



## API Parameter Examples

Examples in this documentation are using version 2 of the Fluxx API. The commands are using cURL. The class being used is GrantRequest (unless stated).

---

# Table of Contents

Overview .....	3
Cols .....	3
Data .....	5
Content .....	6
Query .....	7
Filter .....	9
Sort .....	10
Sort_attribute .....	12
Sort_order .....	14
Page .....	16
Per_page .....	18
Priority_index .....	20
Show_mavs .....	21
Enhanced_mavs .....	22
Relation .....	23

---

# Overview

In this section, we will go over some of the commonly used parameters and flags available and their best practices as part of the Fluxx API.

---

## Cols

The cols parameter specifies a list of fields that should be fetched and returned.

Only v2 supports the cols parameter.

Any field (core or dynamic) may be provided.

At least one field must be provided.

### Parameter Example

```
cols=["<--Field-->"]
```

### Example API Call

```
curl --globoff 'https://<your Fluxx site>/api/rest/v2/grant_request?cols=["program_organization_id","program_id","amount_requested"]' --header 'Authorization: Bearer <--token-->'
```

### Returned Payload

```
{
  "records": {
    "grant_request": [
      {
        "id": 20259900,
        "program_organization_id": 10032883,
        "program_id": 3586,
        "amount_requested": 35000
      },
      {
        "id": 20643675,
        "program_organization_id": 10079459,
        "program_id": 3586,
        "amount_requested": 10000
      },
      {
        "id": 20643676,
```

```
"program_organization_id": 10079727,  
"program_id": 3586,  
"amount_requested": 47143  
},  
{  
  "id": 20643677,  
  "program_organization_id": 10123836,  
  "program_id": 3586,  
  "amount_requested": 300000  
}  
]  
},  
"total_pages": 1,  
"total_entries": 5,  
"current_page": 1,  
"per_page": 25  
}
```

---

# Data

The data parameter allows you to pass in data when updating or creating a record.

## Parameter Example

data=<--Field-->:<--Value-->

Please note that the Field should always be wrapped in double quotes. The Value does not need double quotes if it is an integer.

## Example API Call

```
curl --globoff -X PUT "https://<your Fluxx site>/api/rest/v2/grant_request/21277590" -F
'data={"project_title":"This is an API Demo.,"updated_by_id":263502}' -F
'cols=["program_organization_id","program_id","project_title","updated_by"]' --header 'Authorization:
Bearer <--token-->'
```

## Returned Payload

```
{
  "grant_request": {
    "id": 21277590,
    "program_organization_id": 10079459,
    "program_id": 3586,
    "project_title": "This is an API Demo."
    "updated_by": [
      263502
    ]
  }
}
```

---

# Content

The content parameter allows you to specify the document that is uploaded when creating a new document in Fluxx.

The content parameter can only be used on the Model Document class.

## Parameter Example

content=@<--Path to File-->/<--File Name-->

File Name should include the extension.

## Example API Call

If you would like to know which values to pass in the fields, please see the Upload Document via API article.

```
curl --globoff -X POST "https://<your Fluxx site>/api/rest/v2/model_document" -F
'content=@/home/<user>/document_test.txt' -F
'data={"doc_label":"default","owner_model":{"model_type":"grant_request","id":21277590},"storage_t
ype":"file","content_type":"text/plain","file_name":"document_test.txt","document_type":"file","creat
ed_by_id":263502}' -F
'cols=["document_file_name","document_content_type","document_file_size","documentable_type",
documentable_id","model_document_type_id","document_type"]' --header 'Authorization: Bearer <--
token-->'
```

## Returned Payload

```
{
  "model_document": {
    "id": 6293430,
    "document_file_name": "document_test.txt",
    "document_content_type": "text/plain",
    "document_file_size": 36,
    "documentable_type": "GrantRequest",
    "documentable_id": 21277590,
    "document_type": "file"
    "created_by": [
      263502
    ]
  }
}
```

---

# Query

The query parameter allows you to do a full text search against your data. (similar to card searching).

## Parameter Example

query=<--Search Term-->

For multi-word searches add a + symbol in the white spaces.

Example, if you want to search for Unity of Provinces, your search term will be unity+of+provinces.

For this example, the search term does not need to be in single or double quotes.

