

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.

Audit

Updated 4 September 2026

Operations on audit resources. 5 endpoints, each relative to `https://api.nxgen.cloud/api/v1` and authenticated as described under [Authentication](#).

POST /log/audit

Add new audit record. In general all CRUD APIs captures the user actions in audit log when the API is successfully executed. This AUDIT POST API is used to record user interaction which is not possible to record through other API calls.

Request body

Audit filtering and sorting

```
{
  "action": "string",
  "category": "string",
  "mqttPush": "boolean",
  "payload": {
    "sourceEntityLevel": "sensor | device | site | account",
    "sourceEntityId": "string"
  },
  "deltaPayload": "string",
  "source": "string",
  "subCategory": "string",
  "userAction": "string",
  "actionStatus": "string",
  "actionTime": "number",
  "workflowId": "string",
  "workflowName": "string",
  "userName": "string"
}
```

Responses

StatusDescriptionBody200Successobject204No Content400Invalid input401Unauthorized403-

Forbidden

Example 200 response:

```
{
  "data": "Audit added successfully"
}
```

GET /log/audit/eventlist/workflowid/{id}

Get list of events associated with a workflow.

Parameters

NameInTypeRequiredDescriptionidpathstringyesWork flow id

Responses

StatusDescriptionBody200Successobject400Invalid input403Forbidden

Example 200 response:

```
{
  "data": {
    "errorCode": 0,
    "errorMessage": "if errorCode above is a negative number then this errorMessage field will give more details.",
    "message": [
      "62cba820173001197cf10420",
      "62cbac84a520010f074a4403"
    ]
  }
}
```

Example 400 response:

```
{
  "errorCode": "ERR001",
  "errorDesc": "Invalid Request"
}
```

POST /log/audit/filter

Gets audit logs 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

Audit 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": "Audit",
"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": [
{
"action": "Search or Tree node click",
"actionStatus": "Success",
"category": "Video Activity Search",
"sourceEntityName": "8428 Moll Kaarst Lackierhalle",
"subCategory": "Search",
"timestamp": 1656405046000,
"userAction": "Event search result has been fetched",
"userEmail": "admin@tenant.de",
"deltaPayload": "Import Report"
}
],
"totalRecordsCount": 3552,
"filteredBy": [],
"sortedBy": [
{
"columnName": "dateTime",
"isAscending": false
}
],
"dispatchedAll": false,
"isDefaultProjectionMutated": false,
"isDefaultFilterMutated": false,
"isDefaultSortingMutated": false
}
```

GET /log/audit/report/workflowid/{id}/timezone/{tzone}

A PDF file with workflow details.

Parameters

NameInTypeRequiredDescriptionidpathstringyesWork flow idtzonepathstringnoTime Zone of client application.

Responses

StatusDescriptionBody200Success400Invalid input403Forbidden

POST /log/eventlog

Get list of eventlogs for a particular entity.

Request body

Get event logs.

```
{
  "entityId": "string",
  "startTime": "number",
  "endTime": "number",
  "page": "number"
}
```

Responses

StatusDescriptionBody200Successobject400Invalid input403Forbidden404Not Found

Example 200 response:

```
{
  "data": {
    "errorCode": 0,
    "message": {
      "eventLogs": [
        {
          "workFlowId": "8a81bc4c820160182151cd2e86160",
          "type": "WORKFLOW",
          "timestamp": 1658211586783,
          "userName": "Christian John",
          "status": "CLOSED",
          "workflowOutcome": "alarm_false",
          "message": "Verbindungsausfall V1.1",
          "workflowOutcomeName": "Falsealarm"
        }
      ]
    }
  }
}
```

```
}  
],  
"nextPage": 1  
}  
}  
}
```

Example 400 response:

```
{  
  "errorCode": "ERR001",  
  "errorDesc": "Invalid Request"  
}
```

Example 404 response:

```
{  
  "errorCode": "404",  
  "errorDesc": "Not Found"  
}
```