

Select a spec

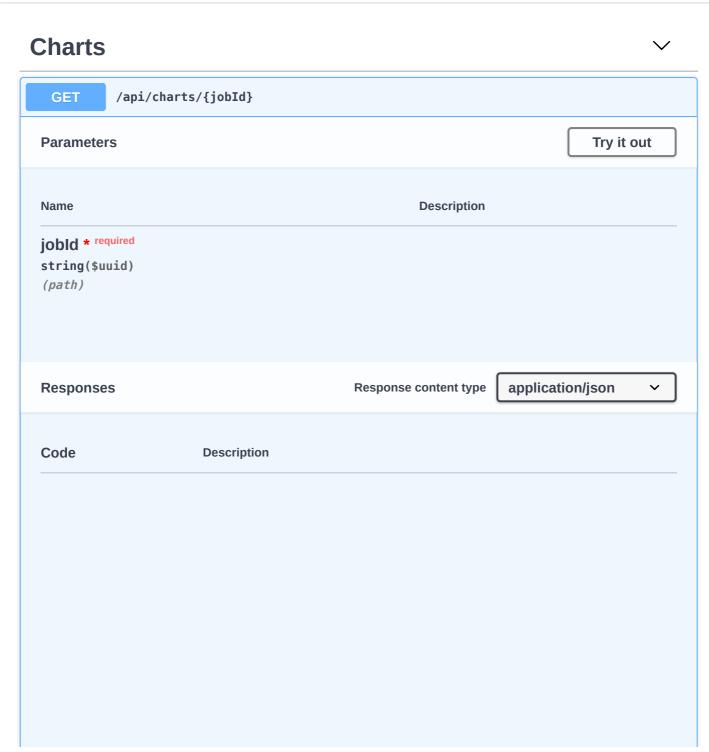
Socneto Api

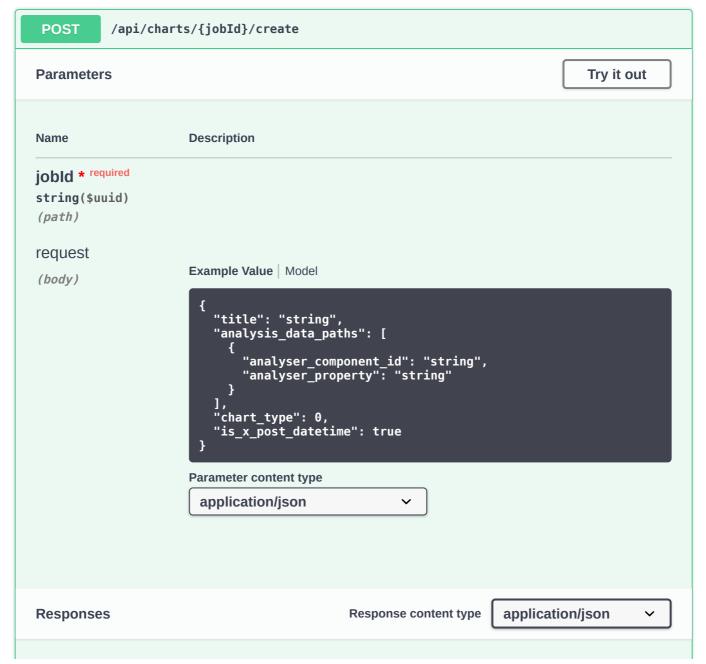


## Socneto Api

/swagger/v1/swagger.json

Socneto task api - no users yet





```
Code

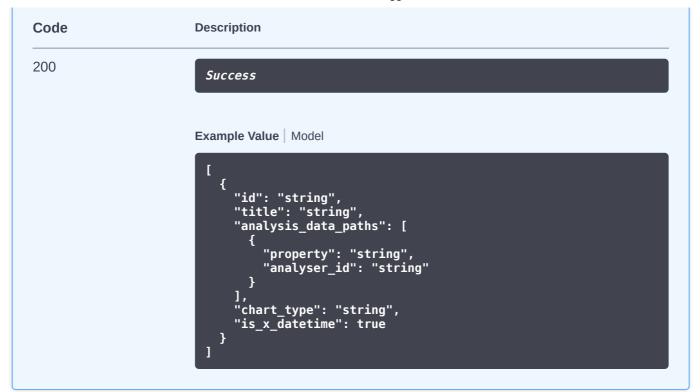
Description

Success

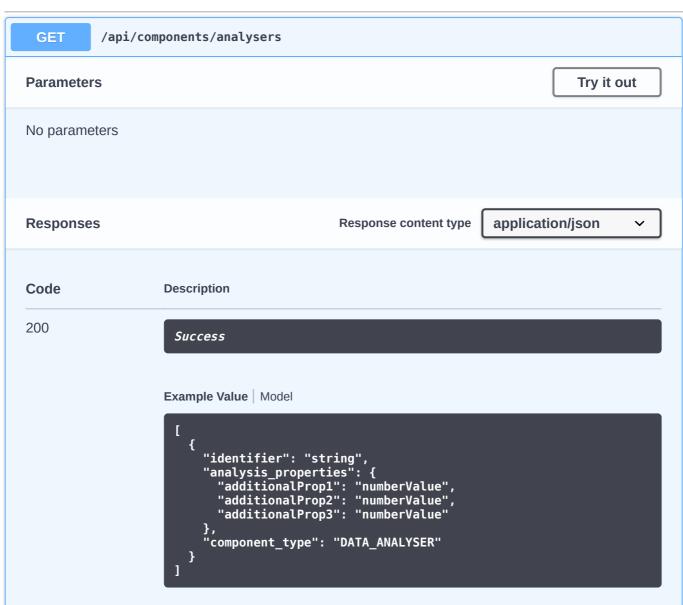
