API Data response example

uRADMonitor
Global Environmental Monitoring Network
www.uradmonitor.com

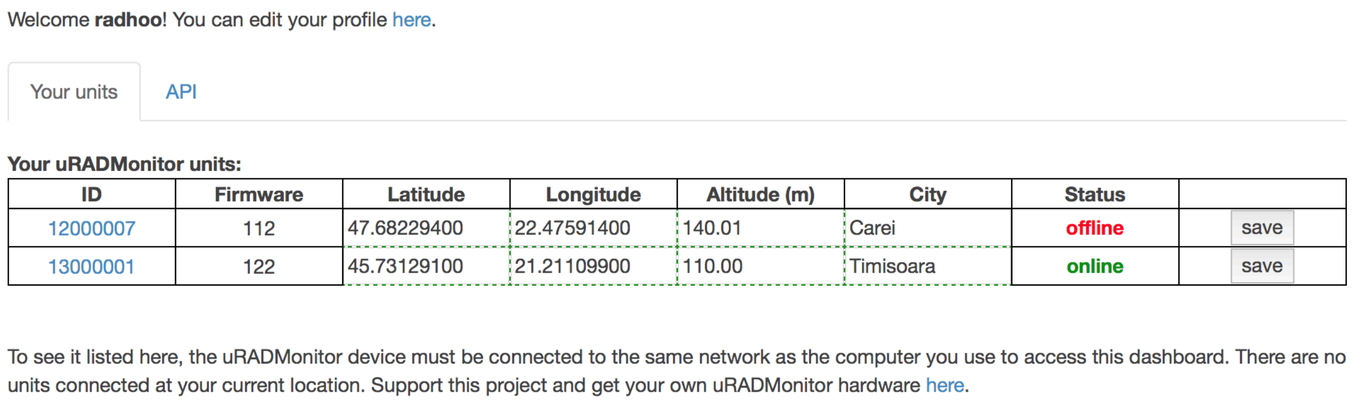
document v 1.01

About uRADMonitor

uRADMonitor is a global network of environmental monitors based on IoT topology. It is composed of hardware products of proprietary design that form a presence of several hundreds units worldwide. It addresses end users for those interested in monitoring their home environment, B2B including cities, offices or production spaces. The (big) data collected is a valuable asset for institutes interested in environmental and health research.
Environmental big data opens new perspectives for understanding the means of improving life quality. Technology is put to use constantly monitoring parameters that can directly impact our health. Better than ever we now have the necessarily capabilities to deploy hardware and use advanced connectivity options to have real time readings from the various sensors. Early warnings and notification systems are already possible, serving to offer the shortest reaction time in case of any incidents.

uRADMonitor Dashboard

To access the uRADMonitor Data you can use the uRADMonitor API presented on the Dashboard. Please go to www.uradmonitor.com/dashboard and create a user account to access the data.
The Dashboard shows the list attached to the current user account:



Welcome **radhoo**! You can edit your profile here.

Your uRADMonitor units:

ID	Firmware	Latitude	Longitude	Altitude (m)	City	Status	
12000007	112	47.68229400	22.47591400	140.01	Carei	offline	save
13000001	122	45.73129100	21.21109900	110.00	Timisoara	online	save

To see it listed here, the uRADMonitor device must be connected to the same network as the computer you use to access this dashboard. There are no units connected at your current location. Support this project and get your own uRADMonitor hardware here.

