

Migration of Naming REST API

by lars.johansson@ess.eu, ICS Software

Table of Contents

Migration of Naming REST API.....1

 Introduction.....1

 Getting started.....1

 Migration of Naming REST API.....2

 Endpoints.....2

 DeviceNames endpoints migrated to Names endpoints.....2

 Healthcheck endpoint migrated to Healtcheck endpoint.....4

 History endpoints migrated to Names endpoint and Structures endpoint.....4

 Parts endpoints migrated to Structures endpoints.....5

 Models.....6

 DeviceNameElement → NameElement.....6

 HistoryElement → NameElement.....7

 HistoryElement → StructureElement.....8

 PartElement → StructureElement.....9

 Examples.....10

 Retrieval of content for a name entry.....10

 Retrieval of content for a structure entry.....12

Introduction

This document will give guide to migration of Naming REST API, from old to new.

This document is not a description of Naming REST API, neither old nor new.

It is assumed that reader has some knowledge of Naming REST API.

Example content is given in json.

Getting started

In various places are framed boxes with information that is noteworthy and may help understanding

Note!

Text that is noteworthy, summary, help & more

Migration of Naming REST API

Naming REST API – old

Parameters are case sensitive.
Search endpoints allow regular expressions.

Json and Xml may be used.

Naming REST API – new

Regular expressions are not supported for searches. Regex-like behavior is available with _ underscore, 0 or 1 occurrences of any character, % percent, any number of any character.

Exact match may be achived through no use of _ and % characters.

Json is used.

Below are tables with old and new endpoints, and old and new models (content), and how endpoints and models are mapped. Focus is only to describe migration from this point of view.

Endpoints

DeviceNames endpoints migrated to Names endpoints

List of DeviceNameElement corresponds to List of NameElement in ResponsePageNameElements.

DeviceNameElement corresponds to NameElement.

| HTTP method | Path & Query string | Parameter P – path Q – query | Description | Return | Old → New | HTTP method | Path & Query string | Parameter P – path Q – query | Description | Return |
|-------------|---|------------------------------------|---|---------------------------|-----------------|-------------|---------------------|------------------------------------|---|--------------------------|
| GET | /deviceNames | | Lists all device names Returns a list of (all) device names. | List of DeviceNameElement | | GET | api/v1/names | | Find valid names (search) Find valid names (search). Return paged array of name elements. | ResponsePageNameElements |
| GET | /deviceNames/devicetype/search/{devicetype} | P: devicetype | Finds all device names by device type search Note device type (search, case sensitive, regex). Returns a list of device names. | List of DeviceNameElement | | GET | api/v1/names | similar Q: devicestructure | Find valid names (search) Find valid names (search). Return paged array of name elements. | ResponsePageNameElements |