Example Value | Model

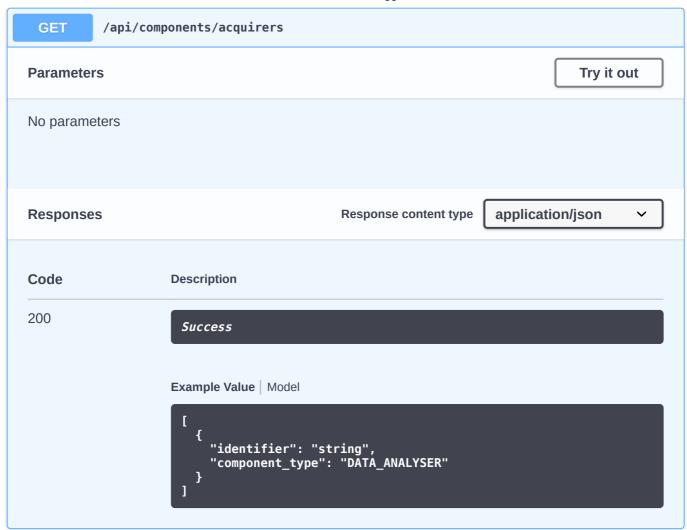
{
    "id": "string",
    "title": "string",
    "analysis_data_paths": [
    {
        "property": "string",
        "analyser_id": "string",
        "chart_type": "string",
    "is_x_datetime": true
}
```