Figure 1 - List of units attached to the user account

uRADMonitor Data response example

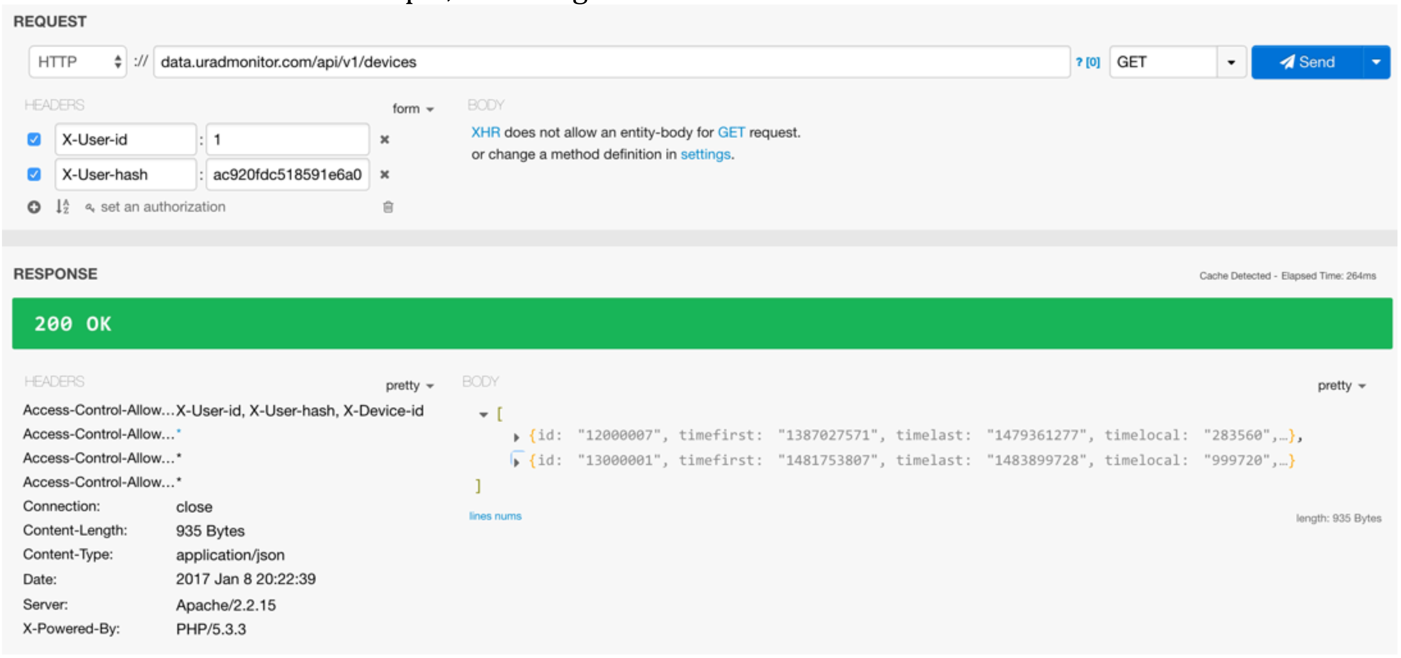
The first api call returns the list of devices attached to the user account:
`http://data.uradmonitor.com/api/v1/devices`

The response if JSON formatted.
In case of error, you will receive a message detailing the error:
`{"error":"Authentification failed"}`
If the call is successful, the response is a JSON containing the list of units attached to the current user account, including several useful details:

id = unit ID in uRADMonitor network; "timefirst"=unix timestamp with the first time the unit connected to the network; "timelast"=unix timestamp with the last transmission received from the unit; "timelocal"=unit local time in seconds from last restart;"latitude","longitude","altitude","speed"=GPS information for the unit;"city"=unit city location; "country"=unit country code;"versionsw"=firmware version;"vershionhw"=hardware version;"status"=1 if online;"mobile"=1 if it is a mobile unit;"detector"=geiger tube type for units equipped with radiation detectors;"factor"=cpm to equivalent dose linear conversion factor;"avg_temperature"=temperature average on the last 24 hours;"avg_cpm"=radiation cpm last 24hours average;"avg_voltage"=geiger tube voltage last 24 hours average;

```
[
   {
      "id":"12000007",
      "timefirst":"1387027571",
      "timelast":"1479361277",
      "timelocal":"283560",
      "latitude":"47.68229400",
      "longitude":"22.47591400",
      "altitude":140.01,
      "speed":0,
      "city":"Carei",
      "country":"RO",
      "versionsw":"112",
      "versionhw":"108",
      "status":null,
      "mobile":null,
      "detector":"SI29BG",
      "factor":0.01,
      "avg_temperature":null,
      "avg_cpm":null,
      "avg_voltage":null,
      "avg_duty":null
   },
   {
      "id":"13000001",
      "timefirst":"1481753807",
      "timelast":"1483899728",
      "timelocal":"999720",
      "latitude":"45.73129100",
      "longitude":"21.21109900",
      "altitude":110,
      "speed":0,
      "city":"Timisoara",
      "country":"RO",
      "versionsw":"122",
      "versionhw":"105",
      "status":"1",
      "mobile":null,
      "detector":"SBM19",
      "factor":0.0015,
      "avg_temperature":"-9.70",
      "avg_pressure":"101401",
      "avg_humidity":"36.67",
      "avg_luminosity":"307",
      "avg_voc":"16",
      "min_voc":"5",
      "max_voc":"264",
      "avg_co2":"0",
      "avg_ch2o":"0.00",
      "avg_pm25":"3",
      "avg_battery":"0.00",
      "avg_cpm":"77.78",
      "avg_voltage":"380.19",
      "avg_duty":"353.73"
   }
]
```

Here is a HTTP GET call example, including the authorization header variables:



data.uradmonitor.com/api/v1/devices

X-User-id	1
X-User-hash	ac920fdc518591e6a0

XHR does not allow an entity-body for GET request.
or change a method definition in settings.

```
Access-Control-Allow...X-User-id, X-User-hash, X-Device-id
Access-Control-Allow...*
Access-Control-Allow...*
Access-Control-Allow...*
Connection:          close
Content-Length:      935 Bytes
Content-Type:        application/json
Date:                2017 Jan 8 20:22:39
Server:              Apache/2.2.15
X-Powered-By:        PHP/5.3.3
```

```
[
  {id:  "12000007", timefirst:  "1387027571", timelast:  "1479361277", timelocal:  "283560",…},
  {id:  "13000001", timefirst:  "1481753807", timelast:  "1483899728", timelocal:  "999720",…}
]
```

Figure 2 - HTTP GET Call example

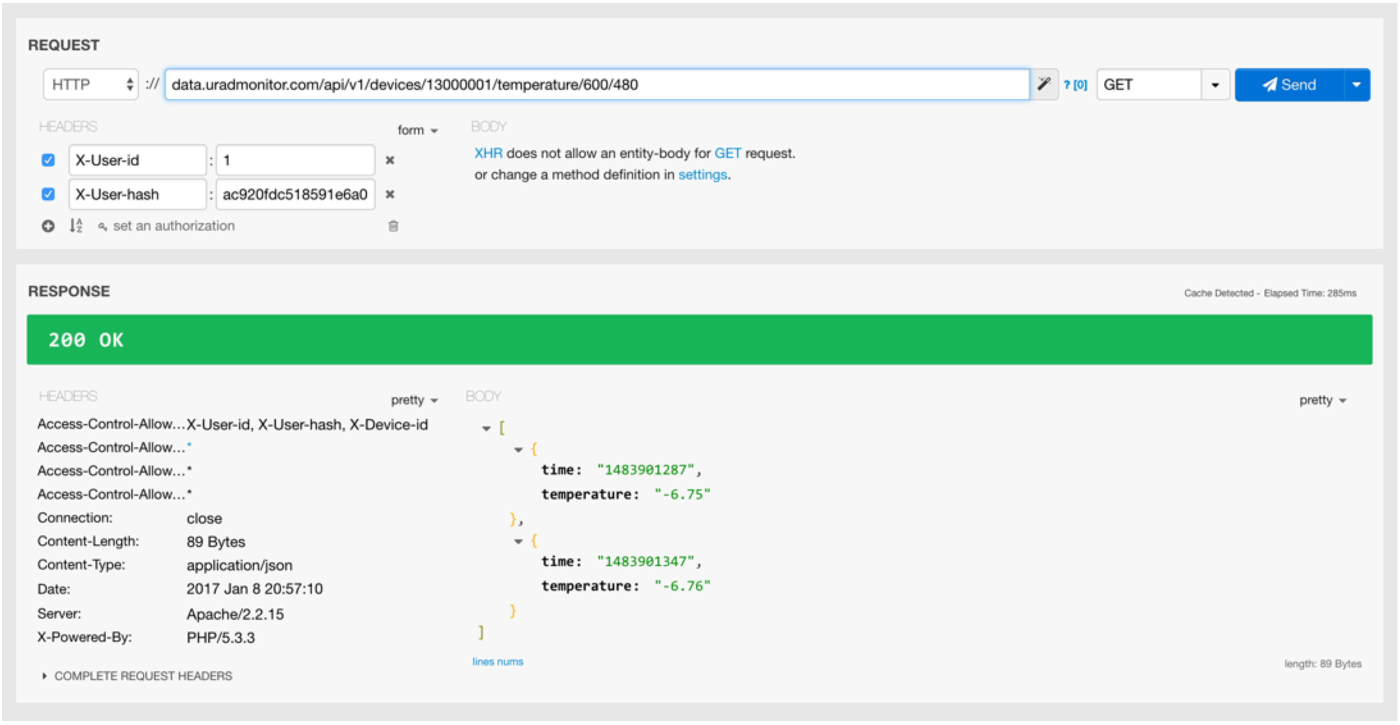
The other two API methods are to be called in the same manner:
`http://data.uradmonitor.com/api/v1/devices/13000001`