| | | | | | | | | | | |
|-----|---|--------------------|---|---------------------------|--|-----|---------------------|-------------------------------|--|--------------------------|
| GET | /deviceNames/devicetype/{devicetype} | P: devicetype | Finds all device names by device type Note device type (exact match, case sensitive). Returns a list of device names. | List of DeviceNameElement | | GET | api/v1/names | similar Q: devicestructure | Find valid names (search) Find valid names (search). Return paged array of name elements. | ResponsePageNameElements |
| GET | /deviceNames/discipline/search/{discipline} | P: discipline | Finds all device names by discipline search Note discipline (search, case sensitive, regex). Returns a list of device names. | List of DeviceNameElement | | GET | api/v1/names | similar Q: devicestructure | Find valid names (search) Find valid names (search). Return paged array of name elements. | ResponsePageNameElements |
| GET | /deviceNames/discipline/{discipline} | P: discipline | Finds all device names by discipline Note discipline (exact match, case sensitive). Returns a list of device names. | List of DeviceNameElement | | GET | api/v1/names | similar Q: devicestructure | Find valid names (search) Find valid names (search). Return paged array of name elements. | ResponsePageNameElements |
| GET | /deviceNames/search/{deviceName} | P: deviceName | Finds all device names by device name search Note device name (search, case sensitive, regex). Returns a list of device names. | List of DeviceNameElement | | GET | api/v1/names | Q: name | Find valid names (search) Find valid names (search). Return paged array of name elements. | ResponsePageNameElements |
| GET | /deviceNames/subsystem/search/{subsystem} | P: subsystem | Finds all device names by subsystem search Note subsystem (search, case sensitive, regex). Returns a list of device names. | List of DeviceNameElement | | GET | api/v1/names | similar Q: systemstructure | Find valid names (search) Find valid names (search). Return paged array of name elements. | ResponsePageNameElements |
| GET | /deviceNames/subsystem/{subsystem} | P: subsystem | Finds all devices by subsystem Note subsystem (exact match, case sensitive). Returns a list of device names. | List of DeviceNameElement | | GET | api/v1/names | similar Q: systemstructure | Find valid names (search) Find valid names (search). Return paged array of name elements. | ResponsePageNameElements |
| GET | /deviceNames/system/search/{system} | P: system | Finds all devices by system search Note system (search, case sensitive, regex). Returns a list of device names. | List of DeviceNameElement | | GET | api/v1/names | similar Q: systemstructure | Find valid names (search) Find valid names (search). Return paged array of name elements. | ResponsePageNameElements |
| GET | /deviceNames/system/{system} | P: system | Finds all devices by system Note system (exact match, case sensitive). Returns a list of device names. | List of DeviceNameElement | | GET | api/v1/names | similar Q: systemstructure | Find valid names (search) Find valid names (search). Return paged array of name elements. | ResponsePageNameElements |
| GET | /deviceNames/{uuid} | P: uuid P: name | Finds device name by uuid or name Note uuid or name (exact match, case sensitive). Returns a device name (most recent). | DeviceNameElement | | GET | api/v1/names/{name} | Q: name Q: uuid | Find valid names by name or uuid (search) Find valid names by name or uuid (search). Return paged array of name elements. | ResponsePageNameElements |

Healthcheck endpoint migrated to Healtcheck endpoint

It works in the same way.

| HTTP method | Path & Query string | Parameter P – path Q – query | Description | Return | Old → New | HTTP method | Path & Query string | Parameter P – path Q – query | Description | Return |
|-------------|---------------------|------------------------------------|---|--------|-----------------|-------------|---------------------|------------------------------------|--|--------|
| GET | /healthcheck | | Performs healthcheck of Naming application Note HTTP response code. Returns server timestamp. | | | GET | /healthcheck | | perform healthcheck for Naming application Perform healthcheck for Naming application in general and healthcheck endpoint in particular. To be used mainly for checking HTTP response code, in particular HTTP STATUS OK - 200. | |

History endpoints migrated to Names endpoint and Structures endpoint

List of DeviceNameElement corresponds to List of NameElement in ResponsePageNameElements.

List of PartElement corresponds to List of StructureElement in ResponsePageStructureElements.

DeviceNameElement corresponds to NameElement.

PartElement corresponds to StructureElement.

| HTTP method | Path & Query string | Parameter P – path Q – query | Description | Return | Old → New | HTTP method | Path & Query string | Parameter P – path Q – query | Description | Return |
|-------------|----------------------------------|------------------------------------|---|------------------------|-----------------|-------------|-----------------------------------|------------------------------------|---|-------------------------------|
| GET | /history/deviceNames/uuid/{uuid} | P: uuid | Finds history for ESS name by uuid Note uuid (exact match). Returns a list of entries for history of ESS name. | List of HistoryElement | | GET | /api/v1/names/history/{uuid} | P: uuid | Find history for name by uuid (exact match) Find history for name by uuid (exact match). Return paged array of name elements. | ResponsePageNameElements |
| | /history/parts/uuid/{uuid} | P: uuid | Finds history for name part by uuid Note uuid (exact match). Returns a list of entries for history of name part. | List of HistoryElement | | | /api/v1/structures/history/{uuid} | P: uuid | Find history for structure by uuid (exact match) Find history for structure by uuid (exact match). Return paged array of structure elements. | ResponsePageStructureElements |

Parts endpoints migrated to Structures endpoints

List of PartElement corresponds to List of StructureElement in ResponsePageStructureElements.