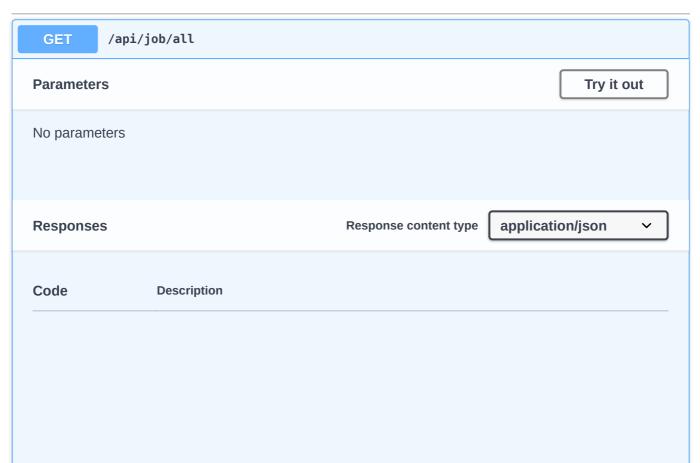


## **Components**

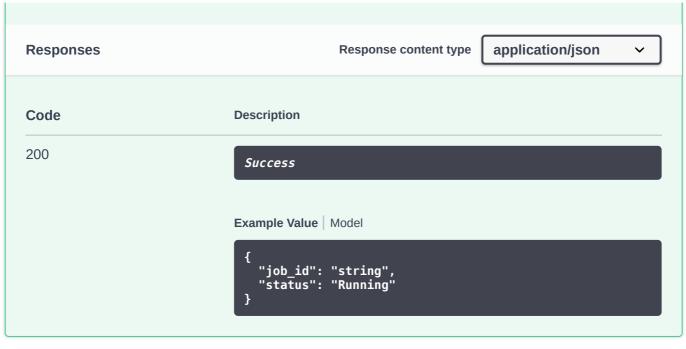


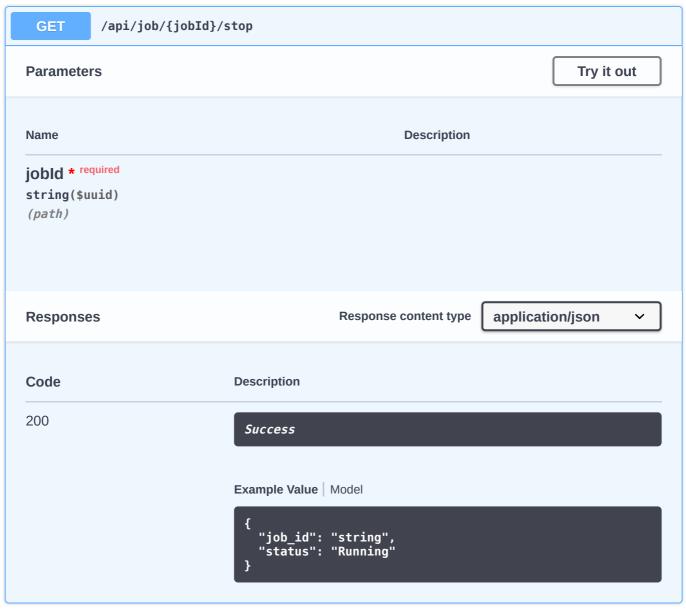


Job



```
POST
                  /api/job/create
Parameters
                                                                                                                                          Try it out
Name
                                          Description
request
                                          Example Value | Model
(body)
                                                "job_name": "string",
"topic_query": "string",
"selected_acquirers": [
   "string"
                                                ],
"selected_analysers": [
                                                    "string"
                                                ],
"attributes": {
                                                    "additionalProp1": {
                                                       "additionalProp1": "string",
"additionalProp2": "string",
"additionalProp3": "string"
                                                    "additionalProp1": "string",
"additionalProp2": "string",
"additionalProp3": "string"
                                                   },
"additionalProp3": {
   "additionalProp1": "string",
   "additionalProp2": "string",
   "additionalProp3": "string"
                                                }
                                            }
                                          Parameter content type
                                             application/json
```





```
GET /api/job/{jobId}/status

Parameters Try it out
```

```
Name
                                                                            Description
jobid * required
string($uuid)
(path)
                                                               Response content type
                                                                                              application/json
Responses
                          Description
Code
200
                            Success
                          Example Value | Model
                               "job_id": "string",
"job_name": "string"
                               "topic_query": "string",
                              "status": "Running",
"started_at": "2020-02-25T11:09:30.066Z",
"finished_at": "2020-02-25T11:09:30.066Z"
```

