', 'mw.buoybay.org', 0);
$client->return_type = 'jsonrpcvals';
$client->setDebug($debug);
$msg =& new jsonrpcmsg('QueryData');
$p1 =& new jsonrpcval($p1, 'string');
$msg->addparam($p1);
$p2 =& new jsonrpcval($p2, 'string');
$msg->addparam($p2);
$p3 =& new jsonrpcval($p3, 'string');
$msg->addparam($p3);
$p4 =& new jsonrpcval($p4, 'string');
$msg->addparam($p4);
$p5 =& new jsonrpcval($p5, 'string');
$msg->addparam($p5);
$p6 =& new jsonrpcval($p6, 'string');
$msg->addparam($p6);
$res =& $client->send($msg, 0, '');
if ($res->faultcode()) return $res; else return php_jsonrpc_decode($res->value());
}
?>
```

ListPlatforms

```
<?php
/**
* CDRH JSON-RPC listPlatformsinConstellation function (constellation[req], api_key[req])
* @param string $p1
```

```
* @param string $p2
* @param int $debug when 1 (or 2) will enable debugging of the underlying jsonrpc call (defaults to 0)
* @return string (or an jsonrpcresp obj instance if call fails)
*/
function jsonrpc_ListPlatforms ($p1, $p2, $debug=0) {
$client =& new jsonrpc_client('/studs/studs_cdrh/jsonrpc_cdrh/server.php', 'mw.buoybay.org', 0);
$client->return_type = 'jsonrpcvals';
$client->setDebug($debug);
$msg =& new jsonrpcmsg('ListPlatforms');
$p1 =& new jsonrpcval($p1, 'string');
$msg->addparam($p1);
$p2 =& new jsonrpcval($p2, 'string');
$msg->addparam($p2);
$res =& $client->send($msg, 0, '');
if ($res->faultcode()) return $res; else return php_jsonrpc_decode($res->value());
}
?>
```

ListParams

```
<?php
/**
* CDRH JSON-RPC listParamsonPlatform function (constellation[req], stationid[req], api_key[req])
* @param string $p1
* @param string $p2
* @param string $p3
* @param int $debug when 1 (or 2) will enable debugging of the underlying jsonrpc call (defaults to 0)
* @return string (or an jsonrpcresp obj instance if call fails)
*/
function jsonrpc_ListParameters ($p1, $p2, $p3, $debug=0) {
$client =& new jsonrpc_client('/studs/studs_cdrh/jsonrpc_cdrh/server.php', 'mw.buoybay.org', 0);
$client->return_type = 'jsonrpcvals';
$client->setDebug($debug);
$msg =& new jsonrpcmsg('ListParameters');
$p1 =& new jsonrpcval($p1, 'string');
$msg->addparam($p1);
$p2 =& new jsonrpcval($p2, 'string');
$msg->addparam($p2);
$p3 =& new jsonrpcval($p3, 'string');
$msg->addparam($p3);
$res =& $client->send($msg, 0, '');
if ($res->faultcode()) return $res; else return php_jsonrpc_decode($res->value());
}
?>
```

GetStationStatus

```
<?php
/**
* CDRH JSON-RPC GetStationStatus function (constellation[req], stationid[req], api_key[req])
```

```
* @param string $p1
* @param string $p2
* @param string $p3
* @param int $debug when 1 (or 2) will enable debugging of the underlying jsonrpc call (defaults to 0)
* @return integer (or an jsonrpcresp obj instance if call fails)
*/
function jsonrpc_GetStationStatus ($p1, $p2, $p3, $debug=0) {
$client =& new jsonrpc_client('/studs/studs_cdrh/jsonrpc_cdrh/server.php', 'mw.buoybay.org', 0);
$client->return_type = 'jsonrpcvals';
$client->setDebug($debug);
$msg =& new jsonrpcmsg('GetStationStatus');
$p1 =& new jsonrpcval($p1, 'string');
$msg->addparam($p1);
$p2 =& new jsonrpcval($p2, 'string');
$msg->addparam($p2);
$p3 =& new jsonrpcval($p3, 'string');
$msg->addparam($p3);
$res =& $client->send($msg, 0, '');
if ($res->faultcode()) return $res; else return php_jsonrpc_decode($res->value());
}
?>
```

Example XML Clients

Documentation can be found here: http://php.net/manual/en/book.xmlrpc.php

ListStationswithParam