PartElement corresponds to StructureElement.

| HTTP method | Path & Query string | Parameter P – path Q – query | Description | Return | Old → New | HTTP method | Path & Query string | Parameter P – path Q – query | Description | Return |
|-------------|---|------------------------------------|---|---------------------|-----------------|-------------|--|------------------------------------|---|-------------------------------|
| GET | /parts/mnemonic/search/{mnemonic} | P: mnemonic | Finds all name parts by mnemonic search Note mnemonic (search, case sensitive, regex). Returns a list of name parts. | List of PartElement | | GET | /api/v1/structures/mnemonic/{mnemonic} | P: mnemonic | Find valid structures by mnemonic (search) Find valid structures by mnemonic (search). Return paged array of structure elements. | ResponsePageStructureElements |
| GET | /parts/mnemonic/{mnemonic} | P: mnemonic | Finds all name parts by mnemonic Note mnemonic (exact match, case sensitive). Returns a list of name parts. | List of PartElement | | GET | /api/v1/structures/mnemonic/{mnemonic} | P: mnemonic | Find valid structures by mnemonic (search) Find valid structures by mnemonic (search). Return paged array of structure elements. | ResponsePageStructureElements |
| GET | /parts/mnemonicPath/search/{mnemonicPath} | P: mnemonicPath | Finds all name parts by mnemonic path search Note mnemonic path (search, case sensitive, regex). Returns a list of name parts. | List of PartElement | | GET | /api/v1/structures/mnemonicPath/{mnemonicPath} | P: mnemonic | Find valid structures by mnemonic path (search) Find valid structures by mnemonic path (search). Return paged array of structure elements. | ResponsePageStructureElements |

Models

DeviceNameElement* → *NameElement

| DeviceNameElement | | | Old → New | NameElement | | |
|---|--|-------------|-----------------|-----------------|---|--|
| Field | Type | Description | | Field | Type | Description |
| uuid | string UUID | | | uuid | string UUID | Identity (uuid) of the name entry. Value is created server-side. |
| systemGroup | string | | | systemStructure | string | Mnemonic path for for the system structure. |
| system | string | | | systemStructure | string | Mnemonic path for for the system structure. |
| subsystem | string | | | systemStructure | string | Mnemonic path for for the system structure. |
| discipline | string | | | deviceStructure | string | Mnemonic path for for the device structure. |
| deviceType | string | | | deviceStructure | string | Mnemonic path for for the device structure. |
| instanceIndex | string | | | index | string | Index (instance) of the name entry (if the name entry refers to device structure). |
| name | string | | | name | string | Name (verbose) of the name entry. |
| description | string | | | description | string | Description of the name entry. |
| status | string Enum [DELETED, ACTIVE, OBSOLETE] | | | status | string Enum [APPROVED, ARCHIVED, CANCELLED, PENDING, REJECTED] | A collection of values for status of a name and structure entry. It is used to show where in its lifecycle that a name and structure entry is located. |
| | | | | latest | boolean | If the structure entry is latest (with status APPROVED) in its line of entries. |
| | | | | deleted | boolean | If the structure entry is deleted. |
| | | | | | | |
| No exact match for <i>systemGroup</i> , <i>system</i> , <i>subsystem</i> , <i>discipline</i> , <i>deviceType</i> . Instead system structure and device structure mnemonic paths are available. <i>DeviceNameElement</i> status is not same as <i>NameElement</i> status. It corresponds to fields <i>status</i> , <i>latest</i> , <i>deleted</i> . | | | | | | |

