

Disclaimer

Legal disclaimer

This guide and its contents (the "Guide") are provided by NXGEN. All rights to the Guide and to the intellectual property embodied in it — including, without limitation, trademarks, trade names, logos and similar marks appearing in or on it — are protected by law. All right, title and interest in the Guide and in any related intellectual property rights belong to and remain with NXGEN and its licensors.

The Guide is provided free of charge, "as is" and for informational purposes only, without warranty of any kind. It is neither intended nor suitable to establish any legal relationship between the reader and NXGEN or its affiliates, including, without limitation, any obligation of NXGEN or its affiliates toward the reader.

The obligations of NXGEN and its affiliates in respect of NXGEN products or services purchased by a customer are governed exclusively by the terms of the agreement under which those products or services were purchased.

For clarification

The reader bears the entire risk as to the use, results and performance of the Guide. To the fullest extent permitted by applicable law, NXGEN disclaims and excludes all warranties, whether statutory, express or implied — including, without limitation, implied warranties of merchantability, fitness for a particular purpose, title and non-infringement of third-party rights — as well as any liability or warranty arising from suggestions, specifications or samples related to the Guide.

Sensor

Updated 4 September 2026

Operations on sensor resources. 9 endpoints, each relative to `https://api.nxgen.cloud/api/v1` and authenticated as described under [Authentication](#).

POST /sensor/

Create new Sensor

Request body

Add a Sensor

```
{
  "payload": [
    {
      "id": "string",
      "siteEventProviderId": "string",
      "customerEventProviderId": "string",
      "customerId": "string",
      "serviceProvidersEventProviderId": "string",
      "associatedDeviceName": "string",
      "associatedDeviceId": "string",
      "catalogName": "string",
      "commercialName": "string (required)",
      "deviceType": "string",
      "sensorType": "string",
      "description": "string",
      "macAddress": "string",
      "modelNo": "string",
      "hardwareVersion": "string",
      "firmwareVersion": "string",
      "manufacturer": "string",
      "jsonSchema": "object",
      "photos": [
        "string"
      ],
      "specification": "string",
      "geoCoordinates": {
        "latitude": "string",
        "longitude": "string",
        "plusCode": "string",
        "what3Words": "string"
      },
      "siteParent": "string",
      "siteId": "string",
    }
  ]
}
```

```
"assetStatus": "string",
"tags": [
  "string"
],
"eventTags": [
  "string"
],
"channelId": "number",
"type": "string",
"syncStatus": "boolean",
"isPtzCamera": "boolean",
"isThermalCamera": "boolean"
}
]
```

Responses

StatusDescriptionBody200Success204No Content400Invalid input401Unauthorized403Forbidden

POST /sensor/filter

Gets sensor list with filtering, sorting and pagination. It allows hierarchical level searching. It can be filtered using specified hierarchy or it uses the logged in user hierarchy. This gives the option to get additional entity fields in the response. Also provides configuration meta data about entity.

Request body

Sensor filtering and sorting

```
{
  "filters": [
    {
      "columnName": "string (required)",
      "type": "distinctList | date | number | StringConditionFilter (required)",
      "filter": "object (required)"
    }
  ],
  "sortOrder": [
    {
      "columnName": "string",
      "isAscending": "boolean"
    }
  ],
  "primaryColumns": [
    "string"
  ],
  "hierarchyLevel": "string",
  "hierarchyId": "string",
}
```

```
"nPerPage": "number",
"pageNumber": "number",
"isConfigDetailRequired": "boolean",
"getConfig": "boolean",
"getChildOrParent": "boolean"
}
```

Responses

StatusDescriptionBody200Successobject204No Content400Invalid input401Unauthorized403-

Forbidden

Example 200 response:

```
{
  "config": {
    "columns": [
      {
        "visibility": true,
        "columnName": "commercialName",
        "DisplayName": "Sensor",
        "isFilterable": true,
        "filterType": "stringCondition",
        "sticky": true,
        "isRightBorder": true,
        "isMandatory": true,
        "isNotDraggable": true
      }
    ],
    "defaultNPerPage": 50,
    "defaultFilterCondition": [],
    "defaultSortingOrder": [
      {
        "columnName": "dateTime",
        "isAscending": false
      }
    ],
    "maxVisibleColumnCount": 6,
    "exportLimitNPerPage": 500
  },
  "dataResult": [
    {
      "commercialName": "Camera 01",
      "assetStatus": "active",
      "enableAudio": "Disabled",
      "manufacturer": "HSION",
      "uniqueId": "b89e4e809ce547f01234567eba36f825",
      "dateTime": 1648453119079,
      "associatedDeviceId": "e5560be4e809ce547f012547f76547f01234",
      "associatedDeviceName": "Device"
    },
    {
      "commercialName": "Camera 03",
      "assetStatus": "active",

```

```
"enableAudio": "Disabled",
"manufacturer": "HSION",
"uniqueId": "p05e4e107ce547f7654321eba36f790",
"dateTime": 1648453119079,
"associatedDeviceId": "561eba3591c690ed03c4547f0123456345c4",
"associatedDeviceName": "Device"
},
],
"totalRecordsCount": 3552,
"filteredBy": [],
"sortedBy": [
{
"columnName": "dateTime",
"isAscending": false
}
],
"dispatchedAll": false,
"isDefaultProjectionMutated": false,
"isDefaultFilterMutated": false,
"isDefaultSortingMutated": false
}
```