```
 <?php
/**
* CDRH XML-RPC ListStationswithParam Function (constellation[req], parameter[req], api_key[req])
* @param string $p1
* @param string $p2
* @param string $p3
* @param int $debug when 1 (or 2) will enable debugging of the underlying xmlrpc call (defaults to 0)
* @return array (or an xmlrpcresp obj instance if call fails)
*/
function xmlrpc_xmlrpc_cdrh_ListStationsWithParam ($p1, $p2, $p3, $debug=0) {
$client =& new xmlrpc_client('/studs/studs_cdrh/xmlrpc_cdrh/server.php', 'mw.buoybay.org', 0);
$client->return_type = 'xmlrpcvals';
$client->setDebug($debug);
$msg =& new xmlrpcmsg('xmlrpc_cdrh.ListStationsWithParam');
$p1 =& new xmlrpcval($p1, 'string');
$msg->addparam($p1);
$p2 =& new xmlrpcval($p2, 'string');
$msg->addparam($p2);
$p3 =& new xmlrpcval($p3, 'string');
$msg->addparam($p3);
$res =& $client->send($msg, 0, '');
```

```
if ($res->faultcode()) return $res; else return php_xmlrpc_decode($res->value());
}
?>
```

RetrieveCurrentReadings

```
<?php
/**
* CDRH XML-RPC RetrieveCurrentReadings Function (constellation[req], stationid[req], api_key[req])
* @param string $p1
* @param string $p2
* @param string $p3
* @param int $debug when 1 (or 2) will enable debugging of the underlying xmlrpc call (defaults to 0)
* @return string (or an xmlrpcresp obj instance if call fails)
*/
function xmlrpc_xmlrpc_cdrh_RetrieveCurrentReadings ($p1, $p2, $p3, $debug=0) {
$client =& new xmlrpc_client('/studs/studs_cdrh/xmlrpc_cdrh/server.php', 'mw.buoybay.org', 0);
$client->return_type = 'xmlrpcvals';
$client->setDebug($debug);
$msg =& new xmlrpcmsg('xmlrpc_cdrh.RetrieveCurrentReadings');
$p1 =& new xmlrpcval($p1, 'string');
$msg->addparam($p1);
$p2 =& new xmlrpcval($p2, 'string');
$msg->addparam($p2);
$p3 =& new xmlrpcval($p3, 'string');
$msg->addparam($p3);
$res =& $client->send($msg, 0, '');
if ($res->faultcode()) return $res; else return php_xmlrpc_decode($res->value());
}
?>
```

QueryData

```
 <?php
/**
* CDRH XML-RPC Query Data function (constellation, stationid, measurement, beg_date, end_date,
api_key[req])
* @param string $p1
* @param string $p2
* @param string $p3
* @param string $p4
* @param string $p5
* @param string $p6
* @param int $debug when 1 (or 2) will enable debugging of the underlying xmlrpc call (defaults to 0)
* @return string (or an xmlrpcresp obj instance if call fails)
*/
function xmlrpc_xmlrpc_cdrh_QueryData ($p1, $p2, $p3, $p4, $p5, $p6, $debug=0) {
$client =& new xmlrpc_client('/studs/studs_cdrh/xmlrpc_cdrh/server.php', 'mw.buoybay.org', 0);
$client->return_type = 'xmlrpcvals';
$client->setDebug($debug);
```

```
$msg =& new xmlrpcmsg('xmlrpc_cdrh.QueryData');
$p1 =& new xmlrpcval($p1, 'string');
$msg->addparam($p1);
$p2 =& new xmlrpcval($p2, 'string');
$msg->addparam($p2);
$p3 =& new xmlrpcval($p3, 'string');
$msg->addparam($p3);
$p4 =& new xmlrpcval($p4, 'string');
$msg->addparam($p4);
$p5 =& new xmlrpcval($p5, 'string');
$msg->addparam($p5);
$p6 =& new xmlrpcval($p6, 'string');
$msg->addparam($p6);
$res =& $client->send($msg, 0, '');
if ($res->faultcode()) return $res; else return php_xmlrpc_decode($res->value());
}
?>
```

ListPlatformsinConstellation

