

# Sensor Integration Guide

The Configuration Administration API provides endpoints for creating, retrieving, and managing configurations.

The API uses role based access control so any interaction will be limited by the requesting user's role(s).



Please read the [General Integration Guide](#) to learn the general concepts and common data structures used throughout the Argus API.

## Introduction

Sensors are entities (analog or digital) that transmit data to Argus' Agents for eventual processing and analysis. They have a specific Type and exist at one of a Customer's defined Locations or a Global Location.

The [Swagger API documentation](#) is always up to date and lets you try out any request with your user session or an API key.

The roles most relevant to Sensor interactions are

- SENSOR-MANAGER
- SENSOR-VIEW
- SENSOR-STATUS

Users can interact with the Sensor service's REST API in many different ways. This guide will explore and explain them.

- [Introduction](#)
- [Sensors](#)
  - [Create a Sensor](#)
  - [Get a Sensor](#)
  - [Update a Sensor](#)
  - [Enable a Sensor](#)
  - [Bulk Enable Sensors](#)
  - [Disable a Sensor](#)
  - [Bulk Disable Sensors](#)
  - [Enable Monitoring for a Sensor](#)
  - [Disable Monitoring for a Sensor](#)
  - [Bulk Enable Monitoring for Sensors](#)
  - [Bulk Disable Monitoring for Sensors](#)
  - [Schedule Downtime for a Sensor](#)
  - [Bulk Schedule Downtime for Sensors](#)
  - [Cancel Downtime for a Sensor](#)
  - [Bulk Cancel Downtime for Sensors](#)
  - [Delete a Sensor](#)
- [Location](#)
  - [Create a Location](#)
  - [Get a Location](#)
  - [Update a Location](#)
  - [Delete a Location](#)
- [Sensor Interface](#)
  - [Create a Sensor Interface](#)
  - [Get a Sensor's Interface](#)
  - [Update a Sensor's Interface](#)
  - [Delete a Sensor's Interface](#)
- [Sensor Type/Application](#)
  - [Create a Sensor Type/Application](#)
  - [Get a Sensor Type/Application](#)
  - [Update a Sensor Type/Application](#)

## Sensors

### Create a Sensor

To create a Sensor POST an AddSensorRequest to the base URL:

```
curl -X POST "https://api.mnemonic.no/sensors/v1/sensor" -H "Argus-API-Key: my/api/key" -H "accept: application/json" -H "Content-Type: application/json" -d "{
  \"sslTerminating\": true,
  \"location\": \"1\",
  \"customer\": \"1\",
  \"application\": \"1\",
  \"clusterSensorID\": 0,
  \"expectedUpdateTime\": 0,
  \"scheduledDowntimeFromTime\": 0,
  \"scheduledDowntimeUntilTime\": 0,
  \"information\": \"Optional information\",
  \"hostname\": \"01-example-01.hostname.no\",
  \"additionalData\": \"Optional additional data\",
  \"active\": false,
  \"monitored\": false,
  \"monitoredOnlyDaytime\": false,
  \"initialTuning\": false,
  \"overrideDefaultUpdateTime\": false,
  \"sharedCustomerData\": false,
  \"ipAddress\": \"10.0.0.1\"
}"
```

For such a request to be successful

- The user must have the **addSensor** permission for the Customer they're creating to Sensor for, a role that grants it, or belong to a group with said permission
- The Customer for whom we want to create the Sensor must exist
- The Location where the Sensor will be created must exist
- The Application (Sensor Type) for the Sensor must be valid
- At least one of Hostname or IP must be present in the payload
- If clustering, the targetted Sensor must exist, belong to the same Customer, and be the same Sensor Type



#### IP or Hostname

**NB:** One of or both IP or Hostname must be provided when creating a Sensor. IPs and hostnames must be unique per Customer.

