Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

Dune Hardware DB Issues & Resolutions

Stephen White, Vladimir Podstavkov

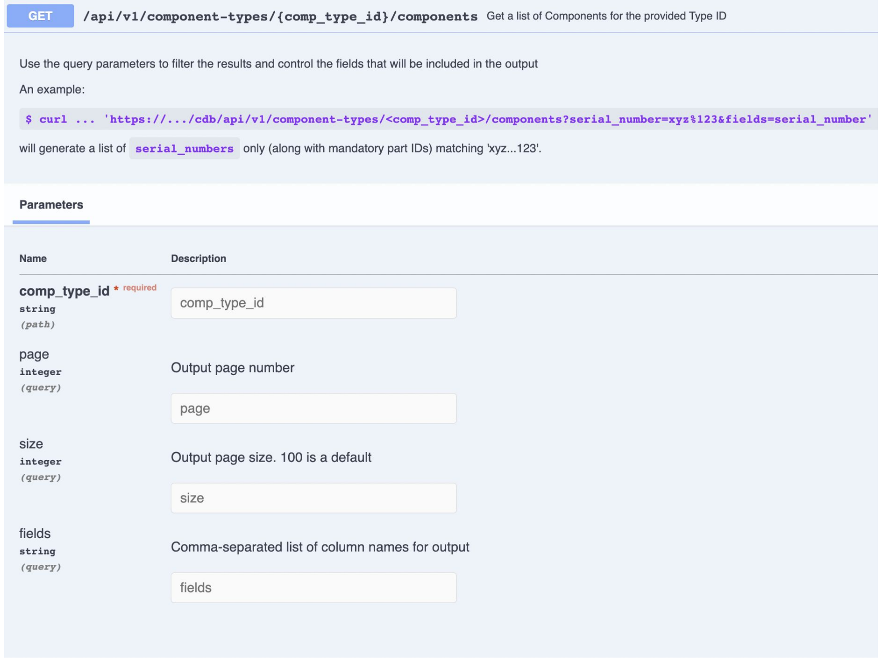
December 2022

Request to Allow Querying By Serial Numbers

- Unique querying by serial numbers was specifically not supported in the original design. They were requested to be optional, by DUNE, as many parts do not have one. (From the original Colorado Meetings.)
- The underlying issue is, serial numbers are under the control of individual 3rd party manufacturers and not all manufacturers make them unique.
- While we will not replace Part-ids with serial numbers, we can provide a solution.

Request to Allow Querying By Serial Numbers (2)

- What we can do.
  - Alter specific APIs to allow filtering by Serial Numbers as well as by other attributes.
  - This APIs already returns a ***list*** of Components
  - The API to be extended is /api/v1/component-types/{comp_type_id}/components
  - Consideration is being given to supporting ranges for Part-ids. No promises.
- If you need other APIs, let us know.



GET `/api/v1/component-types/{comp_type_id}/components` Get a list of Components for the provided Type ID

Use the query parameters to filter the results and control the fields that will be included in the output

An example:

```
$ curl ... 'https://.../cdb/api/v1/component-types/<comp_type_id>/components?serial_number=xyz%123&fields=serial_number'
```

will generate a list of `serial_numbers` only (along with mandatory part IDs) matching 'xyz...123'.

Parameters

Name	Description
comp_type_id * required string *(path)*	comp_type_id
page integer *(query)*	Output page number page
size integer *(query)*	Output page size. 100 is a default size
fields string *(query)*	Comma-separated list of column names for output fields

Wildcards For Bulk Retrievals

A nice side effect of APIs returning lists of components is we can add support for Wildcards on both Part-Ids and Serial Numbers. Our goal is to end the need for parallel requests.