PUT /sensor/isolate?sensorId={sensorId}&enable={enable}

Update isolation for sensor

Parameters

NameInTypeRequiredDescriptionsensorIdquerystringyesenablequerybooleanyesisolateStartTimequery-numbernodurationquerynumberno

Responses

StatusDescriptionBody200Success204No Content401Unauthorized403Forbidden404Not Found

POST /sensor/listBySiteIds

Get sensors by site id list

Request body

Get sensors

```
{
"siteIds": [
"string"
```

```
]
}
```

Responses

StatusDescriptionBody200Successobject204No Content400Invalid input401Unauthorized403-

Forbidden

Example 200 response:

```
{
  "data": [
    {
      "key": "7f01237864567eba89736f825",
      "label": "sensor",
      "deviceId": "ce546437f01234567e6f825",
      "deviceName": "device",
      "siteName": "site",
      "customerName": "customer"
    }
  ]
}
```

POST /sensor/search

Search sensors

Request body

Search sensors

```
{
  "skip": "number",
  "limit": "number",
  "siteName": "string",
  "siteId": "string (required)",
  "sensorName": "string",
  "isAll": "boolean",
  "searchTag": [
    "string"
  ]
}
```

Responses

StatusDescriptionBody200Successobject400Invalid input401Unauthorized403Forbidden404Not Found

Example 200 response:

```
{
  "data": [
    {
      "sensorId": "89d42e2953b8b5d662b3006bfdd444e",
      "name": "Camera 03",
      "isActive": true,
      "type": "camera",
      "mapInfo": {
        "latitude": "13.035775322405794",
        "longitude": "80.17819431590344",
        "plusCode": "25PH+87 Chennai, Tamil Nadu, India",
        "what3Words": "climbing.rested.alongside",
        "zone": ""
      }
    }
  ],
  "isSuccess": true
}
```

GET /sensor/{id}

Get by sensor id

Parameters

NameInTypeRequiredDescriptionidpathstringyes

Responses

StatusDescriptionBody200Success204No Content401Unauthorized403Forbidden404Not Found

PUT /sensor/{id}

Update an existing Sensor

Parameters

NameInTypeRequiredDescriptionidpathstringyes

Request body

```
{
  "payload": {
    "id": "string",
    "siteEventProviderId": "string",
    "customerEventProviderId": "string",
    "customerId": "string",
    "imageFile": "string",
    "serviceProvidersEventProviderId": "string",
    "associatedDeviceName": "string",
    "associatedDeviceId": "string",
    "catalogName": "string",
    "commercialName": "string (required)",
    "deviceType": "string",
    "sensorType": "string",
    "description": "string",
    "macAddress": "string",
    "modelNo": "string",
    "hardwareVersion": "string",
    "firmwareVersion": "string",
    "manufacturer": "string",
    "jsonSchema": "object",
    "jsonSchemaResult": "object",
    "eventTypes": [
      "object"
    ],
    "actionSupported": [
      "object"
    ],
    "photos": [
      "object"
    ],
    "photoSizeInPixel": [
      "object"
    ],
    "specification": "string",
    "assetId": "string",
    "prerequisite": [
      "object"
    ],
    "geoCoordinates": {
      "latitude": "string",
      "longitude": "string",
      "plusCode": "string",
      "what3Words": "string",
      "zone": "string"
    },
    "siteParent": "string",
    "siteId": "string",
    "assetStatus": "string",
    "tags": [
      "string"
    ]
  }
}
```

```
,
"eventTags": [
  "string"
],
"isInArmState": "boolean",
"uniqueId": "string",
"channelId": "number",
"syncStatus": "boolean",
"roiCoordinates": [
  [
    "number"
  ]
],
"isIsolate": "boolean",
"isolatedTime": "string",
"isolateStartTime": "string",
"isPtzCamera": "boolean",
"isThermalCamera": "boolean",
"enableAudio": "boolean",
"timerMaskCoordinates": [
  [
    "number"
  ]
],
"analyticsROI": [
  "object"
],
"analytics": "object",
"analyticsSPDefined": [
  "object"
]
}
}
```

Responses

StatusDescriptionBody200Success204No Content400Invalid input401Unauthorized403Forbidden

DELETE /sensor/{id}

Delete a sensor by id

Parameters

NameInTypeRequiredDescriptionidpathstringyesforceDeletequerystringno force delete sensor

Responses

StatusDescriptionBody200Success204No Content400Invalid input401Unauthorized403Forbidden

GET /sensor/{id}/streamingEndpoints

Get streaming endpoints

Parameters

NameInTypeRequiredDescriptionidpathstringyesSensor idstartTimequerynumbernoEpoch time in millisecondsendTimequerynumbernoEpoch time in millisecondsexpiryquerynumbernoIn seconds

Responses

StatusDescriptionBody200Success204No Content401Unauthorized403Forbidden404Not Found