# Naming REST API – Excel guide

by [lars.johansson@ess.eu](mailto:lars.johansson@ess.eu), ICS Software

**Table of Contents**

[Naming REST API – Excel guide 1](#__RefHeading___Toc224_3823123896)

[Introduction 1](#__RefHeading___Toc8424_4048880772)

[Getting started 2](#__RefHeading___Toc238_3823123896)

[Excel to Json – Excel to Server 3](#__RefHeading___Toc5244_4048880772)

[Names 3](#__RefHeading___Toc5751_4048880772)

[Endpoints 3](#__RefHeading___Toc5753_404888077213)

[Template 3](#__RefHeading___Toc5753_40488807721)

[Columns & Descriptions 3](#__RefHeading___Toc5753_4048880772)

[Operations & Columns 4](#__RefHeading___Toc5755_4048880772)

[Examples 4](#__RefHeading___Toc966_2402704391)

[Structures 5](#__RefHeading___Toc5757_4048880772)

[Endpoints 5](#__RefHeading___Toc5753_4048880772131)

[Template 5](#__RefHeading___Toc8572_4048880772)

[Columns & Descriptions 5](#__RefHeading___Toc5759_4048880772)

[Operations & Columns 5](#__RefHeading___Toc5761_4048880772)

[Examples 6](#__RefHeading___Toc968_2402704391)

[Excel from Json – Excel from Server 7](#__RefHeading___Toc5246_4048880772)

[Names 7](#__RefHeading___Toc5763_4048880772)

[Endpoints 7](#__RefHeading___Toc5753_4048880772132)

[Columns & Descriptions 7](#__RefHeading___Toc5765_4048880772)

[Example 8](#__RefHeading___Toc970_2402704391)

[Structures 8](#__RefHeading___Toc5767_4048880772)

[Endpoints 8](#__RefHeading___Toc5753_4048880772133)

[Columns & Descriptions 8](#__RefHeading___Toc5769_4048880772)

[Example 9](#__RefHeading___Toc972_2402704391)

[Reference 9](#__RefHeading___Toc234_38231238961)

## Introduction

This document will give guide to use of Excel in Naming REST API.

Excel files may be used to upload names and structures to Naming and download names and structures from Naming. However, there are no endpoints for create, update, delete, etc. content by uploading Excel file and there are no endpoints to search for names and structures and download Excel file. Instead, Excel file to upload content is converted to json that then is used for upload. And conversely, for download, json is retrieved from an endpoint and then converted to Excel file that may be downloaded.

This guide will describe

* endpoints to convert Excel to json, and json to Excel
* what Excel files should look like if intention is to convert to json and then use for upload

This guide will describe endpoints to convert Excel to Json and Json to Excel. Other endpoints are not in focus. However some explanation is given by describing what Excel should look like in order to be converted to Json and what Excel will look like after Json has been converted.

When upload is mentioned, it should be understood as what Excel should look like in order for endpoints to convert and upload to run as expected. E.g. upload to create names is about what Excel file should look like in order for convert Excel to Json and create names with converted Json to work.

When download is mentioned, it should be understood as what Excel will look like after endpoints to download and convert content have been run.

Upload and download include names and structures. Endpoints include Read and Create Update Delete.

Examples are available for upload and download of names and structures.

## 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

It is assumed that

* reader is familiar with or willing to learn!
  + Naming backend
    - Swagger UI
  + Documents
    - Introduction to Naming REST API, Cheat Sheet
  + Templates
    - names *NameElementCommand.xlsx*
    - structures *StructureElementCommand.xlsx*

Mentioned templates are used to encapsulate information for upload operations that are sent to Naming. Such operations are Create Update Delete.

There are no templates for Excel files that are sent from Naming. Such operations are Read for which files are generated by server. However, templates and received files share columns. Received files contain additional information that is not needed for operation or it is generated by server.

Upload operations rely on uuid for which names and structures that are affected and referred. Uuid may be found by REST API read operations.