```
<?php
/**
* CDRH XML-RPC ListParamsonPlatform function (constellation[req], stationid[req], api_key[req])
* @param string $p1
* @param string $p2
* @param string $p3
* @param int $debug when 1 (or 2) will enable debugging of the underlying xmlrpc call (defaults to 0)
* @return string (or an xmlrpcresp obj instance if call fails)
*/
function xmlrpc_xmlrpc_cdrh_ListParameters ($p1, $p2, $p3, $debug=0) {
$client =& new xmlrpc_client('/studs/studs_cdrh/xmlrpc_cdrh/server.php', 'mw.buoybay.org', 0);
$client->return_type = 'xmlrpcvals';
$client->setDebug($debug);
$msg =& new xmlrpcmsg('xmlrpc_cdrh.ListParameters');
$p1 =& new xmlrpcval($p1, 'string');
$msg->addparam($p1);
$p2 =& new xmlrpcval($p2, 'string');
$msg->addparam($p2);
$p3 =& new xmlrpcval($p3, 'string');
$msg->addparam($p3);
$res =& $client->send($msg, 0, '');
if ($res->faultcode()) return $res; else return php_xmlrpc_decode($res->value());
}
?>
```

ListParamsonPlatform

```
<?php
/**
* CDRH XML-RPC ListParamsonPlatform function (constellation[req], stationid[req], api_key[req])
```

```
* @param string $p1
* @param string $p2
* @param string $p3
* @param int $debug when 1 (or 2) will enable debugging of the underlying xmlrpc call (defaults to 0)
* @return string (or an xmlrpcresp obj instance if call fails)
*/
function xmlrpc_xmlrpc_cdrh_ListParameters ($p1, $p2, $p3, $debug=0) {
$client =& new xmlrpc_client('/studs/studs_cdrh/xmlrpc_cdrh/server.php', 'mw.buoybay.org', 0);
$client->return_type = 'xmlrpcvals';
$client->setDebug($debug);
$msg =& new xmlrpcmsg('xmlrpc_cdrh.ListParameters');
$p1 =& new xmlrpcval($p1, 'string');
$msg->addparam($p1);
$p2 =& new xmlrpcval($p2, 'string');
$msg->addparam($p2);
$p3 =& new xmlrpcval($p3, 'string');
$msg->addparam($p3);
$res =& $client->send($msg, 0, '');
if ($res->faultcode()) return $res; else return php_xmlrpc_decode($res->value());
}
?>
```

GetStationStatus

