

Aggregate Details

Get Definition Details

HTTP REQUEST: GET

Obtain details about aggregate definitions. You can query by ID, project, project ID, cube ID, status, and subtype.

```
GET /aggregates/orgId/{orgId}
```

URI PARAMETERS

Name	Type	Description
orgId	string	ID of the organization for which you want to obtain aggregate definition details.

QUERY PARAMETERS

Name	Type	Description
id	string	ID of the aggregate definition
project	string	Name of the project for which you need definitions
projectId	string	ID of the project for which you need definitions
cubeId	string	ID of the cube for which you need definitions
status	string	Status of the definitions you need. Value should be 'active', 'invalid', or 'pending'.
subtype	string	Type of the definitions you need. Value should be 'prediction_defined', 'demand_defined', or 'manual' (for user-defined aggregates).

RESPONSES

Code	Description
------	-------------

200	Success.
400	Invalid request for example: Invalid parameters, missing parameters.
401	Unauthorized. This error will occur if calling user does not have access right for the API.
404	Unauthorized. This error will occur if calling user does not have access right for the API.
500	A server error occurred. Check the logs for more details.

200 Reponse Body

```
{
  "response": [
    {
      "data": [
        {
          "id": "c197f3b9-7f4b-4f0a-84b4-def0159a53d3",
          "name": "UDA color",
          "type": "user_defined",
          "subtype": "manual",
          "incremental": false,
          "organization_id": "default",
          "project_id": "8d42bf05-2571-437a-586c-9b60862b5936",
          "project_caption": "Alpha",
          "cube_id": "d623656f-9a54-4e13-49b6-dabc108128f3",
          "cube_name": "Alpha 1",
          "cube_caption": "Alpha 1",
          "connection_id": "con1",
          "system_defined_with_distinct_count": false,
          "promoted": false,
          "distribution": {
            "distribution_keys": [],
            "bucket_count": null
          },
          "partitions": [],
          "indexes": {
            "index_columns": []
          },
        }
      ]
    }
  ]
}
```

```

"attributes": [
  {
    "name": "Color",
    "type": "dimension",
    "partitioned": false
  },
  {
    "name": "Color Key",
    "type": "key",
    "partitioned": false
  }
],
"stats": {
  "created_at": "2023-07-03T09:43:16.622946Z",
  "average_build_duration": 236,
  "query_utilization": 0
},
"latest_instance": {
  "id": "e0bc4492-8ce8-495d-a227-254f6dab834d",
  "status": "active",
  "message": "Normal aggregate activated",
  "table_name": "as_agg_e0bc4492_uda_color",
  "table_schema": "as_adventure",
  "stats": {
    "materialization_start_time": "2023-07-03T09:43:16.840573Z",
    "materialization_end_time": "2023-07-03T09:43:17.077188Z",
    "build_duration": 236,
    "number_of_rows": 8
  },
  "build_query_id": "0a0c1e36-9065-4357-bf73-81b5789c38e3",
  "project_id": "8d42bf05-2571-437a-586c-9b60862b5936",
  "cube_id": "d623656f-9a54-4e13-49b6-dabc108128f3",
  "definition_id": "c197f3b9-7f4b-4f0a-84b4-def0159a53d3",
  "connection_id": "con1"
},
"active_instance": {
  "id": "e0bc4492-8ce8-495d-a227-254f6dab834d",
  "status": "active",
  "message": "Normal aggregate activated",
  "table_name": "as_agg_e0bc4492_uda_color",
  "table_schema": "as_adventure",

```

```

    "stats": {
      "materialization_start_time": "2023-07-03T09:43:16.840573Z",
      "materialization_end_time": "2023-07-03T09:43:17.077188Z",
      "build_duration": 236,
      "number_of_rows": 8
    },
    "build_query_id": "0a0c1e36-9065-4357-bf73-81b5789c38e3",
    "project_id": "8d42bf05-2571-437a-586c-9b60862b5936",
    "cube_id": "d623656f-9a54-4e13-49b6-dabc108128f3",
    "definition_id": "c197f3b9-7f4b-4f0a-84b4-def0159a53d3",
    "connection_id": "con1"
  },
  "most_recent_status": "active",
  "blocked": false
}
],
"total": 1,
"limit": 10,
"offset": 0,
"force_pull": false
}
],
"status": {
  "i18n_message": "response.ok",
  "message": "OK"
}
}
}