## Get a Sensor

To GET a Sensor, append the sensorID to the base URL:

```
curl -X GET "https://api.mnemonic.no/sensors/v1/sensor/{sensorID}" -H "accept: application/json" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **viewSensors** permission for the Sensor's customer, a role that grants it, or belong to a group with said permission
- The Sensor must exist

## Update a Sensor

To update a Sensor, PUT an UpdateSensorRequest via PUT to the base URL with the sensor's ID appended:

```
curl -X PUT "https://api.mnemonic.no/sensors/v1/sensor/{sensorID}" -H "Argus-API-Key: my/api/key" -H
"accept: application/json" -H "Content-Type: application/json"
-d "{
  \"sslTerminating\": true,
  \"location\": \"1\",
  \"application\": \"1\",
  \"clusterSensorID\": 0,
  \"expectedUpdateTime\": 0,
  \"information\": \"Optional updated information\",
  \"hostname\": \"01-example-01.new.hostname.no\",
  \"additionalData\": \"Optional updated additional data\",
  \"initialTuning\": false,
  \"overrideDefaultUpdateTime\": false,
  \"sharedCustomerData\": false,
  \"ipAddress\": \"10.0.0.2\"
}"
```

For the request to be successful

- The user must have the **updateSensor** permission for the Sensor's customer, a role that grants it, or belong to a group with said permission
- The Sensor must exist
- When updating the Sensor's Location
  - The Location must exist
  - The Location must belong to the same Customer as the Sensor
- When updating the IP or hostname
  - The proposed IP or hostname must be unique to the Customer
  - One of either values must always be present and not empty e.g. if erasing a Sensor's hostname then the IP must either already be set or will be set by the same update request
- When updating the cluster Sensor
  - The proposed cluster Sensor must exist
  - The user must have permission to access the cluster Sensor
  - A Sensor cannot be clustered with itself
  - The proposed cluster Sensor must not already be bound to another Sensor
  - The proposed cluster Sensor must belong to the same Customer
  - The proposed cluster Sensor must be of the same Sensor Type
- When updating the Sensor Type
  - The new Sensor Type must be valid and exist
- When overriding the default update time
  - The proposed update time must be valid
  - The **OVERRIDE\_UPDATE\_TIME** flag must either be already set or is set in the request
- Only these fields can be set to blank
  - additional data
  - sensor information
  - IP/hostname IF the other value is present



#### Dedicated Sensor Updates

**NB:** Changing a Sensor's status, monitoring behaviour, and scheduling of downtime is handled by dedicated endpoints and request. Not via the update endpoint.

## Enable a Sensor

To activate a Sensor simply append /enable to the Sensor's path:

```
curl -X PUT "https://api.mnemonic.no/sensors/v1/sensor/{sensorID}/enable" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **enableSensor** permission for the Sensor's customer, a role that grants it, or belong to a group with said permission
- The Sensor must exist
- If the Sensor is already enabled the service will return an error

## Bulk Enable Sensors

To bulk activate Sensors PUT an EnableSensorBulkRequest to the bulk enable endpoint

```
curl -X PUT "https://api.mnemonic.no/sensors/v1/sensor/bulk/enable" -H "accept: application/json" -H "Content-Type: application/json" -H "Argus-API-Key: my/api/key"
-d "{
  \"sensorIDs\": [
    1,
    2,
    3
  ]
}"
```

For the request to be successful

- The user must have the **enableSensor** permission for the Sensors' customer(s), a role that grants it, or belong to a group with said permission
- The Sensors must exist

## Disable a Sensor

To deactivate a Sensor simply append /disable to the Sensor's path:

```
curl -X PUT "https://api.mnemonic.no/sensors/v1/sensor/{sensorID}/disable" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **disableSensor** permission for the Sensor's customer, a role that grants it, or belong to a group with said permission
- The Sensor must exist
- If the Sensor is already disabled the service will return an error

## Bulk Disable Sensors

To bulk deactivate Sensors PUT a DisableSensorBulkRequest to the bulk disable endpoint

```
curl -X PUT "https://api.mnemonic.no/sensors/v1/sensor/bulk/disable" -H "accept: application/json" -H "Content-Type: application/json" -H "Argus-API-Key: my/api/key"
-d "{
  \"sensorIDs\": [
    1,
    2,
    3
  ]
}"
```

For the request to be successful

- The user must have the **disableSensor** permission for the Sensors' customer(s), a role that grants it, or belong to a group with said permission
- The Sensors must exist

## Enable Monitoring for a Sensor

To enable monitoring for a Sensor PUT a request to the Sensor's enable monitoring endpoint. The user can indicate whether to only monitor during daytime.