## Example API Call

```
curl --globoff 'https://<your Fluxx site>/api/rest/v2/grant_request?query=unity+of+provinces&cols=["program_organization_id","program_id","amount_requested"]' --header 'Authorization: Bearer <--token-->'
```

## Returned Payload

```
{
  "records": {
    "grant_request": [
      {
        "id": 20259900,
        "program_organization_id": 10079459,
        "program_id": 3586,
        "amount_requested": 35000
      },
      {
        "id": 20643675,
        "program_organization_id": 10079459,
        "program_id": 3586,
        "amount_requested": 10000
      },
      {
        "id": 20643782,
        "program_organization_id": 10079459,
        "program_id": 3586,
        "amount_requested": 926937
      },
      {
        "id": 20643864,
```

```
"program_organization_id": 10079459,  
"program_id": 3586,  
"amount_requested": 785735  
},  
{  
  "id": 20653211,  
  "program_organization_id": 10079459,  
  "program_id": 3586,  
  "amount_requested": 12500  
}  
]  
},  
"total_pages": 1,  
"total_entries": 5,  
"current_page": 1,  
"per_page": 25  
}
```

---



## Filter

This section has been broken out into its own document to make this document easier to digest.

Please see the [API Filter Document](#).

---

# Sort

The sort parameter allows you to sort your data by a specific field in ascending or descending order.

## Parameter Example

```
sort=["<--Field-->","<--Sort Order-->"]
```

Make sure there are no spaces after the comma.

## Sort Order Options

- asc
- desc

By default, the sort order is ascending.

## Example API Call

```
curl --globoff 'https://<your Fluxx site>/api/rest/v2/grant_request?sort=["amount_requested","asc"]&cols=["program_organization_id","program_id","amount_requested"]' --header 'Authorization: Bearer <--token-->'
```

## Returned Payload

```
{
  "records": {
    "grant_request": [
      {
        "id": 20643675,
        "program_organization_id": 10079459,
        "program_id": 3586,
        "amount_requested": 10000
      },
      {
        "id": 20653211,
        "program_organization_id": 10079459,
        "program_id": 3586,
        "amount_requested": 12500
      },
      {
        "id": 20259900,
        "program_organization_id": 10079459,
        "program_id": 3586,
        "amount_requested": 35000
      },
    ]
  }
}
```

```
{
  "id": 20643864,
  "program_organization_id": 10079459,
  "program_id": 3586,
  "amount_requested": 785735
},
{
  "id": 20643782,
  "program_organization_id": 10079459,
  "program_id": 3586,
  "amount_requested": 926937
}
]
},
"total_pages": 1,
"total_entries": 5,
"current_page": 1,
"per_page": 25
}
```

---

# Sort\_attribute

The sort\_attribute parameter allows you to sort your data by a specific field on a Non-ElasticSearch indexed class in ascending order.

This example will be using the ModelAttributeValue class.

## Parameter Example

sort\_attribute=<--Field-->

A set number of fields are available to sort by and differ by class. Please see the last example in the Sample List/Search Queries list in your API documentation.

## Example API Call

```
curl --globoff 'https://<your Fluxx site>/api/rest/v2/model_attribute_value?sort_attribute=id&cols=["description","value","model_attribute_id"]' --header 'Authorization: Bearer <--token-->'
```

## Returned Payload

```
{
  "records": {
    "model_attribute_value": [
      {
        "id": 11,
        "description": "",
        "value": "Social Media",
        "model_attribute_id": 29
      },
      {
        "id": 15,
        "description": "Education/Curriculum",
        "value": "Education/Curriculum",
        "model_attribute_id": 29
      },
      {
        "id": 22,
        "description": "West Ridge",
        "value": "West Ridge",
        "model_attribute_id": 27
      },
      {
        "id": 23,
        "description": "Uptown",
        "value": "Uptown",

```

```
"model_attribute_id": 27
},
{
  "id": 24,
  "description": "Lincoln Square",
  "value": "Lincoln Square",
  "model_attribute_id": 27
}
]
},
"total_pages": 1,
"total_entries": 5,
"current_page": 1,
"per_page": 25
}
```

---

# Sort\_order

The sort\_order parameter allows you to pick the order in which your data is sorted on a Non ElasticSearch indexed class.

This parameter is paired with the sort\_attribute parameter.

This example will be using the ModelAttributeValue class.