```
 <?php
/**
* CDRH XML-RPC GetStationStatus function (constellation[req], stationid[req], api_key[req])
* @param string $p1
* @param string $p2
* @param string $p3
* @param int $debug when 1 (or 2) will enable debugging of the underlying xmlrpc call (defaults to 0)
* @return integer (or an xmlrpcresp obj instance if call fails)
*/
function xmlrpc_xmlrpc_cdrh_GetStationStatus ($p1, $p2, $p3, $debug=0) {
$client =& new xmlrpc_client('/studs/studs_cdrh/xmlrpc_cdrh/server.php', 'mw.buoybay.org', 0);
$client->return_type = 'xmlrpcvals';
$client->setDebug($debug);
$msg =& new xmlrpcmsg('xmlrpc_cdrh.GetStationStatus');
$p1 =& new xmlrpcval($p1, 'string');
$msg->addparam($p1);
$p2 =& new xmlrpcval($p2, 'string');
$msg->addparam($p2);
$p3 =& new xmlrpcval($p3, 'string');
$msg->addparam($p3);
$res =& $client->send($msg, 0, '');
if ($res->faultcode()) return $res; else return php_xmlrpc_decode($res->value());
}
?>
```

Example Data Calls and the Results

Example xmlrpc_cdrh.RetrieveCurrentReadings

Payload:

<param><value><string>CBIBS</string></value></param>
<param><value><string>an</string></value></param>
<param><value><string>a1655be60c484b273ffec1595f3661b2b9f15c1a</string></value></param>

Response:

<methodResponse> <params> <param> <value><struct> <member><name>constellation</name>
<value><string>CBIBS</string></value> </member> <member><name>station</name>
<value><string>an</string></value> </member> <member><name>measurement</name> <value><array>
<data> <value><string>air_pressure</string></value> <value><string>air_temperature</string></value>
<value><string>concentration_of_chlorophyll_in_sea_water</string></value>
<value><string>concentration_of_oxygen_in_sea_water</string></value>
<value><string>current_average_direction</string></value>
<value><string>current_average_velocity</string></value> <value><string>heat_index</string></value>
<value><string>latitude</string></value> <value><string>longitude</string></value>
<value><string>relative_humidity</string></value> <value><string>seanettle_prob</string></value>
<value><string>sea_surface_maximum_wave_height</string></value>
<value><string>sea_surface_mean_wave_height</string></value>
<value><string>sea_surface_wave_from_direction</string></value>
<value><string>sea_surface_wave_significant_height</string></value>
<value><string>sea_surface_wind_wave_period</string></value>
<value><string>sea_water_electrical_conductivity</string></value>
<value><string>sea_water_freezing_point</string></value>
<value><string>sea_water_salinity</string></value> <value><string>sea_water_temperature</string></value>
<value><string>simple_turbidity</string></value> <value><string>wave_direction_spread</string></value>
<value><string>wind_chill</string></value> <value><string>wind_from_direction</string></value>
<value><string>wind_speed</string></value> <value><string>wind_speed_of_gust</string></value> </data>
</array></value> </member> <member><name>time</name> <value><array> <data> <value><string>2014-
07-25 17:00:00</string></value> <value><string>2014-07-25 17:00:00</string>

<value><double>24.3</double></value> <value><double>38.9628</double></value> <value><double>-76.447022</double></value> <value><double>49.5</double></value>
<value><double>9.31</double></value> <value><double>0.6</double></value>
<value><double>0.17</double></value> <value><double>264</double></value>
<value><double>0.28</double></value> <value><double>2.5</double></value>
<value><double>1.34</double></value> <value><double>-0.4092</double></value>
<value><double>7.51</double></value> <value><double>26.41</double></value>
<value><double>5.74</double></value> <value><double>66.4</double></value>
<value><double>24.3</double></value> <value><double>311</double></value>
<value><double>3.9</double></value> <value><double>6</double></value> </data> </array></value>
</member> <member><name>units</name> <value><array> <data> <value><string>mb</string></value>
<value><string>C</string></value> <value><string>ug/L</string></value>