```
curl -X PUT "https://api.mnemonic.no/sensors/v1/sensor/{sensorID}/monitoring/enable?daytime={true|false}" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **enableSensorMonitoring** permission for the Sensor's customer, a role that grants it, or belong to a group with said permission
- The Sensor must exist

## Disable Monitoring for a Sensor

To disable monitoring for a Sensor PUT a request to the Sensor's disable monitoring endpoint

```
curl -X PUT "https://api.mnemonic.no/sensors/v1/sensor/{sensorID}/monitoring/disable" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **disableSensorMonitoring** permission for the Sensor's customer, a role that grants it, or belong to a group with said permission
- The Sensor must exist

## Bulk Enable Monitoring for Sensors

To bulk disable monitoring for Sensors PUT a request to the bulk enable monitoring endpoint. Similar to the single Sensor enable monitoring endpoint the user can indicate, for all Sensor in the request not per Sensor, whether to only monitor during daytime.

```
curl -X PUT "https://api.mnemonic.no/sensors/v1/sensor/bulk/monitoring/enable?daytime={true|false}" -H "accept: application/json" -H "Content-Type: application/json" -H "Argus-API-Key: my/api/key" -d "{
  \"sensorIDs\": [
    1,
    2,
    3
  ]
}"
```

For the request to be successful

- The user must have the **enableSensorMonitoring** permission for the Sensors' customer(s), a role that grants it, or belong to a group with said permission
- The Sensor(s) must exist

## Bulk Disable Monitoring for Sensors

To bulk disable monitoring for Sensors PUT a request to the bulk disable monitoring endpoint

```
curl -X PUT "https://api.mnemonic.no/sensors/v1/sensor/bulk/monitoring/disable" -H "accept: application/json" -H "Content-Type: application/json" -H "Argus-API-Key: my/api/key" -d "{
  \"sensorIDs\": [
    1,
    2,
    3
  ]
}"
```

For the request to be successful

- The user must have the **disableSensorMonitoring** permission for the Sensor's customer, a role that grants it, or belong to a group with said permission
- The Sensor(s) must exist

## Schedule Downtime for a Sensor

To schedule downtime for a Sensor POST a SensorScheduleDowntimeRequest to the Sensor's downtime endpoint

```
curl -X POST "https://api.mnemonic.no/sensors/v1/sensor/{sensorID}/downtime" -H "accept: application/json" -H "Content-Type: application/json" -H "Argus-API-Key: my/api/key" -d "{
  \"fromTime\": 1,
  \"toTime\": 2,
  \"keepDowntimeOnUpdate\": true,
  \"comment\": \"optional comment\"
}"
```

For the request to be successful

- The user must have the **scheduleSensorDowntime** permission for the Sensor's customer, a role that grants it, or belong to a group with said permission
- The Sensor must exist
- The downtime range must be valid i.e. downtime cannot start or end in the past and must end after it starts

## Bulk Schedule Downtime for Sensors

To bulk schedule downtime for Sensors POST a `SensorScheduleDowntimeBulkRequest` to the bulk schedule downtime endpoint

```
curl -X POST "https://api.mnemonic.no/sensors/v1/sensor/bulk/downtime" -H "accept: application/json" -H "Content-Type: application/json" -H "Argus-API-Key: my/api/key"
-d "{
  \"fromTime\": 1,
  \"toTime\": 2,
  \"keepDowntimeOnUpdate\": true,
  \"comment\": \"optional comment\",
  \"sensorIDs\": [
    1,
    2,
    3
  ]
}"
```

For a request to be successful

- The user must have the **scheduleSensorDowntime** permission for the Sensor's customer, a role that grants it, or belong to a group with said permission
- The Sensor(s) must exist
- The downtime range must be valid i.e. downtime cannot start or end in the past and must end after it starts

## Cancel Downtime for a Sensor

To cancel downtime for a Sensor send a DELETE request to the Sensor's downtime endpoint