HistoryElement → *NameElement*

| HistoryElement | | | Old → New | NameElement | | |
|--|--|-------------|-----------------|-------------|---|--|
| Field | Type | Description | | Field | Type | Description |
| uuid | string UUID | | | uuid | string UUID | Identity (uuid) of the name entry. Value is created server-side. |
| fullName | string | | | name | string | Name (verbose) of the name entry. |
| mnemonic | string | | | index | string | Index (instance) of the name entry (if the name entry refers to device structure). |
| description | string | | | description | string | Description of the name entry. |
| status | string Enum [Archived, Pending, Approved, Rejected, Cancelled, an empty string] | | | status | string Enum [APPROVED, ARCHIVED, CANCELLED, PENDING, REJECTED] | A collection of values for status of a name and structure entry. It is used to show where in its lifecycle that a name and structure entry is located. |
| | | | | latest | boolean | If the name entry is latest (with status APPROVED) in its line of entries. |
| | | | | deleted | boolean | If the name entry is deleted. |
| date | string | | | when | string date-time | Date and time when the name entry was created. |
| user | string | | | who | string | Name (user) of who created the name entry. |
| message | string | | | comment | string | Comment of the name entry command. |
| | | | | | | |
| HistoryElement status is not same as NameElement status. It corresponds to fields status, latest, deleted. | | | | | | |

HistoryElement → StructureElement

| HistoryElement | | | Old → New | StructureElement | | |
|---|--|-------------|-----------------|------------------|---|--|
| Field | Type | Description | | Field | Type | Description |
| uuid | string UUID | | | uuid | string UUID | Identity (uuid) of the name entry. Value is created server-side. |
| fullName | string | | | name | string | Name (verbose) of the name entry. |
| mnemonic | string | | | mnemonic | string | Mnemonic of the structure entry. |
| description | string | | | description | string | Description of the name entry. |
| status | string Enum [Archived, Pending, Approved, Rejected, Cancelled, an empty string] | | | status | string Enum [APPROVED, ARCHIVED, CANCELLED, PENDING, REJECTED] | A collection of values for status of a name and structure entry. It is used to show where in its lifecycle that a name and structure entry is located. |
| | | | | latest | boolean | If the structure entry is latest (with status APPROVED) in its line of entries. |
| | | | | deleted | boolean | If the structure entry is deleted. |
| date | string | | | when | string date-time | Date and time when the name entry was created. |
| user | string | | | who | string | Name (user) of who created the name entry. |
| message | string | | | comment | string | Comment of the name entry command. |
| | | | | | | |
| HistoryElement status is not same as StructureElement status. It corresponds to fields status, latest, deleted. | | | | | | |

PartElement* → *StructureElement

| PartElement | | | Old → New | StructureElement | | |
|---|-------------|-------------|-----------------|------------------|--|--|
| Field | Type | Description | | Field | Type | Description |
| type | string | | | type | string Enum [SYSTEMGROUP, SYSTEM, SUBSYSTEM, DISCIPLINE, DEVICEGROUP, DEVICETYPE] | string A collection of values for type of structure entry. It is used to show what kind of entry and which level in parent / child hierarchy for system structure and device structure. |
| uuid | string UUID | | | uuid | string UUID | Identity (uuid) of the structure entry. Value is created server-side. |
| level | string | | | level | string | Level of the structure entry. |
| name | string | | | name | string | Name of the structure entry. |
| mnemonic | string | | | mnemonic | string | Mnemonic of the structure entry. |
| description | string | | | description | string | Description of the structure entry. |
| status | string | | | status | string Enum [APPROVED, ARCHIVED, CANCELLED, PENDING, REJECTED] | A collection of values for status of a name and structure entry. It is used to show where in its lifecycle that a name and structure entry is located. |
| | | | | latest | boolean | If the structure entry is latest (with status APPROVED) in its line of entries. |
| | | | | deleted | boolean | If the structure entry is deleted. |
| namePath | string | | | | | |
| mnemonicPath | string | | | mnemonicPath | string | Mnemonic path of the structure entry. |
| | | | | | | |
| No correspondence for <i>namePath</i> in <i>StructureElement</i> . <i>PartElement</i> <i>status</i> is not same as <i>StructureElement</i> <i>status</i> . It corresponds to fields <i>status</i> , <i>latest</i> , <i>deleted</i> . | | | | | | |

