

Leibniz Data Manager

REST-API Reference

This notebook documents LDM's API (CKAN), for developers who want to write code that interacts with LDM sites and their data.

LDM's Action API (CKAN) is a powerful, RPC-style API that exposes all of CKAN's core features to API clients. All of a CKAN website's core functionality (everything you can do with the web interface and more) can be used by external code that calls the CKAN API.

For more details and a complete list of API calls available visit the CKAN's "[API Guide](#)": "API Guide"

The following examples demonstrate by using python code scripts how to use LDM's REST API for retrieving and manipulating data and content over an instance of LDM.

SUMMARY:

- Example 1: List Organizations
- Example 2: List Public Datasets
- Example 3: Show Organization's data
- Example 4: Show Dataset's data
- Example 5: Search Datasets
- Example 6: Setting user token to perform privileged operations (Example: Create Organization)
- Example 7: Create Dataset
- Example 8: Show Private Dataset's data
- Example 9: Create User
- Example 10: Upload a Dataset
- Example 11: Download a Dataset

API URL:

The API URL is accessible in <LDM_SITE_URL>/api/3/action/<API_ACTION>, for example:

https://service.tib.eu/ldmservice/api/3/action/organization_list

Setting environment for all examples

```
In [1]: import requests
API_Base_URL = "https://service.tib.eu/ldmservice/api/3/action/"
```

Example 1: List Organizations

```
In [2]: API_URL = API_Base_URL + "organization_list"
print("API URL is: ", API_URL)
print("Method: GET")
```

API URL is: https://service.tib.eu/ldmservice/api/3/action/organization_list
Method: GET

```
In [3]: try:
        response = requests.get(API_URL)
    except requests.exceptions.RequestException as e:
        print("ERROR ACCESSING API: ", API_URL, e.__str__())
```

Showing first 5 results:

```
In [4]: organizations = response.json().get('result')
organizations[0:5]
```

```
Out[4]: ['ag-frahm',
'ag-palm',
'abteilung-ernahrungsphysiologie-und-humanernahrung',
'fachgebiet-it-sicherheit',
'fachgebiet-wissensbasierte-systeme']
```

Example 2: List public Datasets

List of Datasets's names (IDs) available in LDM instance:

```
In [5]: API_URL = API_Base_URL + "package_list"
print("API URL is: ", API_URL)
print("Method: GET")
```

API URL is: https://service.tib.eu/ldmservice/api/3/action/package_list
Method: GET

```
In [6]: try:
        response = requests.get(API_URL)
    except requests.exceptions.RequestException as e:
        print("ERROR ACCESSING API: ", API_URL, e.__str__())
```

Showing first 5 results:

```
In [7]: datasets = response.json().get('result')
datasets[0:5]
```

```
Out[7]: ['code-error-correction-properties-of-an-topological-insulator',
'covid-19-image-repository',
'example-cad-2',
'example-cad-visualizations',
'example-data-formats-visualizations']
```

Example 3: Show Organization's data

```
In [8]: API_URL = API_Base_URL + "organization_show"
print("API URL is: ", API_URL)
print("Method: GET")
```

```
API URL is: https://service.tib.eu/ldmservice/api/3/action/organization_show
Method: GET
```

```
In [9]: params = {"id": "tib"}
try:
    response = requests.get(API_URL, params = params)
except requests.exceptions.RequestException as e:
    print("ERROR ACCESSING API: ", API_URL, e.__str__())
```