```
curl -X DELETE "https://api.mnemonic.no/sensors/v1/sensor/{sensorID}/downtime" -H "Argus-API-Key: my/api/key"
```

For a request to be successful

- The user must have the **cancelSensorDowntime** permission for the Sensor's customer, a role that grants it, or belong to a group with said permission
- The Sensor must exist

## Bulk Cancel Downtime for Sensors

To bulk cancel downtime for Sensors send a list of the SensorIDs to the bulk cancel downtime endpoint

```
curl -X DELETE "https://api.mnemonic.no/sensors/v1/sensor/bulk/downtime?sensorID=1&sensorID=2&sensorID=3" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **cancelSensorDowntime** permission for the Sensors' customer(s), a role that grants it, or belong to a group with said permission
- The Sensor(s) must exist

## Delete a Sensor

To delete a Sensor submit a DELETE request to the Sensor path

```
curl -X DELETE "https://api.mnemonic.no/sensors/v1/sensor/{sensorID}" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **deleteSensor** permission for the Sensor's customer, a role that grants it, or belong to a group with said permission
- The Sensor must exist

## Location

To be able to create and thereafter interact with a Sensor we first need to ensure there a Location to which we can attach the Sensor. A Location represents a physical or digital location in a given time and network zone belonging to a specific Customer.

### Create a Location

To create a Location POST a AddLocationRequest to the base URL

```
curl -X POST "https://api.mnemonic.no/sensors/v1/location" -H "accept: application/json" -H "Content-Type: application/json" -H "Argus-API-Key: my/api/key" -d "{
  \"name\": \"location1\",
  \"shortName\": \"l1\",
  \"networkZone\": \"INTERNAL\",
  \"customerID\": 1,
  \"timeZoneDescription\": \"Europe/Oslo\"
}"
```

For the request to be successful

- The user must have the **addLocation** or **addGlobalLocation** permission depending on the Location's customer, a role that grants it, or belong to a group with said permission
- The Customer for whom we want to create the Sensor must exist
- The AddLocationRequest must be valid
  - The short name must be unique
  - The timezone must exist

### Get a Location

To get a Location append the Location's shortname or ID to the base URL

```
curl -X GET "https://api.mnemonic.no/sensors/v1/location/{idOrShortname}" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **viewLocations** or **viewGlobalLocations** permission depending on the Location's customer, a role that grants it, or belong to a group with said permission
- The Location must exist

### Update a Location

To update a Location PUT an UpdateLocationRequest to the Location's path

```
curl -X PUT "https://api.mnemonic.no/sensors/v1/location/{idOrShortname}" -H "accept: application/json" -H "Content-Type: application/json" -H "Argus-API-Key: my/api/key" -d "{
  \"name\": \"newlocation1\",
  \"shortName\": \"newl1\",
  \"networkZone\": \"DMZ\",
  \"timeZoneDescription\": \"Europe/London\"
}"
```

For the request to be successful

- The user must have the **updateLocation** or **updateGlobalLocation** permission depending on the Location's customer, a role that grants it, or belong to a group with said permission
- The Location must exist
- The UpdateLocationRequest must be valid
  - timezone
  - name
  - shortname

### Delete a Location

To delete a Location submit a DELETE request to the Location's path

```
curl -X DELETE "https://api.mnemonic.no/sensors/v1/location/{idOrShortname}" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **removeLocation** or **removeGlobalLocation** permissions depending on the Location's customer, a role that grants it or belong to a group with said permission
- The Location must exist

## Sensor Interface

### Create a Sensor Interface

To create a Sensor Interface for a Sensor POST an AddSensorInterfaceRequest to the Sensor's base interface path