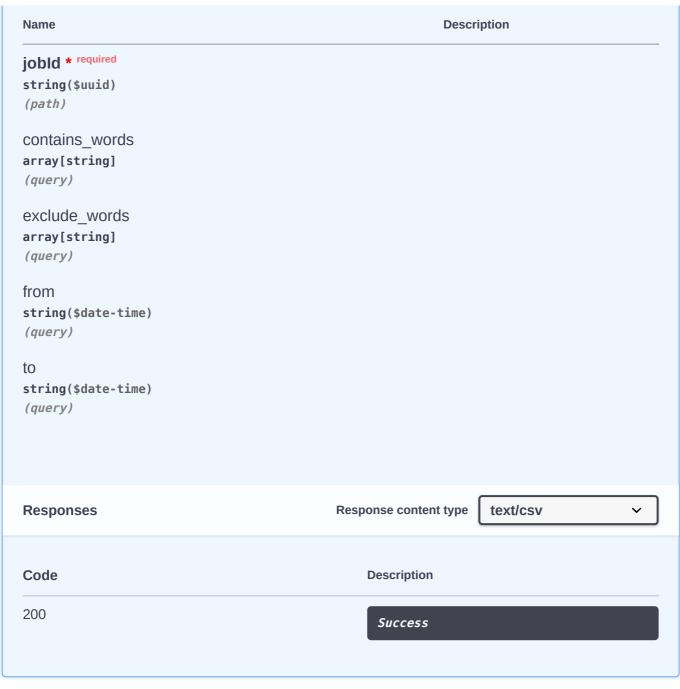
```
GET
           /api/job/{jobId}/posts
                                                                                Try it out
Parameters
Name
                                               Description
jobld * required
string($uuid)
(path)
page
                                               Default value: 1
integer($int32)
(query)
page_size
                                               Default value: 20
integer($int32)
(query)
contains_words
array[string]
(query)
```

```
Name
                                                                     Description
exclude_words
array[string]
(query)
from
string($date-time)
(query)
to
string($date-time)
(query)
Responses
                                                                   Response content type
                                                                                                    application/json
                         Description
Code
200
                            Success
                         Example Value | Model
                               "data": [
                                     "job_id": "string",
"post": {
                                        "original_id": "string",
"text": "string",
"original_text": "string",
"author_id": "string",
"language": "string",
"posted_at": "2020-02-25T11:09:30.071Z"
                                     },
"analyses": {}
                               "pagination": {
                                 "total_size": 0,
"page": 0,
"page_size": 0
```

```
GET /api/job/{jobId}/posts/export

Parameters Try it out

Name Description
```





```
Code

Description

Success

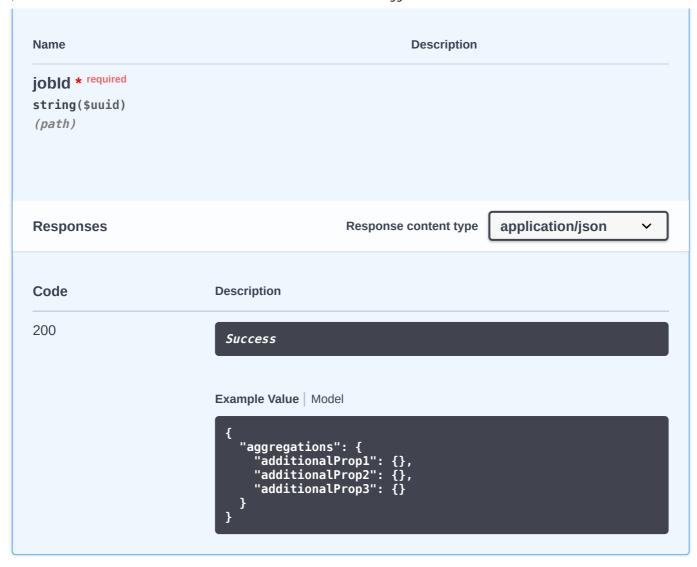
Example Value | Model

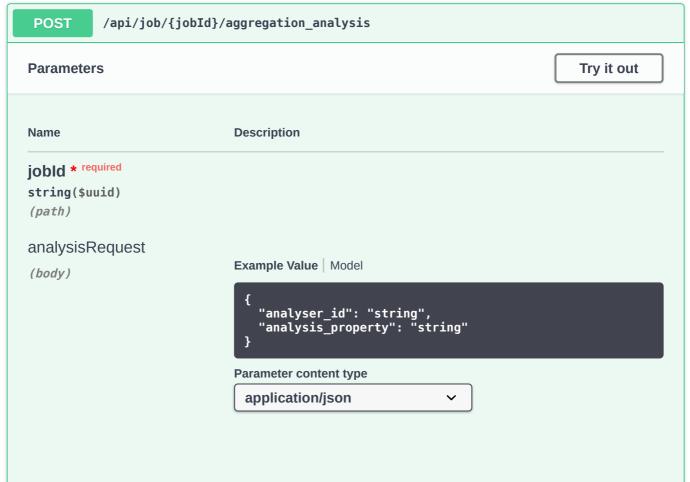
{
    "aggregations": {
        "additionalProp1": {},
        "additionalProp2": {},
        "additionalProp3": {}
}
}
```

