RTDMS REST API

Version 1.0

REST API v1.0

RTDMS REST API communication requires a valid token for authentication of user. This token should be present in header of each request to the server. This token will be provided by the logicladder team.
The token should be kept securely as it is like your username and password. This provides access to your data.

1 API to upload data of a station

This API is used to upload parameters value for a station. A station can be an ETP or a stack. A station can have multiple devices installed to record different environment parameters (cod, bod, tss, pH, flow, Sox, Nox, PM etc).

1.1 EndPoint

http://182.75.69.206:8080/v1.0/industry/<industryId>/station/<stationId>/data

1.2 Method

Request should be through POST method.

1.3 Header

The request should contain a valid token in Authorization header. The token should be Base64 encoded. The header format should be:

```
POST  v1.0/industry/<industryId>/station/<stationId>/data HTTP/1.1
Host:   182.75.69.206:8080
Authorization: Basic <token>
```

Note:- Here industryId and stationId corresponding to industryId and stationId available in vendor system.

1.4 Request Json

Note:-
When device is in Calibration mode or Zero Calibration mode then vendor should capture and push one data point at every 30 sec.

The json should provide an array of data object, specific to each device installed at a station. The request json to be posted would be :

```
[
        {
        "deviceId": "xxxx",
        "params": [
        {
                "parameter": "xxxx",
                "value": xxxx,
                "unit": "xxx",
                "timestamp": xxxx,
               "flag": "U|C|M|F|Z|D"
        }
        ],
        "diagnostics": [
        {
                "diagParam": "xxxx",
                "value": xxxx,
               "timestamp": xxxx
        }
        ]
        }
]
```

Table 1.0: Single station data upload json fields.

Parameters	Data Type	Description
deviceId	String	An id that uniquely identifies a device installed at a station.

params	Array	Array of parameters for which data is to be uploaded. The Array contains data specific to the device id mentioned above, the industry id and station id present in the request.
		e.g:
		[{
		“parameter”:”cod”,
		“value”:190,
		“unit‘:”mg/l”
		“timestamp”:1441686170004
		“flag”:”U”
		},
		{

		“parameter”:”bod”, “value”:28, “unit‘:”mg/l” “timestamp”:1441686170004 “flag”:”U” }]
params: parameter	String	Name of the parameter. Name should be picked from the parameter table in the annexure (Table 5.0). If a parameter is not available in the list, please inform cpcb/logicladder.
params: value	Decimal	Value of parameter in decimal.
params: unit	String	Unit in which device is recording the parameter. Unit should be picked from the unit table in the annexure (Table 6.0). If a unit is not available please inform cpcb/logicladder.
params: timestamp	long	Epoch/Unix/Posix time in milliseconds i.e milliseconds elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970
params: flag	String	Diagnostic flags associated with device. This is subject to changes as and when different vendors flags are unified. Possible values of flag: U:usable C:calibration M:Maintenance F:Faulty Z:Zero D:Calibration Drift
diagnostics	Array	Array of diagnostic information of the device. e.g [{

		“diagParam”:“humidityAlert”, “value”: 0, “timestamp”:1441686170004 }, { “diagParam”:“deviceTemperature”, “value”: 49, “timestamp”:1441686170004 }]
diagnostics: diagParam	String	Name of diagnostic parameter.
diagnostics: value	Decimal	Value of diagnostic parameter in decimal.
diagnostics: timestamp	long	Epoch/Unix/Posix time in milliseconds i.e milliseconds elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970

1.4.1 Example JSON

```
[
        {
        "deviceId": "sn-123-d-56",
        "params": [
        {
                "parameter": "cod",
                "value": 198,
                "unit": "mg/l",
                "timestamp": 1441686170004,
                "flag": "U"
        },
        {
                "parameter": "bod",
                "value": 28,
                "unit": "mg/l",
                "timestamp": 1441686170004,
                 "flag": "U"
        }
        ],
```

```
        "diagnostics": [
        {
                "diagParam": "humidityAlert",
                "value": 0,
                 "timestamp": 1441686170004
        },
        {
                "diagParam": "devTemperature",
                "value": 49,
                 "timestamp": 1441686170004
        }
        ]
        },
        {
        "deviceId": "sn-124-d-58",
        "params": [
        {
                "parameter": "ph",
                "value": 7.8,
                "unit": “”,
                "timestamp": 1441686170004
                "flag": "U",
        },
        ]
        "diagnostics": [
        {
                "diagParam": "devTemperature",
                "value": 49,
                 "timestamp": 1441686170004
        }
        ]
        }
]
```