<value><string>mg/L</string></value> <value><string>degrees</string></value>
<value><string>mm/sec</string></value> <value><string>C</string></value>
<value><string>degrees</string></value> <value><string>degrees</string></value>
<value><string>percent</string></value> <value><string>percent</string></value>
<value><string>m</string></value> <value><string>m</string></value>
<value><string>degrees</string></value> <value><string>m</string></value>
<value><string>s</string></value> <value><string>S/m</string></value> <value><string>C</string></value>
<value><string>PSU</string></value> <value><string>C</string></value>
<value><string>NTU</string></value> <value><string>degrees</string></value>
<value><string>C</string></value> <value><string>degrees</string></value>
<value><string>m/s</string></value> <value><string>m/s</string></value> </data> </array></value>
</member> </struct></value> </param> </params> </methodResponse>

Example JSON Call

RetrieveCurrentReadings

Payload:

"CBIBS","AN","a1655be60c484b273ffec1595f3661b2b9f15c1a"

Response:

{ "id": 1, "error": null, "result":
{"constellation":"CBIBS","station":"AN","measurement":["air_pressure","air_temperature","concentration_of_chlorophyll_in_sea_water","concentration_of_oxygen_in_sea_water","current_average_direction","current_average_velocity","heat_index","latitude","longitude","relative_humidity","seanettle_prob","sea_surface_maximum_wave_height","sea_surface_mean_wave_height","sea_surface_wave_from_direction","sea_surface_wave_significant_height","sea_surface_wind_wave_period","sea_water_electrical_conductivity","sea_water_freezing_point","sea_water_salinity","sea_water_temperature","simple_turbidity","wave_direction_spread","wind_chill","wind_from_direction","wind_speed","wind_speed_of_gust"],"time":["2014-07-25 17:00:00","2014-07-25 17:00:00"],"value":[1017.78,24.3,8.13,7.85,270.0,27.37,24.3,38.9628,-

76.447022,49.5,9.31,0.6,0.17,264.0,0.28,2.5,1.34,-0.4092,7.51,26.41,5.74,66.4,24.3,311.0,3.9,6.0],"units":["mb","C","ug\/L","mg\/L","degrees","mm\/sec","C","degrees","degrees","percent","percent","m","m","degrees","m","s","S\/m","C","PSU","C","NTU","degrees","C","degrees","m\/s","m\/s"]} }

QueryData

"CBIBS","AN","air_temperature","2014-07-01 00:00:00", "2014-07-02 00:00:00","a1655be60c484b273ffec1595f3661b2b9f15c1a"

Response:

{ "id": 1, "error": null, "result": {"measurement":"air_temperature","units":"C","values":{"time":["2014-07-01 00:10:00","2014-07-01 00:20:00","2014-07-01 00:30:00","2014-07-01 00:40:00","2014-07-01 00:50:00","2014-07-01 01:00:00","2014-07-01 01:10:00","2014-07-01 01:20:00","2014-07-01 01:30:00","2014-07-01 01:40:00","2014-07-01 01:50:00","2014-07-01 02:00:00","2014-07-01 02:10:00","2014-07-01 02:30:00","2014-07-01 02:40:00","2014-07-01 02:50:00","2014-07-01 03:00:00","2014-07-01 03:10:00","2014-07-01 03:20:00","2014-07-01 03:30:00","2014-07-01 03:40:00","2014-07-01 03:50:00","2014-07-01 04:00:00","2014-07-01 04:10:00","2014-07-01 04:20:00","2014-07-01 04:30:00","2014-07-01 04:40:00","2014-07-01 04:50:00","2014-07-01 05:00:00","2014-07-01 05:10:00","2014-07-01 05:20:00","2014-07-01 05:30:00","2014-07-01 05:40:00","2014-07-01 05:50:00","2014-07-01 06:00:00","2014-07-01 06:10:00","2014-07-01 06:20:00","2014-07-01 06:30:00","2014-07-01 06:40:00","2014-07-01 06:50:00","2014-07-01 07:00:00","2014-07-01 07:10:00","2014-07-01 07:20:00","2014-07-01 07:30:00","2014-07-01 07:40:00","2014-07-01 07:50:00","2014-07-01 08:00:00","2014-07-01 08:10:00","2014-07-01 08:20:00","2014-07-01 08:30:00","2014-07-01 08:40:00","2014-07-01 08:50:00","2014-07-01 09:00:00","2014-07-01 09:10:00","2014-07-01 09:20:00","2014-07-01 09:30:00","2014-07-01 09:40:00","2014-07-01 09:50:00","2014-07-01 10:00:00","2014-07-01 10:10:00","2014-07-01 10:20:00","2014-07-01 10:30:00","2014-07-01 10:40:00","2014-07-01 10:50:00","2014-07-01 11:00:00","2014-07-01 11:10:00","2014-07-01 11:20:00","2014-07-01 11:30:00","2014-07-01 11:40:00","2014-07-01 11:50:00","2014-07-01 12:00:00","2014-07-01 12:10:00","2014-07-01 12:20:00","2014-07-01 