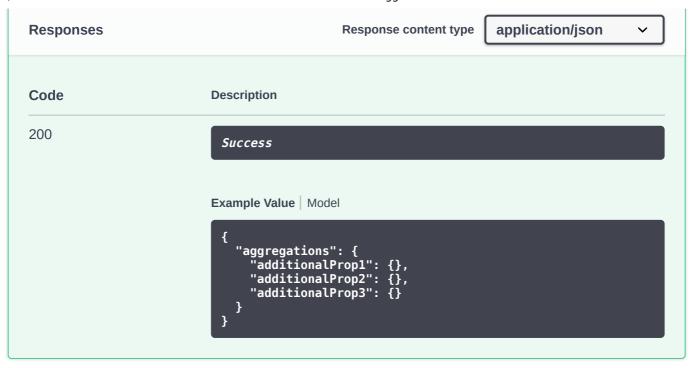
```
/api/job/{jobId}/language_frequency
  GET
                                                                                                            Try it out
Parameters
Name
                                                                          Description
jobld * required
string($uuid)
(path)
Responses
                                                             Response content type
                                                                                            application/json
                                    Description
Code
200
                                      Success
                                    Example Value | Model
                                         "aggregations": {
   "additionalProp1": {},
   "additionalProp2": {},
   "additionalProp3": {}
```

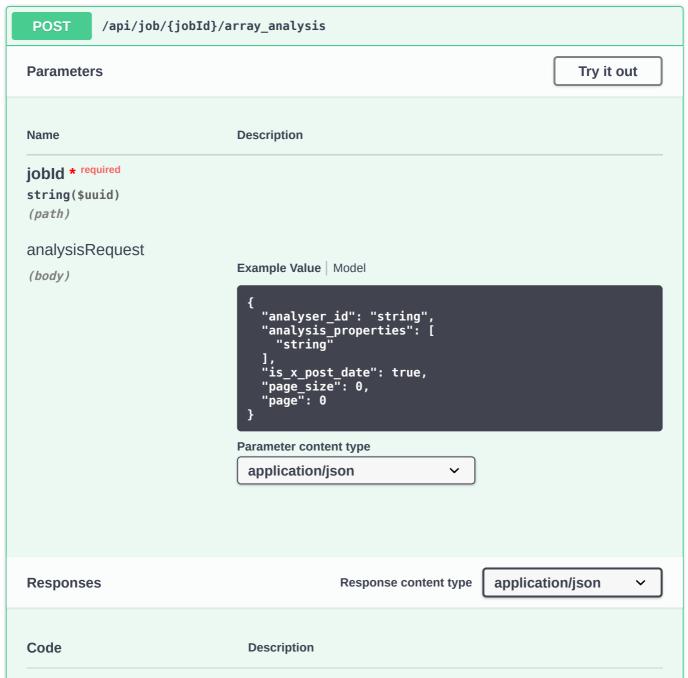
```
GET /api/job/{jobId}/author_frequency

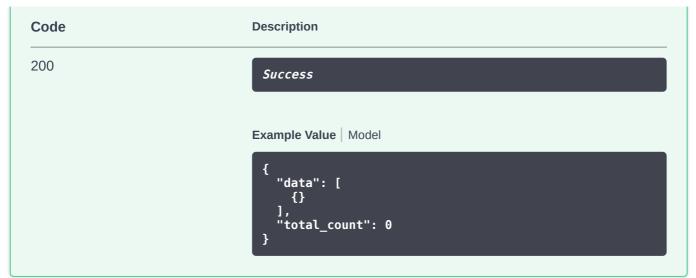
Parameters Try it out
```