## Excel to Json – Excel to Server

### Names

#### Endpoints

/api/v1/convert/excel2Json/names

Convert names data in Excel file, according to template for names, to json. Expected columns as in Excel template for names. This endpoint is for conversion of name element commands, not name elements.

#### Template

File

*NameElementCommand.xlsx*

#### Columns & Descriptions

Left-to-right

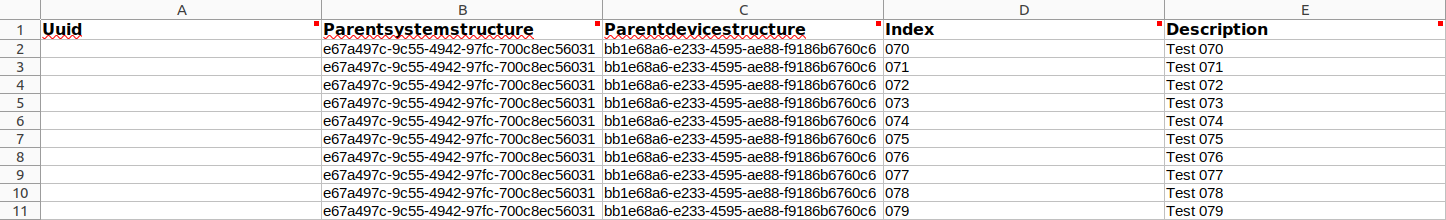
|  |  |
| --- | --- |
| Uuid | Identity (uuid) of the name entry. Value is created server-side. |
| Parentsystemstructure | Identity (uuid) for the system structure parent. |
| Parentdevicestructure | Identity (uuid) for the device structure parent (if the name entry refers to device structure). |
| Index | Index (instance) of the name entry (if the name entry refers to device structure). |
| Description | Description of the name entry. |

#### Operations & Columns

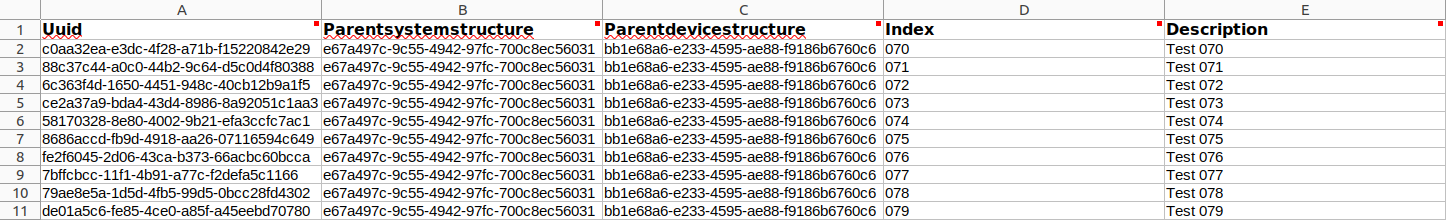
Kind of upload operation and which columns that required, optional or not considered.

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Columns | | Comment |
|  | Required | Optional |  |
| Create | Parentsystemstructure  Description | Parentdevicestructure  Index | Uuid is created server-side |
| Update | Uuid  Parentsystemstructure  Description | Parentdevicestructure  Index |  |
| Delete | Uuid |  | Other attributes are not considered |

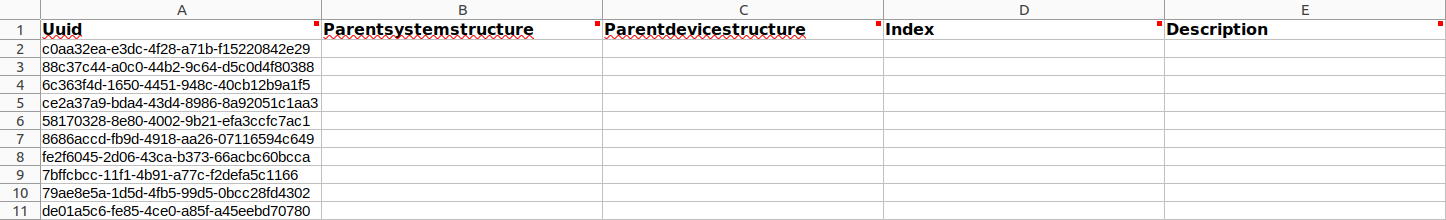
#### Examples

Figure 1:

Create names through upload of json after convert Excel template to json

Figure 2:

Update names through upload of json after convert Excel template to json

Figure 3:

Delete names through upload of json after convert Excel template to json

### Structures

#### Endpoints

/api/v1/convert/excel2Json/structures

Convert structures data in Excel file, according to template for structures, to json. Expected columns as in Excel template for structures. This endpoint is for conversion of structure element commands, not structure elements.

#### Template

File

*StructureElementCommand.xlsx*

#### Columns & Descriptions

Left-to-right

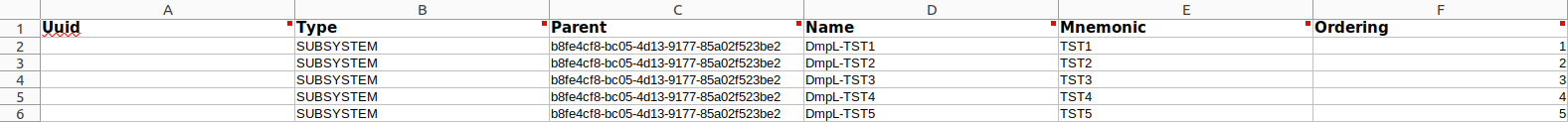
|  |  |
| --- | --- |
| Uuid | Identity (uuid) of the structure entry. Value is created server-side. |
| Type | Type of the structure entry.  Valid values –  SYSTEMGROUP, SYSTEM, SUBSYSTEM, DISCIPLINE, DEVICEGROUP, DEVICETYPE. |
| Parent | Identity (uuid) for the structure entry parent (if the structure entry has a parent). |
| Mnemonic | Mnemonic of the structure entry. |
| Ordering | Ordering of the structure entry. |
| Description | Description (verbose) of the structure entry. |

#### Operations & Columns

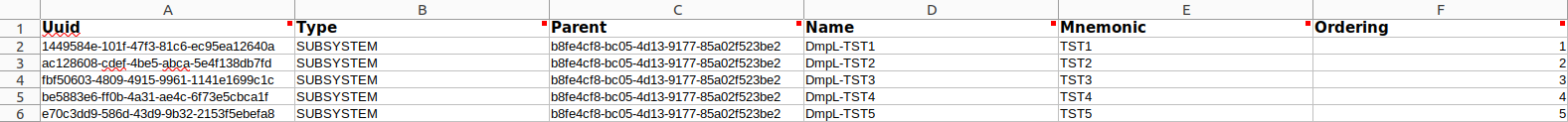
Kind of upload operation and which columns that required, optional or not considered.

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Columns | | Comment |
|  | Required | Optional |  |
| Create | Type  Parent  Mnemonic  Ordering  Description |  | Uuid is created server-side |
| Update | Uuid  Type  Parent  Mnemonic  Ordering  Description |  |  |
| Delete | Uuid  Type |  | Other attributes are not considered |

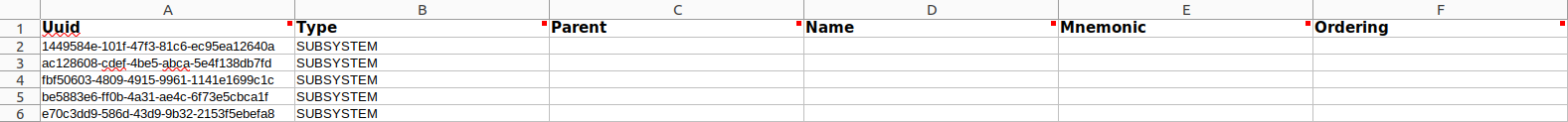
#### Examples

Figure 4:

Create structures through upload of json after convert Excel template to json

Figure 5:

Update structures through upload of json after convert Excel template to json

Figure 6:

Delete structures through upload of json after convert Excel template to json

## Excel from Json – Excel from Server

### Names

No template is available.