1.5 Response

Success: If data uploaded successfully

```
HTTP STATUS         200 OK
{
```

```
        "msg": "success",
        “status”: 1
}
```

Failure: If data upload fails

```
HTTP STATUS         other than 200
{
   "msg": "failed",
   "status": 0,
   "invalidIndustries": {
         "industries": [{
               "industryId": "abc"
         }]
   },
   "invalidStations": {
         "industries": [{
               "industryId": "abc",
               "stations": [{
                     "stationId": "xyz"
               }]
         }]
   },
   "invalidDevices": {
         "industries": [{
               "industryId": "abc",
               "stations": [{
                     "stationId": "xyz",
                     "devices": [{
                           "serialNo": "xyz"
                     }]
               }]
         }]
   },
   "invalidParameters": {
         "industries": [{
               "industryId": "abc",
               "stations": [{
                     "stationId": "xyz",
                     "devices": [{
```

```
                                "serialNo": "xyz",
                                "parameters": [{
                                        "parameter": "abc"
                                }]
                        }]
                }]
        }]
    },
    "invalidUnits": {
        "industries": [{
                "industryId": "abc",
                "stations": [{
                        "stationId": "xyz",
                        "devices": [{
                                "serialNo": "xyz",
                                "parameters": [{
                                        "parameter": "abc",
                                        "unit": "mg/L"
                                }]
                        }]
                }]
        }]
    }
}
```


For status codes please refer Annexure (Table 7.0)

2 API to upload Data of an Industry

This api is used to upload parameters value of multiple stations of an industry. A station can be an ETP or a stack. A station can have multiple devices installed to record different environment parameters (cod, bod, tss, pH, flow, Sox, Nox, PM etc).

2.1 EndPoint

http://182.75.69.206:8080/v1.0/industry/<industryId>/data

Note:- Here industryId corresponding to industryId available in vendor system.

2.2 Method

Request should be through POST method.

2.3 Header

The request header should contain a valid token in Authorization header. The token should be Base64 encoded. The header format should be:

```
POST  v1.0/industry/<industryId>/data HTTP/1.1
Host:   182.75.69.206:8080
Authorization: Basic <token>
```

2.4 Request Json

Note:-
When device is in Calibration mode or Zero Calibration mode then vendor should capture and push one data point at every 30 sec.

The request accepts an array of stations with its data. The request json to be posted would be :

```
[
        {
        "stationId": "xxxx",
        "data": [
        {
                "deviceId": "xxxx",
                "params": [
                {
                "parameter": "xxxx",
                "value": xxxx,
                "unit": "xxx",
                "timestamp": xxxx,
                "flag": "U|C|M|F|Z|D"
                }
                ],
                "diagnostics": [
                {
                 "diagParam": "xxxx",
                 "value": xxxx,
                   "timestamp": xxxx
                }
                ]
        }
        ]
```

```
        }
]
```

Table 2.0: Multiple Station data upload json fields.

Parameters	Data Type	Description
stationId	String	Id of the station - ETP or Stack.
data	Array	Data for the particular station. Each element of array corresponds to a device installed at a station. For fields within data please refer Table 1.0

2.4.1 Example Json

```
[{
   "stationId": "stack1",
   "data": [{
          "deviceId": "sn-123-d-56",
          "params": [{
                 "parameter": "cod",
                 "value": 198,
                 "unit": "mg/l",
                 "timestamp": 1441686170004,
                 "flag": "U"
          }, {
                 "parameter": "bod",
                 "value": 28,
                 "unit": "mg/l",
                 "timestamp": 1441686170004,
                 "flag": "U"
          }],
          "diagnostics": [{
                 "diagParam": "humidityAlert",
                 "value": 0,
                 "timestamp": 1441686170004
          }, {
                 "diagParam": "devTemperature",
```

```
                "value": 49,
                "timestamp": 1441686170004
        }]
    }, {
        "deviceId": "sn-124-d-58",
        "params": [{
                "parameter": "ph",
                "value": 7.8,
                "unit": " ",
                "timestamp": 1441686170004,
                "flag": "U"
        }],
        "diagnostics": [{
                "diagParam": "devTemperature",
                "value": 51,
                "timestamp": 1441686170004
        }]
    }]
}, {
    "stationId": "stack2",
    "data": [...]
}]
```

2.5 Response

Success: If data uploaded successfully

```
HTTP STATUS         200 OK
{
        "msg": "success",
        “status”: 1
}
```