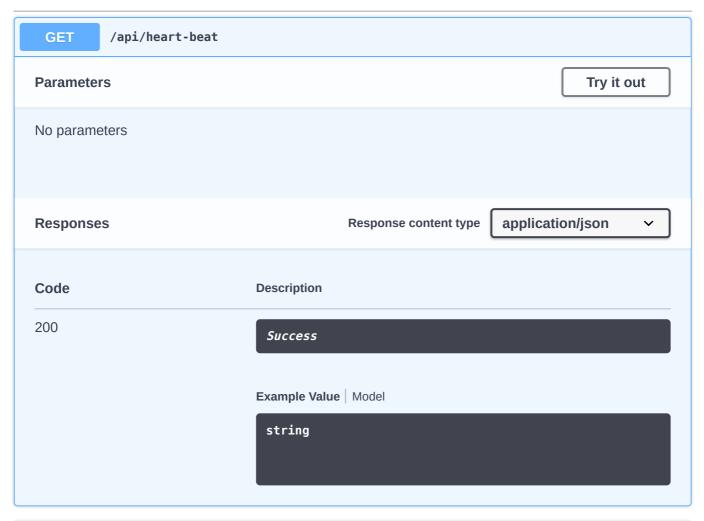


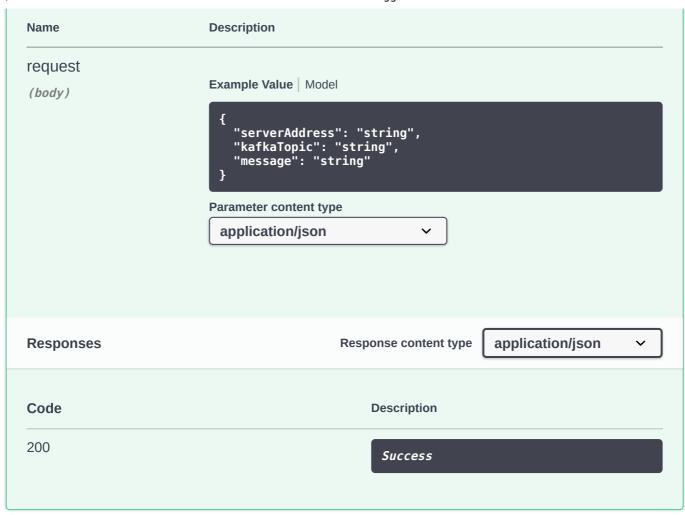


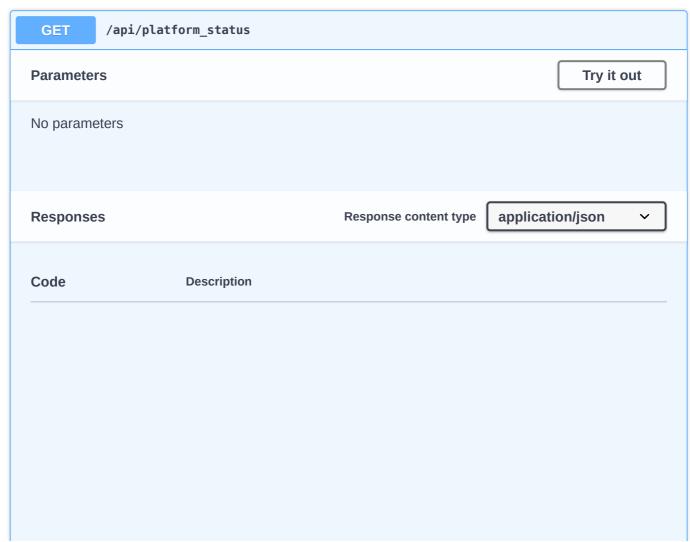


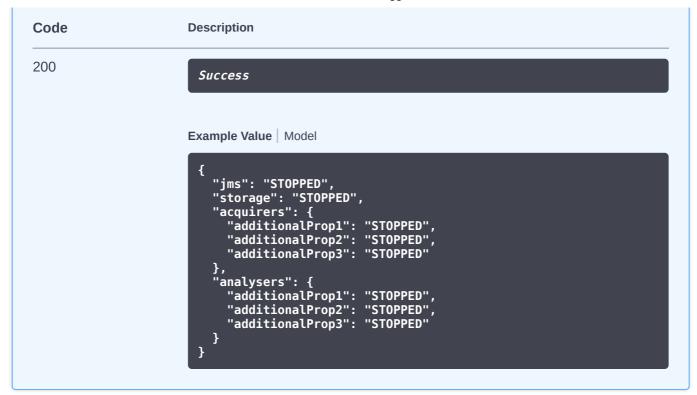


## Management

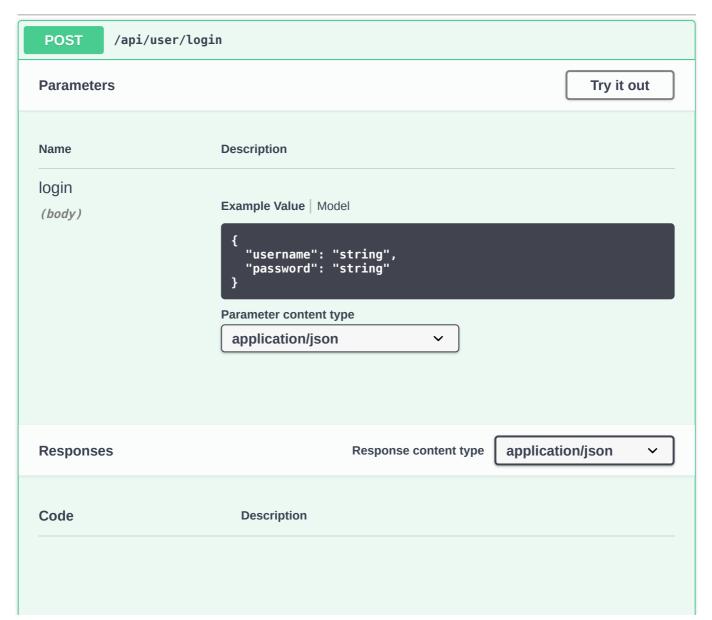








User



```
Code

Description

Success

Example Value | Model

{
    "username": "string"
    }
```

```
Models
   ChartDefinitionDto √ {
                            string($uuid)
       title
                            string
       analysis_data_paths
                            uniqueItems: false
                           AnalysisDataPath > {...}]
       chart_type
is_x_datetime
                            string
                           boolean
   }
   AnalysisDataPath → {
                           string
       property
       analyser_id
                           string
   }
   CreateChartDefinitionRequest > {
       title
                            string
       analysis_data_paths
                            uniqueItems: false
                            AnalysisDataPathRequest > {...}]
       chart_type
                            integer($int32)
                             > Array [ 8 ]
       is_x_post_datetime
                           boolean
   }
   AnalysisDataPathRequest ∨ {
       analyser_component_id string
       analyser_property
                           string
    }
```

```
AnalyserDto \( \{ \) identifier \( \) string \( \) analysis_properties \( \) \( \) \( \) component_type \( \) string \( \) Enum: \( \) \( \) DATA_ANALYSER, DATA_ACQUIRER \( \) \\ \}
```

```