Examples

Retrieval of content for a name entry

Name: A2T-010PRL:RFS-PRLTap-054

| What | URL | JSON |
|--|--|--|
| Retrieve name in old Naming REST API | /deviceNames/A2T-010PRL%3ARFS-PRLTap-054 | { "uuid": "07bce0ae-0947-47c8-941e-cc76678fd29a", "systemGroup": "Acc", "system": "A2T", "subsystem": "010PRL", "discipline": "RFS", "deviceType": "PRLTap", "instanceIndex": "054", "name": "A2T-010PRL:RFS-PRLTap-054", "description": null, "status": "ACTIVE" } |
| Retrieve name in new Naming REST API | /api/v1/names/A2T-010PRL%3ARFS-PRLTap-054?page=0&pageSize=100 | { "message": "", "details": "", "field": "", "totalCount": 1, "listSize": 1, "page": 0, "pageSize": 100, "list": [{ "uuid": "07bce0ae-0947-47c8-941e-cc76678fd29a", "parentSystemStructure": "c2fce615-ed5d-40f9-8fb5-0b91502536e5", "parentDeviceStructure": "bb1e68a6-e233-4595-ae88-f9186b6760c6", "index": "054", "description": null, "comment": null, "systemStructure": "A2T-010PRL", "deviceStructure": "RFS-PRLTap", "name": "A2T-010PRL:RFS-PRLTap-054", "status": "APPROVED", "latest": true, "deleted": false, "when": "2017-10-20T12:53:27.229+00:00", "who": "johannorin" }] } |
| Retrieve history for name in old Naming REST API | /history/deviceNames/uuid/07bce0ae-0947-47c8-941e-cc76678fd29a | [{ "uuid": "07bce0ae-0947-47c8-941e-cc76678fd29a", "fullName": "A2T-050PRL:RFS-PRLTAP-054", "mnemonic": "054", "description": "", "status": "Approved", "date": "2016-11-07", "user": "johannorin", "message": null }, { "uuid": "07bce0ae-0947-47c8-941e-cc76678fd29a", "fullName": "A2T-050PRL:RFS-PRLTap-054", "mnemonic": "054", "description": "", "status": "Approved", "date": "2016-11-07", "user": "johannorin", "message": null }] |

| | | |
|--|--|--|
| | | <pre> "status": "Approved", "date": "2017-05-16", "user": "danielpisofernandez", "message": null }, { "uuid": "07bce0ae-0947-47c8-941e-cc76678fd29a", "fullName": "A2T-010PRL:RFS-PRLTap-054", "mnemonic": "054", "description": "", "status": "Approved", "date": "2017-10-20", "user": "johannorin", "message": null }] </pre> |
| Retrieve history for name in new Naming REST API | <pre> /api/v1/names/history/07bce0ae-0947-47c8-941e-cc76678fd29a? orderBy=WHEN&isAsc=true&page=0&pageSize=100 </pre> | <pre> { "message": "", "details": "", "field": "", "totalCount": 3, "listSize": 3, "page": 0, "pageSize": 100, "list": [{ "uuid": "07bce0ae-0947-47c8-941e-cc76678fd29a", "parentSystemStructure": "3a46aabf-68ba-402c-9086-d290550c15a6", "parentDeviceStructure": "bb1e68a6-e233-4595-ae88-f9186b6760c6", "index": "054", "description": null, "comment": null, "systemStructure": "A2T-050PRL", "deviceStructure": "RFS-PRLTAP", "name": "A2T-050PRL:RFS-PRLTAP-054", "status": "APPROVED", "latest": false, "deleted": false, "when": "2016-11-07T11:48:38.435+00:00", "who": "johannorin" }, { "uuid": "07bce0ae-0947-47c8-941e-cc76678fd29a", "parentSystemStructure": "3a46aabf-68ba-402c-9086-d290550c15a6", "parentDeviceStructure": "bb1e68a6-e233-4595-ae88-f9186b6760c6", "index": "054", "description": null, "comment": null, "systemStructure": "A2T-050PRL", "deviceStructure": "RFS-PRLTap", "name": "A2T-050PRL:RFS-PRLTap-054", "status": "APPROVED", "latest": false, "deleted": false, "when": "2017-05-16T08:18:00.690+00:00", "who": "danielpisofernandez" }, { "uuid": "07bce0ae-0947-47c8-941e-cc76678fd29a", "parentSystemStructure": "c2fce615-ed5d-40f9-8fb5-0b91502536e5", "parentDeviceStructure": "bb1e68a6-e233-4595-ae88-f9186b6760c6", "index": "054", "description": null, "comment": null, "systemStructure": "A2T-010PRL", "deviceStructure": "RFS-PRLTap", </pre> |

| | | |
|--|--|--|
| | | <pre>"name": "A2T-010PRL:RFS-PRLTap-054", "status": "APPROVED", "latest": true, "deleted": false, "when": "2017-10-20T12:53:27.229+00:00", "who": "johannorin" }] }</pre> |
|--|--|--|