```

Get Instance Details

HTTP REQUEST: GET

Obtain details about aggregate instances. You can query by definition ID, project ID, cube ID, status, batch ID, table name, start time, and end time.

```
GET /aggregates/orgId/{orgId}/instances
```

URI PARAMETERS

Name	Type	Description

orgId	string	ID of the organization for which you want to obtain aggregate instance details.
-------	--------	---

QUERY PARAMETERS

Name	Type	Description
definitionId	string	ID of the aggregate definition
projectId	string	ID of the project for which you need instances
cubeId	string	ID of the cube for which you need instances
status	string	Status of the definitions you need. Value should be 'active', 'invalid', 'deleted', or 'pending'.
batchId	string	Batch ID.
tableName	string	Table name.
startTime	string	Start time when the instance was materialized. The format is like this: '2023-07-03T09:43:16'
endTime	string	End time when the instance was materialized. The format is like this: '2023-07-03T09:43:17'

RESPONSES

Code	Description
200	Success.
400	Invalid request for example: Invalid parameters, missing parameters.
401	Unauthorized. This error will occur if calling user does not have access right for the API.
404	Unauthorized. This error will occur if calling user does not have access right for the API.

500	A server error occurred. Check the logs for more details.
-----	---

200 Reponse Body

```
{
  "status": {
    "code": 0,
    "message": "200 OK"
  },
  "responseCreated": "2023-07-03T12:49:03.463342Z",
  "response": {
    "data": [
      {
        "id": "e0bc4492-8ce8-495d-a227-254f6dab834d",
        "status": "active",
        "message": "Normal aggregate activated",
        "table_name": "as_agg_e0bc4492_uda_color",
        "table_schema": "as_adventure",
        "stats": {
          "materialization_start_time": "2023-07-03T09:43:16.840573Z",
          "materialization_end_time": "2023-07-03T09:43:17.077188Z",
          "build_duration": 236,
          "number_of_rows": 8
        },
        "build_query_id": "0a0c1e36-9065-4357-bf73-81b5789c38e3",
        "project_id": "8d42bf05-2571-437a-586c-9b60862b5936",
        "cube_id": "d623656f-9a54-4e13-49b6-dabc108128f3",
        "definition_id": "c197f3b9-7f4b-4f0a-84b4-def0159a53d3",
        "connection_id": "con1"
      },
      {
        "id": "d2c12d26-2dfc-4a35-9d57-32946ac6108e",
        "status": "active",
        "message": "Normal aggregate activated",
        "table_name": "as_agg_d2c12d26_shpdcstmwkofmn",
        "table_schema": "as_adventure",
        "stats": {
          "materialization_start_time": "2023-07-03T08:28:18.740059Z",
          "materialization_end_time": "2023-07-03T08:28:18.826658Z",
          "build_duration": 86,

```

```

    "number_of_rows": 6
  },
  "build_query_id": "ed7b17be-ed4c-47bb-974c-5e6fb92db00b",
  "project_id": "8d42bf05-2571-437a-586c-9b60862b5936",
  "cube_id": "d623656f-9a54-4e13-49b6-dabc108128f3",
  "definition_id": "52a54b83-d27e-4b5a-bc41-872db945a4f7",
  "connection_id": "con1"
}
],
"limit": 10,
"offset": 0,
"force_pull": false
}
}

```

Deactivate Instance

HTTP REQUEST: DELETE

Deactivate the specified aggregate instances.

```
DELETE/aggregates/orgId/{orgId}/definitionId/{defId}/instanceId/{instId}
```

URI PARAMETERS

Name	Type	Description
orgId	string	ID of the organization in which you want to deactivate an instance.
defId	string	ID of the definition for which you want to deactivate an instance.
instId	string	ID of the instance you want to deactivate.

RESPONSES

Code	Description
200	Success.

400	Invalid request for example: Invalid parameters, missing parameters.
401	Unauthorized. This error will occur if calling user does not have access right for the API.
404	Unauthorized. This error will occur if calling user does not have access right for the API.
500	A server error occurred. Check the logs for more details.

200 Reponse Body

```
{  
  "status": {  
    "code": 0,  
    "message": "200 OK"  
  },  
  "responseCreated": "2023-07-03T13:36:58.078573Z",  
  "response": {  
    "deleted": true  
  }  
}
```