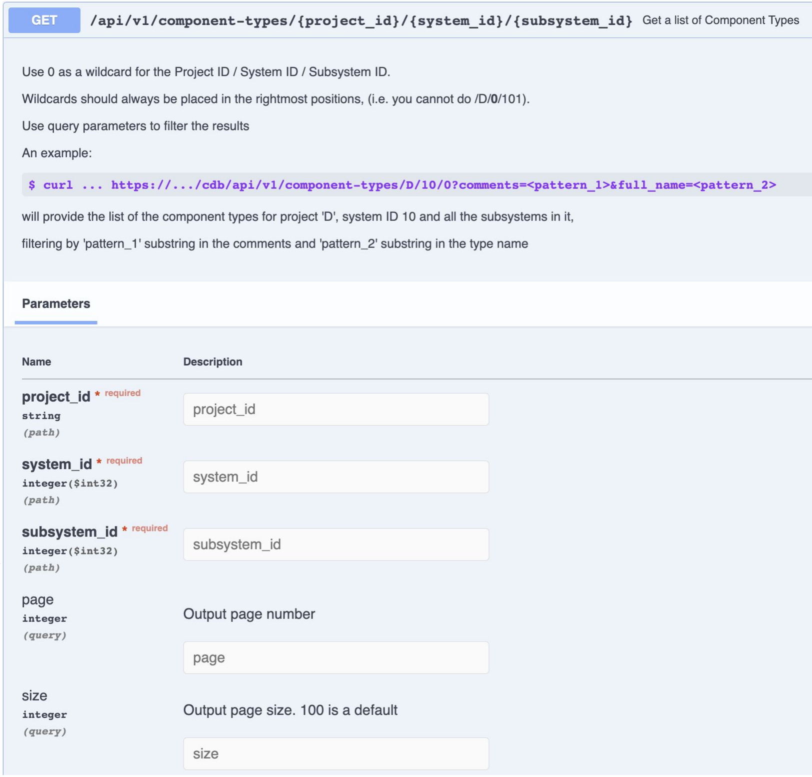
- 'D%1__0' – give me the items from project D having “1” on the fourth place from the end and “0” at the end.
    - % -- wildcard for multiple characters.
    - _ -- wildcard for single characters.
    - We are not using regular expressions.
- This functionality will be limited to the places where returning an array of components would make sense.
- There *may* be a, possibly overridable, limit established on how many records can be returned with one call. Yes, I’m being fuzzy here.
- Return a default set of values with optional additional JSON specified attributes.
    - Defaults have not been specified.

Bulk Component Creation/Updating

- We could not duplicate the unique constraint error which Leon experienced.
    - Vladimir ran 100 curl commands in parallel creating 1000 part-ids.
    - Leon, if you could send us the code which generated the error. We still want to resolve this.
- The existing API, /api/v1/component-types/{part_type_id}/bulk-add, allows you to create as many new component records as are needed in one call.
    - You do not need to do parallel calls to create components.
    - Do be careful. If you ask for 500 new Part-ids, it creates 500 records in the DB.
- We are adding a method to allow bulk updates. So, you can add a serial number, or other data, for each of those newly created 500 components.
    - Generally speaking the API will expect a list of JSON objects.
    - Possible form : [{“part_id”: “XXXX”, “serialnumber” : “YYYY”, “somethingElse” : “aa”,...}, {}, {},...]

Test Specifications & New APIs

- We are adding support for test specifications.
- The specifications will be added with yaml in a new field.
- They will be displayed as non-editable information on the test entry form.
- Versioning of tests will be removed, as mentioned at the last meeting.
- Additional New APIs
  - APIs will be provided for obtaining lists of manufacturers, institutions, and countries with filtering.
  - API for listing all available Component Types with filtering.
  - We will work with Hajime on the parameters.
- Expected Release Date: End of January 2023.
- If we missed any issues please let us know.



`/api/v1/component-types/{project_id}/{system_id}/{subsystem_id}` Get a list of Component Types

Use 0 as a wildcard for the Project ID / System ID / Subsystem ID.

Wildcards should always be placed in the rightmost positions, (i.e. you cannot do /D/**0**/101).

