AERON REST API

Version 1.0

REST API v1.0

AERON REST API can be used to pull the data of Aeron's data loggers. The communication requires a valid token for authentication of the user. This token should be present in the header of each request to the server. This token will be acquired from an API given below.

1. API to acquire access token

This API is used to authenticate the user and to receive the access token.

1.1. EndPoint

https://getiotservice.aeronsystems.com:9007/v1.0/gettoken

1.2. Method

The request should be through the POST method.

1.3. Header

The header should be:

```
Host: getiotservice.aeronsystems.com:9007
Method: POST
Content Body: username=<userLoginId>&password=<userPassword>&grant_type=password
Content-Type : application/x-www-form-urlencoded
Authorization : Basic + baseEncoded(<clientName> : <clientSecret>)
Accept : application/json
```

Note: *<clientName> and <clientSecret> will be given by the Aeron team after the confirmation of PO for API.*
<userLoginId> and <userPassword> will be acquired on registration to Aeron portal. The Aeron team will register the user.

1.4. Response Json

The JSON will be provided with an access token.
The response JSON would be:
HTTP STATUS 200 OK

```
{
    "access_token": "xxxx",
    "token_type": "bearer",
    "refresh_token": "xxxx",
    "expires_in": xxxx,
    "scope": "read write"
}
```

Table 1.0: JSON data fields

Name	Data type	Description
access_token	String	Used to request data API
token_type	String	Bearer
refresh_token	String	Used to acquire access token if current token expires
expires_in	Decimal	Period of time when the access token will be valid in minutes
scope	String	Scope of the user

1.4.1. Example JSON

```
{
    "access_token": "b654b85b-6c5a-45e9-a17c-76346c6113ad",
    "token_type": "bearer",
    "refresh_token": "8d3174bd-a98d-4f25-ba7a-e212c3a726c3",
    "expires_in": 43199,
    "scope": "read write"
}
```

2. API to acquire new access token

This API is used to receive a new access token if the current access token expires.

2.1. EndPoint

https://getiotservice.aeronsystems.com:9007/v1.0/gettoken

2.2. Method

The request should be through the POST method.

2.3. Header

The header should be like this:

```
Host: getiotservice.aeronsystems.com:9007
Method: POST
Content Body: client_secret=<clientSecret>&client_id=<clientName>&refresh_token=<user's refresh token>&grant_type=<refresh_token>
Content-Type : application/x-www-form-urlencoded
Authorization : Basic + baseEncoded(<clientName> : <clientSecret>)
Accept : application/json
```

2.4. Response Json

The JSON will be provided with an access token.
The response json would be :
HTTP STATUS 200 OK

```
{
    "access_token": "xxxx",
    "token_type": "bearer",
    "refresh_token": "xxxx",
    "expires_in": xxxx,
    "scope": "read write"
}
```

2.4.1. Example JSON

```
{
    "access_token": "b654b85b-6c5a-45e9-a17c-76346c6113ad",
    "token_type": "bearer",
    "refresh_token": "8d3174bd-a98d-4f25-ba7a-e212c3a726c3",
    "expires_in": 43199,
    "scope": "read write"
}
```

3. API to get latest data of a datalogger

This API is used to fetch recent recorded parameters value of a data logger. A data logger can have multiple sensors installed to record different environmental parameters. Each data logger will be identified by its USN.

3.1. EndPoint

https://getiotservice.aeronsystems.com:9007/v1.0/getdata?usn=<device_usn>

3.2. Method

The request should be through the GET method.

3.3. Header

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

```
Host: getiotservice.aeronsystems.com:9007
Method: GET
URL Parameter: usn
Authorization: Bearer <token>
Accept: application/json
```

Note: *For <token> refer section 1 and 2.*

3.4. Response Json

The json will be provided with a data object of a specific data logger installed at a station. The response json would be :

HTTP STATUS 200 OK

```
{
      "params": [{
                  "caption": "xxxx",
                  "value": xxxx,
                  "unit": "xxx"
            },
            {
                  "caption": "xxxx",
                  "value": xxxx,
                  "unit": "xxx"
            },
            ...
      ],
```

```
      "health": {
            "charging": x,
            "battery": x,
            "supply": x,
            "network": x,
            "network_reg": x,
            "gpsfix": x
      },
      "location": {
            "latitude": x,
            "longitude": x,
            "altitude": x
      },
      "timestamp": {
            "date": "yyyy-m-d",
            "time": "h:m:s"
      }
}
```

Table 2.0: JSON data fields

Name	Data Type	Description
caption	String	Name of the parameter.
value	Decimal	Value of the parameter at mentioned timestamp.
unit	String	Unit in which device is recording the parameter.
charging	Decimal	Charging condition 0/1. (Note: Not applicable for XTM type devices)
battery	Decimal	Battery of the device on the scale of 0 to 5. (Note: Not applicable for XTM type devices)
supply	Decimal	Supply voltage to the datalogger on the scale of 0 to 14. (Note: Not applicable for XTM type of devices)
network	Decimal	Network strength of the datalogger on the scale of 0 to 5.
network_reg	Decimal	1 if a registered SIM is inserted. 0 if the registered network is not detected.
gpsfix	Decimal	1 indicates if GPS signal is received else 0

latitude	Decimal	Latitude of data logger position in decimal degrees format.
longitude	Decimal	Longitude of data logger position in a decimal degrees format.
altitude	Decimal	Altitude of data logger in meters.
date	String	Data recorded date in YYYY-M-D
time	String	Data recorded time in h:m:s

3.4.1. Example JSON

```
{
      "params": [{
                  "caption": "PM1",
                  "value": 7.23,
                  "unit": "ug/m3"
            },
            {
                  "caption": "CO",
                  "value": 0.27,
                  "unit": "mg/m3"
            },
            ...
      ],
      "health": {
            "charging": 0,
            "battery": 3,
            "supply": 9,
            "network": 4,
            "network_reg": 1,
            "gpsfix": 1
      },
      "location": {
            "latitude": 18.5789,
            "longitude": 73.7707,
            "altitude": 550
      },
      "timestamp": {
            "date": "2021-7-13",
            "time": "11:3:59"
      }
}
```

4. API to get data by date of a datalogger

This API is used to fetch parameters value of a datalogger recorded at a given date.