Failure: If data upload fails

```
HTTP STATUS         other than 200
{
```

```
"msg": "failed",
"status": 0,
"invalidIndustries": {
  "industries": [
    {
      "industryId": "abc"
    }
  ]
},
"invalidStations": {
  "industries": [
    {
      "industryId": "abc",
      "stations": [
        {
          "stationId": "xyz"
        }
      ]
    }
  ]
},
"invalidDevices": {
  "industries": [
    {
      "industryId": "abc",
      "stations": [
        {
          "stationId": "xyz",
          "devices": [
            {
              "serialNo": "xyz"
            }
          ]
        }
      ]
    }
  ]
},
"invalidParameters": {
  "industries": [
    {
      "industryId": "abc",
```

```
      "stations": [
       {
        "stationId": "xyz",
        "devices": [
         {
          "serialNo": "xyz",
          "parameters": [
           {
            "parameter": "abc"
           }
          ]
         }
        ]
       }
      ]
     }
    ]
   },
   "invalidUnits": {
    "industries": [
     {
      "industryId": "abc",
      "stations": [
       {
        "stationId": "xyz",
        "devices": [
         {
          "serialNo": "xyz",
          "parameters": [
           {
            "parameter": "abc",
            "unit": "mg/L"
           }
          ]
         }
        ]
       }
      ]
     }
    ]
   }
}
```

For status codes please refer Annexure (Table 7.0)

3 API to Upload Correction

This api is used to submit corrections for a parameter. A station can be an ETP or a stack. A station can have multiple devices installed to record different environment parameters (cod, bod, tss, pH, flow, Sox, Nox, PM etc).

3.1 EndPoint

http://182.75.69.206:8080/v1.0/industry/<industryId>/station/<stationId>/correction

Note:- Here industryId and stationId corresponding to industryId and stationId available in vendor system.

3.2 Method

Request should be through POST method.

3.3 Header

The request should contain a valid token in Authorization header. The token should be Base64 encoded. The header format should be:

```
POST  v1.0/industry/<industryId>/station/<stationId>/data HTTP/1.1
Host:   182.75.69.206:8080
Authorization: Basic <token>
```

3.4 Request Json

The json should provide a change request of the specified station id of a given industry. The request json to be posted would be :

```
{
   "approvedbySPCB": "true/false",
   "regionalOffice": "xxxx",
   "officialName": "xxxx",
   "officialEmail": "xxxx",
```

```
  "officialContact": "xxxx",
  "approvalReason": "xxxx",
  "submittedBy": "xxxx",
  "submissionDateTime": "xxxx",
  "correctionReason": "xxxx",
  "requestID": "xxxx",
  "responseURL": "xxxx",
  "params": [
    {
      "paramName": "xxxx",
      "data": [
        {
          "dataTimestamp": 1441686170004,
          "newValue": 5.14
        },
        {
          "dataTimestamp": 1441686170004,
          "newValue": x.xxxx
        }
      ]
    },
    {
      "paramName": "xxxx",
      "data": [
        {
          "dataTimestamp": 1441686170004,
          "newValue": x.xxxx
        }
      ]
    }
  ]
}
```


Table 3.0: Correction data upload json fields.

Parameters	Data Type	Description
approvedBySPCB	Boolean	Indicator whether approved by SPCB. Valid values are “true” or “false”

regionalOffice	String	SPCB regional office name. Mandatory if *approvedBySPCB* is true
officialName	String	SPCB official name. Mandatory if *approvedBySPCB* is true.
officialEmail	String	SPCB official email address. Mandatory if *approvedBySPCB* is true.
officialContact	String	SPCB contact details of official. Mandatory if *approvedBySPCB* is true
approvalReason	String	SPCB change request approval reason. Mandatory if *approvedBySPCB* is true
submittedBy	String	Name of industrial official who submitted the CR.
submissionDateTime	String	Date of CR submission in yyyy-MM-dd HH:mm:ss format
correctionReason	String	Correction Reason
requestID	String	Unique ID of the request as maintained in your system. Your system would be updated against this ID through a callback URL.
responseURL	String	Callback URL to send back the status of request (whether approved or rejected by CPCB). Check response section for response json format
params	Array	Array of change request of each param.
params:paramName	String	Name of parameter i.e. so2, pm, no2, etc.
params:data	Array	Array of data point containing new value for param and timestamp for value.

params:data:dataTimestamp	long	Epoch/Unix/Posix time in milliseconds i.e milliseconds elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970
params:data:newValue	Decimal	New value for parameter