Use query parameters to filter the results

An example:

```
$ curl ... https://.../cdb/api/v1/component-types/D/10/0?comments=<pattern_1>&full_name=<pattern_2>
```

will provide the list of the component types for project 'D', system ID 10 and all the subsystems in it,

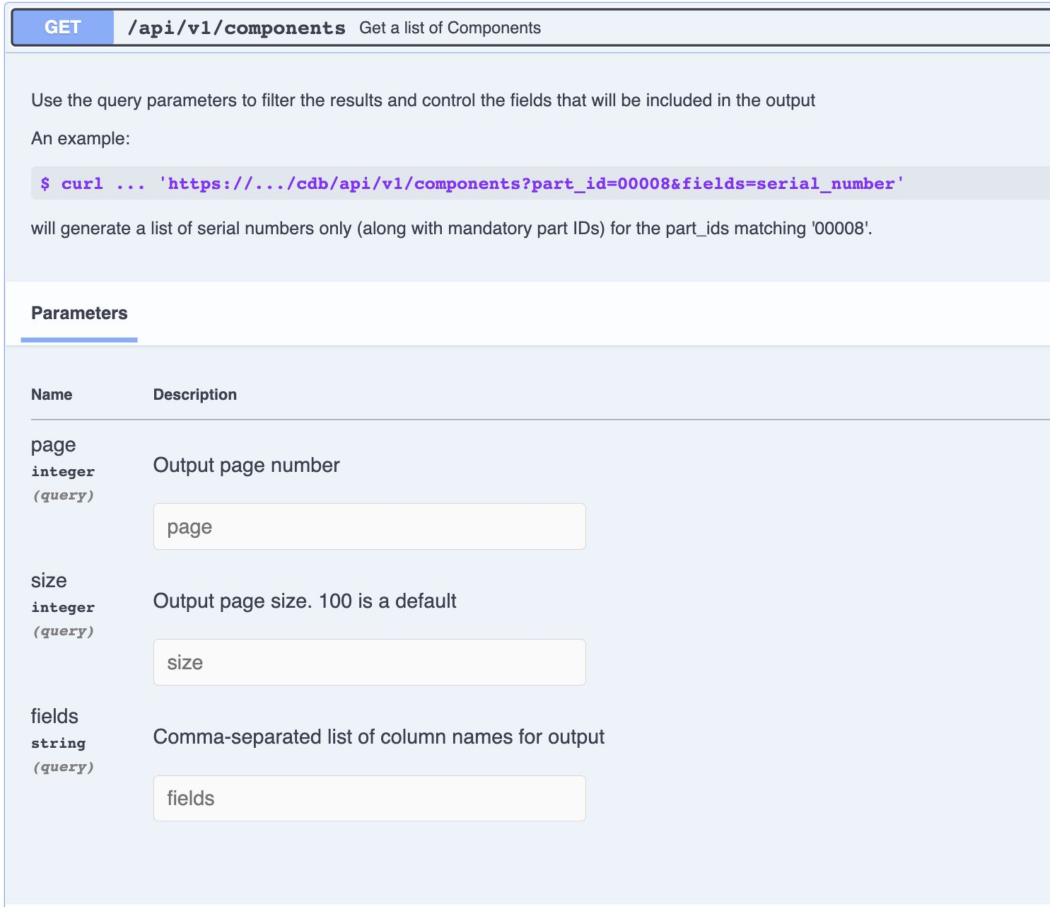
filtering by 'pattern_1' substring in the comments and 'pattern_2' substring in the type name

Parameters

Name	Description
project_id * required string *(path)*	project_id
system_id * required integer($int32) *(path)*	system_id
subsystem_id * required integer($int32) *(path)*	subsystem_id
page integer *(query)*	Output page number page
size integer *(query)*	Output page size. 100 is a default size

Addendum

Example of Bulk Request:



`/api/v1/components` Get a list of Components

Use the query parameters to filter the results and control the fields that will be included in the output

An example:

```
$ curl ... 'https://.../cdb/api/v1/components?part_id=00008&fields=serial_number'
```

will generate a list of serial numbers only (along with mandatory part IDs) for the part_ids matching '00008'.

Parameters

Name	Description
page **integer** *(query)*	Output page number page
size **integer** *(query)*	Output page size. 100 is a default size
fields **string** *(query)*	Comma-separated list of column names for output fields