

Discussion paper “Queries in AASX-Server”

(2022-11-17, Andreas Orzelski, Phoenix Contact)

To investigate the requirements for AAS queries, a basic implementation of ideas has been made in AASX Server (<https://github.com/admin-shell-io/aasx-server/tree/masterV3>).

The implementation is based on the newest work of REST API with V3 data model.

Test servers are running at:

<https://v3-2.admin-shell-io.com/>

<https://v3registry.admin-shell-io.com/>

The corresponding REST APIs are at:

<https://v3.admin-shell-io.com/swagger/index.html>

<https://v3-2.admin-shell-io.com/swagger/index.html>

For queries the REST APIs are used at:

<https://v3.admin-shell-io.com/query/help>

<https://v3-2.admin-shell-io.com/query/help>

An example query is:

SELECT:

submodelelement

FROM:

repository

WHERE:

submodelelement

AND

%idshort contains "Manufacturer"

Such query can be sent to the API by POST and its body or by GET (or browser) with query BASE64URL-encoded (<https://www.base64url.com/>), e.g.:

<https://v3.admin-shell->

[io.com/query/U0VMRUNUOgpzdWJtb2RibGVsZW1lbnQKRlJPTToKcmVwb3NpdG9yeQpXSEVSRTToKc3](https://v3.admin-shell-io.com/query/U0VMRUNUOgpzdWJtb2RibGVsZW1lbnQKRlJPTToKcmVwb3NpdG9yeQpXSEVSRTToKc3)

[VibW9kZWxlbGVtZW50CkFORAolaWRzaG9ydCBjb250YWlucyAiTWFWdWZhY3R1cmVylg](https://v3.admin-shell-io.com/query/U0VMRUNUOgpzdWJtb2RibGVsZW1lbnQKRlJPTToKcmVwb3NpdG9yeQpXSEVSRTToKc3VibW9kZWxlbGVtZW50CkFORAolaWRzaG9ydCBjb250YWlucyAiTWFWdWZhY3R1cmVylg)

(Further examples can be found at the end of the document.)

As shown in the example a syntax referencing to SQL has been used. The general syntax is:

SELECT:

{ set of result elements }

FROM:

{ set of elements to search inside }

WHERE:

{ search conditions which must apply }

With the /query/help an overall syntax description can be retrieved at the endpoint:

Please use POST or add BASE64URL coded query to /query/, e.g. use <https://www.base64url.com/>

```
[ STORE: ] (result of query will be used to search inside by directly following query)
SELECT:
repository | aas | aasid | submodel | submodelid | submodelelement (what will be returned)
FROM:
repository | aas "aasid" | submodel "submodelid" (what will be searched)
WHERE:
aas | submodel | submodelelement (element to search for)
OR | AND
%id | %assetid | %idshort | %value | %semanticid | %path | %semanticidpath <space> == | != | >
| >= | < | <= | contains | !contains <space> "value"
(last line may be repeated after OR and AND)
(options after SELECT: aas [ %id | %idshort | %assetid | !endpoint ])
(options after SELECT: submodel [ %id | %idshort | %semanticid | !endpoint ])
(options after SELECT: submodelelement [ %idshort | %semanticid | %value | !endpoint ])
(WHERE: aas, WHERE: submodel, WHERE: submodelelement may be combined)
```

The query can be made either at the REST API of a specific repository or at a registry.

In case of a registry all included repositories are queried and the combined result is returned.

SELECT: defines the format of the result. The result can be just a “repository”, a list of “aas”, a list of “submodel” or a list of “submodelelement”.

As default the number of successfully fulfilled “WHERE:” conditions and the endpoint of a result element is returned.

Also just a list of “aasid” or “submodelid” may be selected.

For “aas” the following resulting attributes may be selected: %id, %idshort, %assetid.

For “submodel” the following resulting attributes may be selected: %id, %idshort, %semanticid.

For “submodelelement” the following resulting attributes may be selected: %idshort, %semanticid, %value .

If the endpoint shall not be shown in the result, !endpoint can be added.

FROM: defines what will be parsed by the search.

A “repository” or a list of “aas” or a list of “submodels” may be parsed.

In case of aas or submodel the corresponding ID is given and multiple lines may be listed.

WHERE: defines the search condition(s) which must apply to find an element.

Search conditions may be defined for “aas”, “submodel” and/or “submodelements”.

The WHERE: of aas, submodel or submodelelement can be used alone or combined.

First the WHERE: of aas must apply (if existing).

Second the WHERE: of submodel must apply (if existing), but only for the AAS applicable above.

Third the WHERE: of submodelelement must apply (if existing), but only for the submodel(s) applicable above.

At the moment the logical combinations inside WHERE: rules is limited. Only a list of search conditions can be given with OR, i.e. at least one condition must apply, or AND, i.e. all conditions must apply.

A search condition is defined by a triple: %attribute %operation valueString

%attribute can be: %id | %assetid | %idshort | %value | %semanticid | %path | %semanticidpath

%condition can be: == | != | > | >= | < | <= | contains | !contains

%path is the idshort-path in the submodel hierarchy.

%semanticidpath is the path of semantic IDs in the submodel hierarchy.

Multiple search conditions can be listed below OR and AND.

Multiple queries can be combined by **“STORE:”** in the first line.