3.5 Response

Success: If data uploaded successfully

```
HTTP STATUS        200 OK
{
        "msg": "success",
        “status”: 1
}
```

Failure: If data upload fails

```
HTTP STATUS        other than 200
{
        "msg": "failed",
        “status”: 0
}
```

For status codes please refer Annexure (Table 7.0)

Response json Data format for responseURL:

```
{
        “requestId”:”1234”,
        “status”: “Approved/Rejected”,
        “reason”:”valid data”
}
```

Response fields details

Parameter	Description
requestId	Unique ID of the request sent as part of CR request.
status	Action taken by cpcb(Approved or Rejected)
reason	Approval or rejection comment

4 API to upload daily SMS count

This API is used to upload daily SMS count communicated to CPCB, SPCB, INDUSTRY and VENDOR.

4.1 EndPoint

http://182.75.69.206:8080/v1.0/sms/<industry_Id >

4.2 Method

Request should be through POST method.

4.3 Header

The request should contain a valid token in Authorization header. The token should be Base64 encoded. The header format should be:

```
POST  v1.0/sms/<industry_Id >  HTTP/1.1
Host:   182.75.69.206:8080
Authorization: Basic <token>
```

Note:- Here Industry_Id refers the actual Industry Id available at vendor system.

4.4 Request Json

The json should provide the details of parameter with its min and max value along with contact details. The request json to be posted would be :

```
[{
        "parameter": "xx",
        "min_limit": xxx,
        "max_limit": xxx,
        "sms_count": xxx,
        "exceedance_count": xxx,
        "unit" : "xxxxxx",
        "date": "x-xxx-xxxx",
        "contact_details": [{
                "user_type": "xxxxxxxxxx",
                "contact_name": "xxxxxxxx",
                "contact_phone": "xxxxxxxx",
               "contact_email":”xxxxxxxx”
        }, {
                "user_type": "xxxxx",
                "contact_name": "xxxxx",
                "contact_phone": "xxxxxxxxxxxx",
               "contact_email":”xxxxxxxx”
        }]
}, {
        "parameter": "xxx",
        "min_limit": xx,
        "max_limit": xx,
        "sms_count": xx,
        "exceedance_count": xxx,
        "unit": "xx",
        "date": "xx-xxx-xxxx",
        "contact_details": [{
                "user_type": "xxxxxxxxxx",
                "contact_name": "xxxxxxxx",
                "contact_phone": "xxxxxxxxxxxx",
               "contact_email":”xxxxxxxx”
        }]
}]
```

Table 4.0: Daily SMS count upload json fields.

Parameters	Data Type	Description
parameter	String	Name of the parameter and should be picked from the parameter table in the annexure (Table 5.0). If a parameter is not available in the list, please inform cpcb/logicladder. This field is mandatory.
min_limit	long	Min range of parameter (Either min or max should be available)
max_limit	long	Max range of parameter (Either min or max should be available)
sms_count	long	Number of sms communicated that count could be zero or greater than zero.
exceedance_count	long	Provide number of exceedance count found and should be greater than zero.
unit	String	Unit in which device is recording the parameter. Unit should be picked from the unit table in the annexure (Table 6.0). If a unit is not available please inform cpcb/logicladder. This field is mandatory.
date	String	Date at sms communication done with dd-MMM-yyyy format. This field is mandatory.
contact_details	Array	Array of contact details or pass empty contact details array.
contact_details:user_type	String	If contact detail is given than this field is mandatory. Possible user types are: 1. SPCB 2. CPCB 3. INDUSTRY 4. VENDOR

contact_details:contact_name	String	This field is optional or provide contact person name.
contact_details:contact_phone	String	if contact detail is given than this field is mandatory.
contact_details:contact_email	String	This is optional field or provide Email ID of contact person.

4.4.1 Example JSON

```
[{
        "parameter": "pm",
        "min_limit": 10,
        "max_limit": 100,
        "sms_count": 40,
        "exceedance_count": 10,
        "unit" : "mg/Nm3",
        "date": "24-Feb-2016",
        "contact_details": [{
                "user_type": "CPCB",
                "contact_name": "Mr.Rahul",
                "contact_phone": "8512899322",
               "contact_email":”rahul@gmail.com”
        }, {
                "user_type": "INDUSTRY",
                "contact_name": "Mr.vivek",
                "contact_phone": "9310161018",
               "contact_email":”vivek@gmail.com”
        }]
}, {
        "parameter": "ph",
        "min_limit": 0,
        "max_limit": 14,
        "sms_count": 50,
        "exceedance_count":10,
        "unit": "ph",
        "date": "24-Feb-2016",
        "contact_details": [{
                "user_type": "INDUSTRY",
                "contact_name": "Mr.Rahul",
```

```
                    "contact_phone": "8512899322",
                   "contact_email":"rahul@gmail.com"
          }]
}]
```