12:30:00","2014-07-01 12:40:00","2014-07-01 12:50:00","2014-07-01 13:00:00","2014-07-01 13:10:00","2014-07-01 13:20:00","2014-07-01 13:30:00","2014-07-01 13:40:00","2014-07-01 13:50:00","2014-07-01 14:00:00","2014-07-01 14:10:00","2014-07-01 14:20:00","2014-07-01 14:30:00","2014-07-01 14:40:00","2014-07-01 14:50:00","2014-07-01 15:00:00","2014-07-01 15:10:00","2014-07-01 15:20:00","2014-07-01 15:30:00","2014-07-01 15:40:00","2014-07-01 15:50:00","2014-07-01 16:00:00","2014-07-01 16:10:00","2014-07-01 16:20:00","2014-07-01 16:30:00","2014-07-01 16:40:00","2014-07-01 16:50:00","2014-07-01 17:00:00","2014-07-01 17:10:00","2014-07-01 17:20:00","2014-07-01 17:30:00","2014-07-01 17:40:00","2014-07-01 17:50:00","2014-07-01 18:00:00","2014-07-01 18:10:00","2014-07-01 18:20:00","2014-07-01 18:30:00","2014-07-01 18:40:00","2014-07-01 18:50:00","2014-07-01 19:00:00","2014-07-01 19:10:00","2014-07-01 19:20:00","2014-07-01 19:30:00","2014-07-01 19:40:00","2014-07-01 19:50:00","2014-07-01 20:00:00","2014-07-01 20:10:00","2014-07-01 20:20:00","2014-07-01 20:30:00","2014-07-01 20:40:00","2014-07-01 20:50:00","2014-07-01 21:00:00","2014-07-01 21:10:00","2014-07-01 21:20:00","2014-07-01 21:30:00","2014-07-01 21:40:00","2014-07-01 21:50:00","2014-07-01 22:00:00","2014-07-01 22:10:00","2014-07-01 22:20:00","2014-07-01 22:30:00","2014-07-01 22:40:00","2014-07-01 22:50:00","2014-07-01 23:00:00","2014-07-01 23:10:00","2014-07-01 23:20:00","2014-07-01 23:30:00","2014-07-01 23:40:00","2014-07-01

23:50:00"],"value":[26.6,26.5,26.4,26.4,26.4,26.3,26.3,26.3,26.3,26.3,26.3,26.3,26.3,26.3,26.3,26.3,26.2,26.3,26.3,26.4,26.3,26.2,26.1,26.0,25.9,25.9,25.9,25.8,25.7,25.7,25.6,25.6,25.5,25.5,25.4,25.4,25.3,25.3,25.3,25.2,25.2,25.2,25.2,25.2,25.2,25.1,25.1,25.2,25.2,25.2,25.2,25.1,25.0,24.9,24.9,24.9,24.8,24.8,24.8,24.8,24.7,24.7,24.8,24.9,24.9,25.0,25.0,25.1,25.2,25.2,25.3,25.3,25.4,25.4,25.4,25.4,25.5,25.5,25.5,25.6,25.6,25.7,25.8,25.9,25.9,25.9,26.0,25.9,25.9,26.0,26.1,26.1,26.2,26.3,26.4,26.6,26.7,26.6,26.7,26.9,27.0,27.2,27.3,27.5,27.6,27.7,27.7,27.7,27.8,27.8,27.9,28.0,28.1,28.1,28.1,28.0,28.1,28.1,28.1,28.1,28.1,28.2,28.1,28.0,28.0,28.0,28.0,28.0,28.1,28.1,28.1,28.0,28.0,28.0,28.1,28.1,28.1,28.1,28.0,27.9,27.7,27.6]}} }

Function

xmlrpc_cdrh.ListParameters

Payload:

<param><value><string>CBIBS</string></value></param>

<param><value><string>AN</string></value></param>

<param><value><string>a1655be60c484b273ffec1595f3661b2b9f15c1a</string></value></param>

Response:

<methodResponse> <params> <param> <value><array> <data> <value><string>wind_speed</string></value> <value><string>wind_from_direction</string></value> <value><string>wind_speed_of_gust</string></value> <value><string>air_temperature</string></value> <value><string>air_pressure</string></value> <value><string>relative_humidity</string></value> <value><string>sea_water_temperature</string></value> <value><string>sea_water_electrical_conductivity</string></value> <value><string>sea_water_salinity</string></value> <value><string>concentration_of_oxygen_in_sea_water</string></value> <value><string>concentration_of_chlorophyll_in_sea_water</string></value> <value><string>simple_turbidity</string></value> <value><string>sea_surface_mean_wave_height</string></value> <value><string>sea_surface_wave_significant_height</string></value> <value><string>sea_surface_wave_from_direction</string></value> <value><string>wave_direction_spread</string></value> <value><string>sea_surface_wind_wave_period</string></value> <value><string>latitude</string></value> <value><string>longitude</string></value> <value><string>current_average_velocity</string></value> <value><string>current_average_direction</string></value> <value><string>sea_surface_maximum_wave_height</string></value> <value><string>sea_water_freezing_point</string></value> <value><string>wind_chill</string></value> <value><string>seanettle_prob</string></value> <value><string>heat_index</string></value> </data> </array></value> </param> </params> </methodResponse>