## Parameter Example

sort\_order=<--Sort Order-->

## Sort Order Options

- asc
- desc

## Example API Call

```
curl --globoff 'https://<your Fluxx site>/api/rest/v2/model_attribute_value?sort_attribute=id&sort_order=desc&cols=["description","value","model_attribute_id"]' --header 'Authorization: Bearer <--token-->'
```

## Returned Payload

```
{
  "records": {
    "model_attribute_value": [
      {
        "id": 24,
        "description": "Lincoln Square",
        "value": "Lincoln Square",
        "model_attribute_id": 27
      },
      {
        "id": 23,
        "description": "Uptown",
        "value": "Uptown",
        "model_attribute_id": 27
      },
      {
        "id": 22,
        "description": "West Ridge",
        "value": "West Ridge",
        "model_attribute_id": 27
      },
    ]
  }
}
```

```
{
  "id": 15,
  "description": "Education/Curriculum",
  "value": "Education/Curriculum",
  "model_attribute_id": 29
},
{
  "id": 11,
  "description": "",
  "value": "Social Media",
  "model_attribute_id": 29
}
]
},
"total_pages": 1,
"total_entries": 5,
"current_page": 1,
"per_page": 25
}
```

---

## Page

The page parameter defines the current page of results you are viewing. This parameter can have its value updated as part of your scripting to ‘paginate’ between your output. We highly suggest doing this instead of increasing the “per\_page” parameter as when misused, it can cause system ramifications.

## Parameter Example

```
page= <--Integer-->
```

### Example API Call

```
curl --globoff 'https://<your Fluxx  
site>/api/rest/v2/grant_request?page=2&cols=["program_organization_id","program_id","amount_req  
uested"]' --header 'Authorization: Bearer <--token-->'
```

## Returned Payload

```
{
  "records": {
    "grant_request": [
      {
        "id": 20709831,
        "program_organization_id": 10079459,
        "program_id": 3586,
        "amount_requested": 15000
      },
      {
        "id": 20710759,
        "program_organization_id": 10163603,
        "program_id": 3586,
        "amount_requested": 300000
      },
      {
        "id": 20714740,
        "program_organization_id": 10123836,
        "program_id": 3586,
        "amount_requested": 300000
      },
      {
        "id": 20719868,
        "program_organization_id": 10079459,
        "program_id": 3586,
        "amount_requested": 500000
      },
      {

```



```
    "id": 20719869,  
    "program_organization_id": 10079459,  
    "program_id": 3586,  
    "amount_requested": 1500000  
  }  
]  
,  
"total_pages": 2,  
"total_entries": 30,  
"current_page": 2,  
"per_page": 25  
}
```

---

# Per\_page

The per\_page parameter allows you to pull in more data per pagination.

You cannot exceed 500 records per page.

We recommend using the page parameter instead of increasing the per\_page value.

If no per\_page parameter is provided the system defaults to 25. This is similar to cards in Fluxx.

## Parameter Example

per\_page=<--Integer-->

## Example API Call

In this example, we are only calling 3 records per page.

```
curl --globoff 'https://<your Fluxx site>/api/rest/v2/grant_request?per_page=3&cols=["program_organization_id","program_id","amount_requested"]' --header 'Authorization: Bearer <--token-->'
```

## Returned Payload

```
{
  "records": {
    "grant_request": [
      {
        "id": 20259900,
        "program_organization_id": 10032883,
        "program_id": 3586,
        "amount_requested": 35000
      },
      {
        "id": 20643675,
        "program_organization_id": 10079459,
        "program_id": 3586,
        "amount_requested": 10000
      },
      {
        "id": 20643676,
        "program_organization_id": 10079727,
        "program_id": 3586,
        "amount_requested": 47143
      }
    ]
  }
},
```

```
"total_pages": 10,  
"total_entries": 30,  
"current_page": 1,  
"per_page": 3  
}
```

---

# Priority\_index

Force the record being inserted/updated to be re-indexed with or without priority status. Non-priority indexing means it may take 5 minutes before the record is considered for indexing. Inserts are by default considered priority, and updates non-priority.