4.5 Response

Success: If data uploaded successfully

```
HTTP STATUS          200 OK
{
          "msg": "success",
          "status": 1
}
```

Failure: If data upload fails

```
HTTP STATUS          other than 200
{
          "msg": "failed",
          "status": 0
}
```

For status codes please refer Annexure (Table 7.0)

5 Annexure

5.1 Parameter Keys

Please use these keys when submitting data for a parameter.

Table 5.0: Parameter keys table

Parameter Name	Parameter Key
pH	ph
BOD	bod
COD	cod
TSS	tss

Flow	flow
Chromium	chromium
Ammonical Nitrogen	ammonical_nitrogen
Fluoride	fluoride
Phenol	phenol
Cyanide	cyanide
Arsenic	as
Adsorbable organic halogens	aox
Temperature for Effluent	ef_temperature
Temperature for Emission	em_temperature
Particulate Matter	pm
Dust	pm
SPM	pm
Sulfur Dioxide	so2
Sulfur Oxides	sox
Chlorine	cl
Hydrochloric acid	hcl
Ammonia	ammonia
Carbon Monoxide	co
Carbon Dioxide	co2
Hydro carbon	hc
Nitrogen Oxide	no
Nitrogen Dioxide	no2
Nitrogen Oxides	nox
Total carbon	tc
Hydrogen Sulfide	h2s
Hydrogen fluoride	hf
Opacity	opacity
Velocity	velocity
Conductivity	conductivity
Oil & Grease	oil_grease
Ammoniacal nitrogen	nh4-n
Dissolved oxygen	do
Cyanide ion	cn-
Nitrate as Nitrogen	no3-n
Arsenic	as
color	color
H2O	h2o
Total volatile organic compound	tvoc
Vinyl chloride monomer	vcm
O2	o2

Benzene	benzene
Total dissolved solids	tds
Total organic carbon	toc
Ammonium	ammonium
Flow Outlet	flow
Env Temp	temperature
Flow Volume	flow_totalizer
Phosgene(COCl2)	cocl2
Phosphates Concentration	phosphates
HCN	hcn
Flow_inlet_totalizer	flow_inlet_totalizer
Inlet FLow2	inlet flow2
PRIMARY TEMPERATURE	primary_temperature
Sec. Temp	secondary_temperature
Mercury	mercury
Flow Inlet	flow_inlet

5.2 Units

Please use these keys when submitting data for a parameter.

Table 6.0: Unit keys table

Unit	Unit Key
kg/m³	kg/m3
mg/L	mg/l
mg/Nm³	mg/Nm3
mg/m³	mg/m3
umg/m³	umg/m3
ug/m³	ug/m3
ppm	ppm
ppb	ppb
Vol percent	vol%
%	%
m/s	m/s
K	kelvin
°C	celsius
°F	fahrenheit
m³/hr	m3/hr
W/m²	w/m2
mmHg	mmhg
Degree	degree
Mtr/sec	mtr/sec

mm	mm
pH	ph
Torr	torr
LPM	lpm
Hazen	hu
Kg/Hr	kg/hr
m³/day	m3/day
l/hr	l/hr
ton/hr	ton/hr
µg/m³	ug/m3

5.3 Common Status Codes in Response JSON

Table 7.0: Status codes in json response

Status	Description
1	Success
0	Request failed because of unknown reason.
10	Wrong API Key
11	Invalid Json e.g wrong data type, field etc.
101	Invalid industry id i.e this industry id is not present in the system.
102	Invalid station id i.e station id is not present for this industry id.
103	Correction not within 7 days
104	No data present for one of the time stamp in change request
105	Required fields not present in request
106	Some other error while saving CR (Change Request)
107	'responseURL' field present in request is not valid.

108	Invalid device id i.e. device id is not configured with CPCB for given industry and station.

5.4 Throttling

- To prevent the abuse of the system, number of requests per min (cumulative across all APIs) are restricted as per the following rule -

  Max request per min by vendor = No. of devices x 1

- Data point should not be more than 40 in a single request.

Note:
If no of request exceeds the given range, below response you will get.

Status	**Description**
0	Too many Request