#### Endpoints

[/api/v1/convert/json2Excel/names](https://naming-test-02.cslab.esss.lu.se/swagger-ui/index.html" \l "/4. Convert/convertJson2ExcelNames)

Convert names data in json, according to array of name elements, to Excel file. Expected content for json like result for names query. This endpoint is for conversion of name elements, not name element commands.

#### Columns & Descriptions

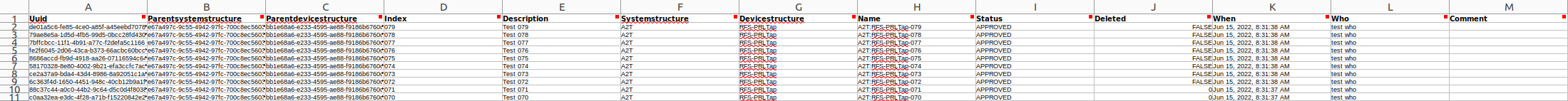
Starts with same columns as NameElementCommand.

Can also be used for upload.

Left-to-right

|  |  |
| --- | --- |
| Uuid | Identity (uuid) of the name entry. Value is created server-side. |
| Parentsystemstructure | Identity (uuid) for the system structure parent. |
| Parentdevicestructure | Identity (uuid) for the device structure parent (if the name entry refers to device structure). |
| Index | Index (instance) of the name entry (if the name entry refers to device structure). |
| Description | Description of the name entry. |
| Systemstructure | Mnemonic path for for the system structure. |
| Devicestructure | Mnemonic path for for the device structure. |
| Name | Name (verbose) of the name entry. |
| Status | Status of the name entry. |
| Deleted | If the name entry is deleted. |
| When | Date and time when the name entry was created. |
| Who | Name (user) of who created the name entry. |
| Comment | Comment of the name entry command. |

#### Example

Figure 7:

Get names through download of Excel file after convert of json to Excel

### Structures

No template is available.

#### Endpoints

/api/v1/convert/json2Excel/structures

Convert structures data in json, according to array of structure elements, to Excel file. Expected content for json like result for structures query. This endpoint is for conversion of structure elements, not structure element commands.

#### Columns & Descriptions

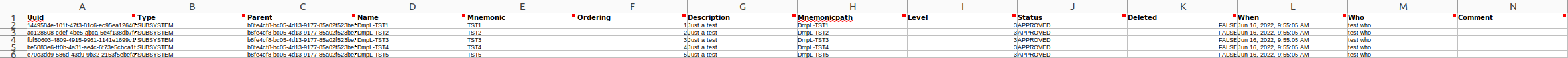
Starts with same columns as StructureElementCommand.

Can also be used for upload.

Left-to-right

|  |  |
| --- | --- |
| Uuid | Identity (uuid) of the structure entry. Value is created server-side. |
| Type | Type of the structure entry.  Valid values –  SYSTEMGROUP, SYSTEM, SUBSYSTEM, DISCIPLINE, DEVICEGROUP, DEVICETYPE. |
| Parent | Identity (uuid) for the structure entry parent (if the structure entry has a parent). |
| Mnemonic | Mnemonic of the structure entry. |
| Ordering | Ordering of the structure entry. |
| Description | Description (verbose) of the structure entry. |
| Mnemonicpath | Mnemonic path of the structure entry. |
| Level | Level of the structure entry. |
| Status | Status of the structure entry. |
| Deleted | If the structure entry is deleted. |
| When | Date and time when the structure entry was created. |
| Who | Name (user) of who created the structure entry. |
| Comment | Comment of the structure entry command. |

#### Example

Figure 8:

Get structures through download of Excel file after convert of json to Excel

## Reference

Naming convention

* <https://chess.esss.lu.se/enovia/link/ESS-0000757/21308.51166.45568.45993/valid>