**API Specification Doc**

**( *Sensor App* )**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 |  |  |  |

Index

[1. Authorization](#_9e3nlm4ei2ob)

[2. Get all sensors](#_ewd6g0923qx8)

[Request](#_kd6tjl5w5wjo)

[Response](#_8b7ijrpuamb8)

[3. Get current value](#_z935xnymao4m)

[Request](#_38pypefkb82)

[Response](#_47awg09ehquu)

[Conventions](#_byd7iz3wxtwk)

Methods

## 1. Authorization

In order to authenticate API requests, a valid API token should be included as HTTP header.

This API token has

**key**: **auth-token** and

**value: 8e4c46fe-5e1d-4382-b7fc-19541f7bf3b0 (string)**

All further API calls must have this key and value in header!

## 2. Get all sensors

* Returns additional information about all sensors in the application i.e. tag, description, measure type.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | api/sensor/all |

|  |  |
| --- | --- |
| **Params** | **Values** |
| No parameters needed | ------ |

## 

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| Success  (200) | **Response will be a list containing sensor items. Each item has the following structure:**  {  "sensorId": id\_of\_the\_provided\_sensor, (type **Guid**)  "tag": name\_of\_the\_provided\_sensor, (type **string**)  "description": description\_about\_behaviour\_of\_the\_provided\_sensor, (type **string**)  "minPollingIntervalInSeconds": minimum\_seconds\_for\_polling\_data\_from\_sensor, (type **int**)  "measureType": type\_of\_returning\_value\_from\_sensor, (type **string**)  }    **An example response is:-**  [  {  "sensorId": "f1796a28-642e-401f-8129-fd7465417061",  "tag": "TemperatureSensor1",  "description":"This sensor will return values between 15 and 28"  "minPollingIntervalInSeconds": 40,  "measureType": "°C"  },  {  "sensorId": "81a2e1b1-ea5d-4356-8266-b6b42471653e",  "tag": "TemperatureSensor2",  "description":"This sensor will return values between 6 and 18",  "minPollingIntervalInSeconds": 30,  "measureType": "°C"  },  {  "sensorId":"61ff0614-64fd-4842-9a05-0b1541d2cc61",  "tag": "HumiditySensor2",  "description":"This sensor will return values between 10 and 90",  "minPollingIntervalInSeconds": 50,  "measureType": "%"  },  {  "sensorId": "7a3b1db5-959d-46ce-82b6-517773327427",  "tag": "OccupancySensor2",  "description":"This sensor will return true or false value",  "minPollingIntervalInSeconds": 80,  "measureType":"(true/false)"  } ] |
| Error  (500)  Error  (400)  Error  (401) | "Unhandled exception in SensorData.Api. Message: {mеssage\_of\_the\_thrown\_exception}" |
| "Invalid data provided by the client SensorData.Api. Message: {mеssage\_of\_the\_thrown\_exception}" |
| "Invalid authentication token provided. Value: {value\_of\_the\_authentication\_token}" |

## 3. Get current value

* Get information about the value returned from the sensor in real time.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | api/sensor/ |

|  |  |
| --- | --- |
| **Params** | **Values** |
| sensorId | Guid |

**sensorId**

The sensorId is unique identifier for each sensor. It was given in the previous request.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| Success | **Response will be an object containing information about the sensor’s**  **value in real time and measure type of returned value.**    {  "timeStamp":date\_and\_time\_of\_returned\_value (type **DateTime?**),  "value":value\_from\_sensor (type **string**),  "valueType": measure\_type\_of\_value (type **string**)  }  **An example response is:-**  {  "timeStampt":"2017-10-18T10:19:32.1274223+03:00",  "value":"19.6",  "valueType":"°C"  } |
| Error  (500)  Error  (400)  Error  (401) | "Unhandled exception in SensorData.Api. Message: {mеssage\_of\_the\_thrown\_exception}" |
| "Invalid data provided by the client SensorData.Api. Message: {mеssage\_of\_the\_thrown\_exception}" |
| "Invalid authentication token provided. Value: {value\_of\_the\_authentication\_token}" |

Glossary

## Conventions

* All the possible responses are listed under ‘Responses’ for each method. Only one of them is issued per request server.
* All request parameters are mandatory unless explicitly marked as [optional]
* The type of values accepted for a *request* parameter are shown in the values column.