## Parameter Example

priority\_index=true

## Example API Call

```
curl --globoff -X POST "https://<your Fluxx site>/api/rest/v2/user?priority_index=true" -F
'data={"first_name":"Test","last_name":"User","salutation":"Dr.,"personal_phone":"555-555-5555"}' -F
'cols=["first_name","last_name","salutation","personal_phone"]' --header 'Authorization: Bearer <--
token-->'
```

## Returned Payload

```
{
  "user": {
    "id": 12346818,
    "first_name": "Test",
    "last_name": "User",
    "salutation": "Dr.",
    "personal_phone": "555-555-5555"
  }
}
```

---

# Show\_mavs

The show\_mavs parameter displays all info about the model\_attribute\_values that are chosen for that dynamic attribute. This allows you to look up all data via the model\_attribute\_values API, including order, hierarchical, etc.

## Parameter Example

show\_mavs=true

## Example API Call

Please note that geographical\_area\_served may not be on your site. This is a dynamic field that is custom to the example site.

```
curl --globoff 'https://<your Fluxx site>/api/rest/v2/grant_request/20729008?cols=["program_organization_id","program_id","geographical_area_served"]&show_mavs=true' --header 'Authorization: Bearer <--token-->'
```

## Returned Payload

```
{
  "grant_request": {
    "id": 20729008,
    "program_organization_id": 10079459,
    "program_id": 3586,
    "geographical_area_served": [
      [
        {
          "id": 9143409,
          "desc": "North America",
          "val": "North America",
          "amount_value": 50
        }
      ],
      [
        {
          "id": 9317909,
          "desc": "Māori",
          "val": "Māori",
          "amount_value": 50
        }
      ]
    ]
  }
}
```

---

# Enhanced\_mavs

The enhanced\_mavs parameter only displays the string value of the model\_attribute\_values that are chosen for that dynamic attribute. However, if the dynamic attribute is a pick-list and hierarchical, it will show all hierarchical values.

## Parameter Example

enhanced\_mavs=true

## Example API Call

Please note that geographical\_area\_served may not be on your site. This is a dynamic field that is custom to the example site.

```
curl --globoff 'https://<your Fluxx site>/api/rest/v2/grant_request/20729008?cols=["program_organization_id","program_id","geographical_area_served"]&enhanced_mavs=true' --header 'Authorization: Bearer <--token-->'
```

## Returned Payload

```
{
  "grant_request": {
    "id": 20729008,
    "program_organization_id": 10079459,
    "program_id": 3586,
    "geographical_area_served": [
      "North America | West | California",
      "Māori"
    ]
  }
}
```

---

# Relation

The relation parameter allows you to access the data related to the records you are attempting to query. For example, because records in models/cards like Request Reports and Request Transactions/Payments can only exist as a child of a Grant Request, they automatically have a data relationship with each other defined by 'model' or 'relation' type fields that point to the IDs of the related data. This parameter allows you to access more data from a single query.

Relations will only be populated as requested. They will not be returned unless explicitly requested. This is to improve performance.

## Parameter Example

```
relation={"relation_name":["<--Field1-->"]}
```

## Example API Call

```
curl --globoff 'https://<your Fluxx site>/api/rest/v2/grant_request?cols=["program_id","amount_requested"]&relation={"program_organization_id":["name"]}' --header 'Authorization: Bearer <--token-->'
```

## Returned Payload

```
{
  "records": {
    "grant_request": [
      {
        "id": 20259900,
        "program_organization_id": [
          {
            "id": 10032883,
            "name": "Unity of Provinces"
          }
        ],
        "program_id": 3586,
        "amount_requested": 35000
      },
      {
        "id": 20643675,
        "program_organization_id": [
          {
            "id": 10079459,
            "name": "Unity of Provinces"
          }
        ],
        "program_id": 3586,
        "amount_requested": 10000
      }
    ]
  }
}
```

```
},
{
  "id": 20643676,
  "program_organization_id": [
    {
      "id": 10079727,
      "name": "Fox Foundation"
    }
  ],
  "program_id": 3586,
  "amount_requested": 47143
},
{
  "id": 20643677,
  "program_organization_id": [
    {
      "id": 10123836,
      "name": "Great Lakes Initiative"
    }
  ],
  "program_id": 3586,
  "amount_requested": 300000
},
{
  "id": 20643782,
  "program_organization_id": [
    {
      "id": 10079459,
      "name": "Unity of Provinces"
    }
  ],
  "program_id": 3586,
  "amount_requested": 926937
}
]
},
"total_pages": 1,
"total_entries": 5,
"current_page": 1,
"per_page": 25
}
```

---