Returns the list of sensors supported, as a JSON array, including the names and the units of measure:
```
{"temperature": ["Temperature","°C"],"cpm":
["Radiation","cpm"],"voltage": ["Voltage","V"],"duty": ["Duty
cycle","‰"],"all": ["All",""]}
```

The number of sensors supported varies from one hardware type to another. Having the sensor name, you can call the third API method, to access the various measurements:
`http://data.uradmonitor.com/api/v1/devices/13000001/temperature/600/480`

As you can see in this example, the API call has several parameters:
`http://data.uradmonitor.com/api/v1/devices/[ID]/[sensor]/[startinterval]/[stopinterval]`

"ID" is the unit ID you want to access data from; "sensor" is the name of the sensor, as returned by the second API method - you can also use the special keyword "all" to access data from all sensors installed on the unit; "startinterval" is the the number of seconds from the moment of the present to get data from; "stopinterval" is optional and represents the number of seconds from the moment of present to get data to. If "stopinterval" is not specified, the moment of present is used as the stop point.
If there is no data for the query specified, you will receive an empty JSON array.
For the example above, we receive two temperature measurements, because we specified an interval of 120 seconds and the unit resolution was 1 minute:



data.uradmonitor.com/api/v1/devices/13000001/temperature/600/480

HEADERS

X-User-id	1
X-User-hash	ac920fdc518591e6a0

BODY
XHR does not allow an entity-body for GET request.
or change a method definition in settings.

Cache Detected - Elapsed Time: 285ms

200 OK

HEADERS

```
Access-Control-Allow...X-User-id, X-User-hash, X-Device-id
Access-Control-Allow...*
Access-Control-Allow...*
Access-Control-Allow...*
Connection:        close
Content-Length:    89 Bytes
Content-Type:      application/json
Date:              2017 Jan 8 20:57:10
Server:            Apache/2.2.15
X-Powered-By:      PHP/5.3.3
```

BODY

```
[
    {
        time:  "1483901287",
        temperature:  "-6.75"
    },
    {
        time:  "1483901347",
        temperature:  "-6.76"
    }
]
```

length: 89 Bytes

Figure 3 - Data access example

The maximum interval for the "startinterval" value is 24 hours of data. For more information on how to read the data, please see the dashboard section on http://www.uradmonitor.com/dashboard, and the code examples provided there.