# Weather Station Service API

POST /users

Register a new user to the system

*Request description*

|  |  |  |
| --- | --- | --- |
| username | string | Publicly visible username |
| name | string | First and family name |
| dateOfBirth | string | Date of birth, RFC3339 format |
| address | string | User street address |
| city | string | User home city |
| country | string | User home country |
| email | string | Valid email address for the user |

*Request example*

{

"username": "example\_user",

"name": "John Doe",

"dateOfBirth": "1990-05-20",

"address": "Measurement Street 567",

"city": "London",

"country": "uk",

"email": "john.doe@demo.com"

}

Response

*Status codes*

|  |  |
| --- | --- |
| 201 | User created successfully |
| 400 | Missing or incorrect data in the request |

*Response data structure description*

|  |  |  |
| --- | --- | --- |
| id | number | The id assigned to the created user |

Response body example

{

"id": 45673265

}

GET /users

Get all the users.

Response

*Status Codes*

|  |  |
| --- | --- |
| 200 | Users displayed successfully |

*Response data structure description*

|  |  |  |
| --- | --- | --- |
| user | object | User Object containing userId and username |
| users | array | An array containing all user objects |

Response body example

{

"users":[

{

"username": "John Doe",

"id": 45673265

},

{

"username": "Nick",

"id": 14335613

}

]

}

POST /users/id/sensors

Register a new sensor

*Request description*

|  |  |  |
| --- | --- | --- |
| deviceType | string | The type of the device |
| description | string | The description of the device |
| latitude | string | The latitude of the device location  ("degrees minutes seconds") |
| longitude | string | The longitude of the device location |
| sensorType | string | The type of the sensor. Valid values:  "temperature", "humidity", "rainfall", "wind", "cloudCoverage" |

*Request example*

{

"deviceType": "arduino uno",

"description": "my device",

"latitude": "44 38 12.32",

"longitude": "-134 12 27.11",

"sensorType": "temperature"

}

Response

*Status codes*

|  |  |
| --- | --- |
| 201 | Added created successfully |
| 400 | Missing or incorrect data in the request |

*Response data structure description*

|  |  |  |
| --- | --- | --- |
| sensorId | string | The id assigned to the added sensor |

*Response body example*

{

"sensorId": "342"

}

POST /users/id/sensors/sensorId/measurement

Add a new measurement for the sensor with sensorId.

*Request description*

|  |  |  |
| --- | --- | --- |
| measurement | string | The measurement data |

*Request example*

{

"measurement": "32C"

}

Response

*Status codes*

|  |  |
| --- | --- |
| 201 | Measurement added successfully |
| 400 | Missing or incorrect data in the request |

*Response data structure description*

|  |  |  |
| --- | --- | --- |
| measurement | string | The measured data |
| date | string | The date of the request.  UTC Format |
| username | string | The name of the user |

*Response body example*

{

"measurement": "32C"

"date": "2020-01-21 15:30"

"username": "John Cena"

}

GET /measurements

Return all the measurements.

Response

*Status codes*

|  |  |
| --- | --- |
| 200 | Measurements displayed successfully |

*Response data structure description*

|  |  |  |
| --- | --- | --- |
| measurementData | object | Object contains username, sensorType, latitude, longitude, date, measurement |
| measurements | array | An array containing all user objects |

Response body example

{

"measurements":[

{

"username": "John Cena",

"sensorType": "temperature",

"latitude": "44 38 12.32",

"longitude": "-134 12 27.11",

"date": "2020-01-21 05:48",

"measurement": "32C"

},

{

"username": "Donald Trump",

"sensorType": "humidity",

"latitude": "12 22 12.62",

"longitude": "-124 44 67.13",

"date": "2020-01-21 20:22",

"measurement": "74%"

}

]

}

GET /measurements/history

Return all the measurements between the given dates.

*Request description*

|  |  |  |
| --- | --- | --- |
| date1 | number | Beginning date, in milliseconds |
| date2 | number | Ending date, in milliseconds |

*Request example*

{

"date1": "1579334415000"

"date2": "1579593615000"

}

Response

*Status codes*

|  |  |
| --- | --- |
| 200 | Measurements displayed successfully |
| 400 | Missing or incorrect data in the request |

*Response data structure description*

|  |  |  |
| --- | --- | --- |
| measurementData | object | Object contains username, sensorType, latitude, longitude, date, measurement |
| measurements | array | An array containing all user objects |

Response body example

{

"users":[

{

"username": "John Cena",

"sensorType": "temperature",

"latitude": "44 38 12.32",

"longitude": "-134 12 27.11",

"date": "2020-01-21",

"measurement": "32C"

},

]

}