In that case the next query will search only inside the result elements of the query before, e.g. the first query will have “aas” in SELECT:, then the next query will only search inside those aas.

Example 1:

“Return AAS of Manufacturer Bosch, i.e. ManufacturerName with semanticID 0173-1#02-AAO677#002 contains Bosch in value”

SELECT:

aas

FROM:

repository

WHERE:

submodelelement

AND

%semanticid contains "AAO677"

%value contains "Bosch"

As BASE64URL:

U0VMRUNUOgphYXMKRIJPTToKcmVwb3NpdG9yeQpXSEVSRTToKc3VibW9kZWxlbGVtZW50CkFORAoIc2VtYW50aWNpZCBjb250YWlucyAiQUFPNjc3IgoldmFsdWUgY29udGFpbnMgIkJvc2NoI

<https://v3.admin-shell-io.com/query/U0VMRUNUOgphYXMKRIJPTToKcmVwb3NpdG9yeQpXSEVSRTToKc3VibW9kZWxlbGVtZW50CkFORAoIc2VtYW50aWNpZCBjb250YWlucyAiQUFPNjc3IgoldmFsdWUgY29udGFpbnMgIkJvc2NoI>

Example 2:

“STORE: the result of Example 1 and in the related AAS find the submodel elements with semanticID 0173-1#02-AAD005#008 and return their value, i.e. the file name.”

STORE:

SELECT:

aas

FROM:

repository

WHERE:

submodelelement

AND

%semanticid contains "AAO677"

%value contains "Bosch"

SELECT:

submodelelement %value leendpoint

FROM:

repository

WHERE:

submodelelement

AND

%semanticid contains "AAD005"

As BASE64URL:

U1RPUkU6CINFTEVDVDoKYWFzCkZST006CnJlcG9zaXRvcnkKV0hFUkU6CnN1Ym1vZGVsZWxlbWVudApBTkQKJXNlbWFudGljaWQgY29udGFpbnMgIkFBTzY3NyIKJXZhbHVlIGNvbnRhaW5zICJCb3NjaCIKCINFTEVDVDoKc3VibW9kZWxlbGVtZW50ICV2YWx1ZSAhZW5kcG9pbnQKRlJPTToKcmVwb3NpdG9yeQpXS EVSRTToKc3VibW9kZWxlbGVtZW50CkFORAolc2VtYW50aWNpZCBjb250YWlucyAiQUFEMDA1Ig

<https://v3.admin-shell-io.com/query/U1RPUkU6CINFTEVDVDoKYWFzCkZST006CnJlcG9zaXRvcnkKV0hFUkU6CnN1Ym1vZGVsZWxlbWVudApBTkQKJXNlbWFudGljaWQgY29udGFpbnMgIkFBTzY3NyIKJXZhbHVlIGNvbnRhaW5zICJCb3NjaCIKCINFTEVDVDoKc3VibW9kZWxlbGVtZW50ICV2YWx1ZSAhZW5kcG9pbnQKRlJPTToKcmVwb3NpdG9yeQpXS EVSRTToKc3VibW9kZWxlbGVtZW50CkFORAolc2VtYW50aWNpZCBjb250YWlucyAiQUFEMDA1Ig>

Example 3:

SELECT:

submodelelement

FROM:

repository

WHERE:

aas

AND

%idshort contains Festo

WHERE:

submodel

AND

%idshort == Nameplate

WHERE:

submodelelement

AND

%idshort == Manufacturer*

As BASE64URL:

U0VMRUNUOgpzdWJtb2RibGVsZW1lbnQKRlJPTToKcmVwb3NpdG9yeQpXSEVSRTToKYWFzCkFORAolaWRzaG9ydCBjb250YWlucyBGZXN0bwpXSEVSRTToKc3VibW9kZWwKQU5ECiVpZHNob3J0ID09IE5hbWVwbGF0ZQpXSEVSRTToKc3VibW9kZWxlbGVtZW50CkFORAolaWRzaG9ydCA9PSBnYW51ZmFjdHVyZXIq

<https://v3.admin-shell-io.com/query/U0VMRUNUOgpzdWJtb2RibGVsZW1lbnQKRlJPTToKcmVwb3NpdG9yeQpXSEVSRTToKYWFzCkFORAolaWRzaG9ydCBjb250YWlucyBGZXN0bwpXSEVSRTToKc3VibW9kZWwKQU5ECiVpZHNob3J0ID09IE5hbWVwbGF0ZQpXSEVSRTToKc3VibW9kZWxlbGVtZW50CkFORAolaWRzaG9ydCA9PSBnYW51ZmFjdHVyZXIq>

Example 4:

On server <https://v3registry.admin-shell-io.com/> a registry also with a query registry interface (/queryregistry) is available. It collects the results of <https://v3.admin-shell-io.com/> and <https://v3-2.admin-shell-io.com/> for the respective query:

<https://v3registry.admin-shell-io.com/queryregistry/U0VMRUNUOgpzdWJtb2RlbGVsZW1lbnQKRlJPTToKcmVwb3NpdG9yeQpXSEVSRT0Kc3VibW9kZWxlbGVtZW50CkFORAolaWRzaG9ydCBjb250YWlucyAiTWFudWZhY3R1cmVylg>