AcquirerDto \( \{ \) identifier string component_type string Enum:
\( \sum \) [ DATA_ANALYSER, DATA_ACQUIRER ]
\( \}
```

```
JobDto ∨ {
   job_id
                         string($uuid)
   job_name
                         string
   topic_query
                         string
   status
                         string
                        Enum:

∨ [ Running, Stopped ]
                        string($date-time)
   started_at
   finished_at
                        string($date-time)
}
```

```
AggregationAnalysisResponse > {
    aggregations > {...}
```

```
GetArrayAnalysisRequest > {
   analyser_id
                       string
   analysis_properties
                        > [...]
   is_x_post_date
                       boolean
                       integer($int32)
   page_size
   page
                       integer($int32)
}
ArrayAnalysisResponse ∨ {
                        > [...]
   total_count
                       integer($int32)
}
ProduceRequest ∨ {
   serverAddress
                       string
   kafkaTopic
                       string
                      string
   message
}
SocnetoComponentsStatus > {
   jms
                       string
                       Enum:

✓ [ STOPPED, STARTING, RUNNING ]
   storage
                       string
                       Enum:

✓ [ STOPPED, STARTING, RUNNING ]
   acquirers
                        > {...}
   analysers
                        > {...}
}
LoginRequest ~ {
   username
                       string
   password
                       string
}
LoginResponse > {
   username
                       string
}
```