```
curl -X POST "https://api.mnemonic.no/sensors/v1/sensor/{idOrShortname}/interface" -H "accept: application/json" -H "Content-Type: application/json" -H "Argus-API-Key: my/api/key" -d "{
  \"interfaceName\": \"string\",
  \"location\": \"locationIdOrShortname\",
  \"active\": false,
  \"customer\": \"customerIdOrShortname\"
}"
```

For the request to be successful

- The user must have the **addSensorInterface** and **updateSensor** permissions for the Sensor's customer, a role that grants them, or belong to a group with said permissions
- The Sensor for which we want to create the Interface must exist
- The Location of the Interface must exist
- The Interface name must be unique for the targetted Sensor i.e. different Sensors can have Interfaces with the same name but a Sensor cannot have multiple Interfaces with the same name
- The Customer must exist and if the shared customer data flag is not set the interface's customer must be the same as the sensor's



#### Sensor Customer

**NB:** Once the Sensor Interface is created, it is not possible to modify its Customer.

## Get a Sensor's Interfaces

To list a Sensor's interfaces GET the Sensor's base interface path

```
curl -X GET "https://api.mnemonic.no/sensors/v1/sensor/{idOrShortname}/interface" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **viewSensorInterface** permission for the Sensor's customer, a role that grants them, or belong to a group with said permissions
- The Sensor must exist

### Get a Sensor's Interface

To get a specific Sensor Interface append the Interface's ID or name to the Sensor's base interface path

```
curl -X GET "https://api.mnemonic.no/sensors/v1/sensor/{idOrShortname}/interface/{idOrName}" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **viewSensorInterface** permission for the Sensor's customer, a role that grants them, or belong to a group with said permissions
- The Sensor must exist
- The Interface must exist

## Update a Sensor's Interface

To update a specific Sensor's Interface PUT an UpdateSensorInterfaceRequest to that Sensor Interface's path

```
curl -X PUT "https://api.mnemonic.no/sensors/v1/sensor/{idOrShortname}/interface/{idOrName}" -H "accept: application/json" -H "Content-Type: application/json" -H "Argus-API-Key: my/api/key"
-d "{
  \"interfaceName\": \"newName\",
  \"location\": \"newIdOrShortname\",
  \"active\": true
}"
```

For the request to be successful

- The user must have the **viewSensors**, **updateSensor**, **viewSensorInterface**, and **updateSensorInterface** permissions for the Sensor's customer, a role that grants them, or belong to a group with said permissions
- The Sensor must exist
- The Interface must exist
- The request must be valid
  - Any new name must be unique to the Sensor
  - Any new Location must exist and belong to the same Customer as the Interface's Sensor

## Delete a Sensor's Interface

To delete a specific Sensor's Interface submit a DELETE request to that Sensor Interface's path

```
curl -X DELETE "https://api.mnemonic.no/sensors/v1/sensor/{idOrShortname}/interface/{idOrName}" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **updateSensor** and **removeSensorInterface** permissions for the Sensor's customer, a role that grants them, or belong to a group with said permissions
- The Sensor must exist
- The Interface must exist

## Sensor Type/Application

A Sensor Type/Application describes the software category/manufacturer to which the Sensor belongs i.e:

- |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• Snort (none)</li> <li>• SiteProtector Proventia 3.2 (none)</li> <li>• Checkpoint FW-1 (log)</li> <li>• Generic (log)</li> <li>• SiteProtector Proventia 4.3 (none)</li> <li>• SiteProtector Proventia X (none)</li> <li>• SiteProtector Proventia 4.6 (none)</li> <li>• Microsoft Windows DNS (log)</li> <li>• Netscape Enterprise Server (log)</li> <li>• Microsoft Windows DHCP (log)</li> <li>• Estmon (none)</li> <li>• Trend Micro Interscan VirusWall (log)</li> <li>• EnVision (none)</li> <li>• Internet Information Server (log)</li> <li>• Apache HTTP Server (log)</li> <li>• Microsoft Exchange Server (log)</li> <li>• FrontPage Server Extensions (log)</li> <li>• Cisco ISE (log)</li> </ul> | <ul style="list-style-type: none"> <li>• McAfee Endpoint Security (log)</li> <li>• Symantec Endpoint Protection (log)</li> <li>• ISS Server Sensor (none)</li> <li>• BlueCoat ProxySG (log)</li> <li>• Finjan (log)</li> <li>• IronPort (log)</li> <li>• ArcSight Logger (log)</li> <li>• ArcSight Connector (log)</li> <li>• Cisco (log)</li> <li>• ArcSight ESM (log)</li> <li>• test (infrastructure)</li> <li>• Microsoft Windows (log)</li> <li>• SourceFire (none)</li> <li>• FireEye (log)</li> <li>• Suricata (none)</li> <li>• Microsoft Azure (log)</li> <li>• Bind DNS (log)</li> <li>• MSN messenger (log)</li> </ul> | <ul style="list-style-type: none"> <li>• Websphere Application Server (log)</li> <li>• Internet Explorer (log)</li> <li>• F5 big-ip (log)</li> <li>• OpenSSH (log)</li> <li>• Sendmail (log)</li> <li>• Postfix (log)</li> <li>• Trend (log)</li> <li>• --Ikke relevant-- (none)</li> <li>• --Andre-- (none)</li> <li>• Tipping Point UnityOne (none)</li> <li>• WebSense (log)</li> <li>• Citrix Netscaler (log)</li> <li>• Microsoft IIS (log)</li> <li>• Passive DNS Sensor (none)</li> <li>• Argus Sample Carver (none)</li> <li>• SiteProtector Proventia XGS (none)</li> <li>• Agent Canary (none)</li> <li>• Fortigate Firewall (log)</li> </ul> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Create a Sensor Type/Application

To create a Sensor Type/Application POST a SensorTypeAddRequest to the base url

```
curl -X POST "https://api.mnemonic.no/sensors/v1/type" -H "accept: application/json" -H "Content-Type: application/json" -H "Argus-API-Key: my/api/key" -d "{
  \"shortName\": \"st1\",
  \"name\": \"sensorType1\",
  \"url\": \"www.sensortype.no\",
  \"expectedSensorUpdateTime\": 1,
  \"department\": \"none\"
}"
```

For the request to be successful

- The user must have the **registerApplication** permission, a role that grants it, or belong to a group with said permission
- The short name must be unique



### Deleting Sensor Types/Applications

**NB:** There currently is no endpoint for deleting Sensor Types/Application so be careful when creating a new one

## Get all Sensor Types/Applications

To list all Sensor Type/Application submit a GET request to the base URL

```
curl -X GET "https://api.mnemonic.no/sensors/type" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **viewApplications** permission, a role that grants it, or belong to a group with said permission

## Get a Sensor Type/Application

To GET a Sensor Type/Application append its ID or shortname to the base URL

```
curl -X GET "https://api.mnemonic.no/sensors/type/{idOrShortname}/interface/{idOrName}" -H "Argus-API-Key: my/api/key"
```

For the request to be successful

- The user must have the **viewApplications** permission, a role that grants it, or belong to a group with said permission
- The short name must be unique

## Update a Sensor Type/Application

To update a Sensor Type/Application PUT a `SensorTypeUpdateRequest` to its path

```
curl -X PUT "https://api.mnemonic.no/sensors/v1/type/{idOrShortname}" -H "accept: application/json" -H "Content-Type: application/json" -H "Argus-API-Key: my/api/key" -d "{
  \"shortName\": \"newst1\",
  \"name\": \"newSensorType1\",
  \"url\": \"www.newsensortype.no\",
  \"expectedSensorUpdateTime\": 2,
  \"department\": \"none\"
}"
```

For the request to be successful

- The user must have the **updateApplication** permission, a role that grants it, or belong to a group with said permission
- The Type/Application must exist
- The request must be valid
  - Any new short name must be unique