4.1. EndPoint

https://getiotservice.aeronsystems.com:9007/v1.0/getdatabydate?usn=<device_usn>&date=<date_value>

4.2. Method

The request should be through the GET method.

4.3. Header

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

Host: getiotservice.aeronsystems.com:9007
Method: GET
URL Parameter: usn, date (YYYY-MM-DD)
Authorization: Bearer <token>
Accept: application/json

4.4. Response Json

The json will be provided with an array of data objects of a specific data logger installed at a station.
The response json would be :
HTTP STATUS 200 OK

```
[{
      "params": [{
                  "caption": "xxxx",
                  "value": xxxx,
                  "unit": "xxx"
            },
            {
                  "caption": "xxxx",
                  "value": xxxx,
                  "unit": "xxx"
            },
            ...
      ],
      "health": {
```

```
            "charging": x,
            "battery": x,
            "supply": x,
            "network": x,
            "network_reg": x,
            "gpsfix": x
      },
      "location": {
            "latitude": x,
            "longitude": x,
            "altitude": x
      },
      "timestamp": {
            "date": "yyyy-m-d",
            "time": "h:m:s"
      }
},
{
…
}]
```

4.4.1. Example JSON

```
[{
      "params": [{
                  "caption": "PM1",
                  "value": 7.23,
                  "unit": "ug/m3"
            },
            {
                  "caption": "CO",
                  "value": 0.27,
                  "unit": "mg/m3"
            },
            ...
      ],
      "health": {
            "charging": 0,
            "battery": 3,
            "supply": 9,
            "network": 4,
            "network_reg": 1,
            "gpsfix": 1
      },
```

```
      "location": {
            "latitude": 18.5789,
            "longitude": 73.7707,
            "altitude": 550
      },
      "timestamp": {
            "date": "2021-7-13",
            "time": "11:3:59"
      }
},
{
      "params": [{
                  "caption": "PM1",
                  "value": 10.23,
                  "unit": "ug/m3"
            },
            {
                  "caption": "CO",
                  "value": 0.25,
                  "unit": "mg/m3"
            },
            ...
      ],
      "health": {
            "charging": 1,
            "battery": 2,
            "supply": 9,
            "network": 3,
            "network_reg": 1,
            "gpsfix": 1
      },
      "location": {
            "latitude": 18.5789,
            "longitude": 73.7707,
            "altitude": 550
      },
      "timestamp": {
            "date": "2021-7-13",
            "time": "11:4:0"
      }
},
...
]
```


Note: *These APIs are applicable for different types of data loggers.*

1. What is the connectivity status of each of the eight existing weather stations? are they online and providing correct readings to their clouds?	All existing weather stations are in a good condition and operating properly
2. Could you please provide screen shoots and API documentation reference for the dashboard of the existing Aeron weather stations?	Attached the API documentation reference of Aeron weather stations
3. For the new 11 weather stations, is it required as part of the installation to provide a 10 m x10 m fence?	No, it is not required to provide a 10x10 m fence
4. Is the sim card monthly bill covered by NARC or the contractor and how many years should the offer cover?	The contractor is responsible for the station's operation and functionality during the entire project period, furthermore, the contractor is obliged to operate the system for one calendar year from the date of handing over as stated in the tender document (3.7 Bid Conditions). Noting that the contractor shall exclude the 5 Aeron Weather stations monthly/annual fees.
It is described in the requirement that the system (logger) should be capable to send out alarms via SMS and email. Is it acceptable that the logger provide SMS alert functionality and the email alert is performed through the cloud solution platform?	The contractor is obliged to fulfil the provided requirements specified in the tender document, as stated under section 3.4 Wherever such modifications are suggested, in his technical proposal the future contractor/supplier shall clearly bring out the benefits that may accrue by way of these modifications of the specific parameters.

We assume that if an "All -in One" Sensor solution supports the required accuracy, such a combined sensor set-up is accepted for this project .Please confirm or correct our assumption?	It is recommended to provide the separate sensors solution for operation and maintenance purposes.
Compared to the previous tender there is now a new software requirement in Section 3.5 to "produce reports" a. What type of reports are required here? b. Must the system be set-up by the vendor to produce these reports?	General reporting requirements considering the necessary calculations as specified in the tender document under section 3.5. The software solution shall be capable to produce reports based on the client requirements and need
Regarding the rain gauge specifications, it is required to measure 1000 mm of rainfall per hour (precipitation intensity). This specification appears to be highly exaggerated based on rainfall data in the Kingdom. Would it be possible to reduce this requirement to 700 mm per hour?	The contractor shall restrict to the provided specification.
بالإشارة إلى العطاء المذكور أعلاه ، فإننا نود أن نطلب من حضراتكم لطفاً التكرم بتمديد موعد تسليم العروض لمدة 3 أسابيع من تاريخ التسليم المقرر(مع الأخذ بعين الاعتبار أن فترة التمديد المطلوبة يتخللها عطلة عيد الأضحى المبارك) ، وذلك ليتسنى لنا تحضير عرضنا بشكل متكامل وبأفضل صورة ممكنة .	موعد تسليم العطاء كما هو في الاعلان ولا يوجد اي تغيير.