Retrieval of content for a structure entry

Mnemonic path: Spk-010RFC

| What | URL | JSON |
|---|--|---|
| Retrieve structure entry in old Naming REST API | /parts/mnemonicPath/search/Spk-010RFC | <pre>[{ "type": "System Structure", "uuid": "7caafb6b-4a82-4486-aecd-ddc7fe952120", "level": "3", "name": "01 RF Cell", "mnemonic": "010RFC", "description": null, "status": "Approved", "namePath": "Accelerator-Spoke Linac-01 RF Cell", "mnemonicPath": "Acc-Spk-010RFC" }]</pre> |
| Retrieve structure entry in new Naming REST API | /api/v1/structures/mnemonicPath/Spk-010RFC?page=0&pageSize=100 | <pre>{ "message": "", "details": "", "field": "", "totalCount": 1, "listSize": 1, "page": 0, "pageSize": 1, "list": [{ "uuid": "7caafb6b-4a82-4486-aecd-ddc7fe952120", "type": "SUBSYSTEM", "parent": "2854e43f-7968-4c9a-baa2-e261451941f7", "name": "01 RF Cell", "mnemonic": "010RFC", "description": null, "comment": "Approved by Daniel Piso", "mnemonicPath": "Spk-010RFC", "level": 3, "status": "APPROVED", "latest": true, "deleted": false, "when": "2017-08-21T15:15:50.496+00:00", "who": "danielpisofernandez" }] }</pre> |

| | | |
|---|---|---|
| Retrieve history for structure entry in old Naming REST API | /history/parts/uuid/7caafb6b-4a82-4486-aecd-ddc7fe952120 | [{ "uuid": "7caafb6b-4a82-4486-aecd-ddc7fe952120", "fullName": "01 RF Cell", "mnemonic": "010RFC", "description": "", "status": "Pending", "date": "2017-07-20", "user": "aijin", "message": "Requested on behalf of the RF group" }, { "uuid": "7caafb6b-4a82-4486-aecd-ddc7fe952120", "fullName": "01 RF Cell", "mnemonic": "010RFC", "description": "", "status": "Approved", "date": "2017-08-21", "user": "danielpisofernandez", "message": "Approved by Daniel Piso" }] |
| Retrieve history for structure entry in new Naming REST API | /api/v1/structures/history/7caafb6b-4a82-4486-aecd-ddc7fe952120?orderBy=WHEN&isAsc=true&page=0&pageSize=100 | { "message": "", "details": "", "field": "", "totalCount": 2, "listSize": 2, "page": 0, "pageSize": 100, "list": [{ "uuid": "7caafb6b-4a82-4486-aecd-ddc7fe952120", "type": "SUBSYSTEM", "parent": "2854e43f-7968-4c9a-baa2-e261451941f7", "name": "01 RF Cell", "mnemonic": "010RFC", "description": null, "comment": "Requested on behalf of the RF group", "mnemonicPath": null, "level": 3, "status": "PENDING", "latest": false, "deleted": false, "when": "2017-07-20T09:24:24.527+00:00", "who": "aijin" }, { "uuid": "7caafb6b-4a82-4486-aecd-ddc7fe952120", "type": "SUBSYSTEM", "parent": "2854e43f-7968-4c9a-baa2-e261451941f7", "name": "01 RF Cell", "mnemonic": "010RFC", "description": null, "comment": "Approved by Daniel Piso", "mnemonicPath": null, "level": 3, "status": "APPROVED", "latest": true, "deleted": false, "when": "2017-08-21T15:15:50.496+00:00", "who": "danielpisofernandez" }] } |