Showing results:

```
In [10]: organization = response.json().get('result')
organization
```

```
Out[10]: {'approval_status': 'approved',
'created': '2017-11-23T17:30:37.757128',
'description': 'The German National Library of Science and Technology, abbreviated TIB, is the national library of the Federal Republic of Germany for all fields of engineering, technology, and the natural sciences.',
'display_name': 'TIB',
'id': '0c5362f5-b99e-41db-8256-3d0d7549bf4d',
'image_display_url': 'https://www.tib.eu/typo3conf/ext/tib_tmpl_bootstrap/Resources/Public/images/TIB_Logo_en.png',
'image_url': 'https://www.tib.eu/typo3conf/ext/tib_tmpl_bootstrap/Resources/Public/images/TIB_Logo_en.png',
'is_organization': True,
'name': 'tib',
'num_followers': 2,
'package_count': 27,
'state': 'active',
'title': 'TIB',
'type': 'organization',
'users': [{'about': '',
'activity_streams_email_notifications': False,
'capacity': 'admin',
'created': '2017-08-08T16:45:41.109676',
'display_name': 'admin',
'email_hash': '59235f35e4763abb0b547bd093562f6e',
'fullname': '',
'id': '17755db4-395a-4b3b-ac09-e8e3484ca700',
'image_display_url': '',
'image_url': '',
'name': 'admin',
'number_created_packages': 115,
'state': 'active',
'sysadmin': True},
{'about': None,
'activity_streams_email_notifications': False,
```

```

'capacity': 'editor',
'created': '2022-01-31T15:31:59.423490',
'display_name': 'Ahmad Sakor',
'email_hash': '7aaf25b2751a307b22cb54d8a4c285f4',
'fullname': 'Ahmad Sakor',
'id': '0728da89-2020-4436-afb1-80580688ed61',
'image_display_url': '',
'image_url': '',
'name': 'ahmadsakor',
'number_created_packages': 1,
'state': 'active',
'sysadmin': False},
{'about': None,
'activity_streams_email_notifications': False,
'capacity': 'admin',
'created': '2022-05-24T09:55:02.452392',
'display_name': 'Mauricio Brunet',
'email_hash': 'f041bb8622f3bb1611ec04d41f6baa9d',
'fullname': 'Mauricio Brunet',
'id': '06591881-c785-44cd-be1b-3f35481c23e1',
'image_display_url': 'https://service.tib.eu/ldmservice/uploads/user/2022-05-24-09
5502.363726profile.png',
'image_url': '2022-05-24-095502.363726profile.png',
'name': 'ldm_demo',
'number_created_packages': 0,
'state': 'active',
'sysadmin': False},
{'about': None,
'activity_streams_email_notifications': False,
'capacity': 'admin',
'created': '2022-01-18T19:46:35.199263',
'display_name': 'Maria-Esther Vidal',
'email_hash': '4c7fffc959a5bf7d265806954ff6ab9d',
'fullname': 'Maria-Esther Vidal',
'id': '244275c1-64be-4d06-ba6a-25bb74bc45f5',
'image_display_url': '',
'image_url': '',
'name': 'mevs',
'number_created_packages': 0,
'state': 'active',
'sysadmin': False}],
'extras': [],
'tags': [],
'groups': []

```

Example 4: Show Dataset's data

List metadata from a particular public Dataset by its name (ID):

```

In [11]: API_URL = API_Base_URL + "package_show"
print("API URL is: ", API_URL)
print("Method: GET")

```

```

API URL is: https://service.tib.eu/ldmservice/api/3/action/package_show
Method: GET

```

```
In [12]: dataset_ID = datasets[0]
dataset_ID
```

```
Out[12]: 'code-error-correction-properties-of-an-topological-insulator'
```

```
In [13]: params = {"id": dataset_ID}
try:
    response = requests.get(API_URL, params = params)
except requests.exceptions.RequestException as e:
    print("ERROR ACCESSING API: ", API_URL, e.__str__())
```

Showing results:

```
In [14]: dataset = response.json().get('result')
dataset
```

```
Out[14]: {'author': 'Amit Jamadagni',
'author_email': 'amit.jamadagni@itp.uni-hannover.de',
'creator_user_id': '17755db4-395a-4b3b-ac09-e8e3484ca700',
'id': 'a28770ef-17da-456c-886c-1e2f11985e17',
'isopen': True,
'license_id': 'cc-by',
'license_title': 'Creative Commons Attribution',
'license_url': 'http://www.opendefinition.org/licenses/cc-by',
'maintainer': '',
'maintainer_email': '',
'metadata_created': '2021-07-23T10:18:26.020147',
'metadata_modified': '2021-07-23T10:19:34.751108',
'name': 'code-error-correction-properties-of-an-topological-insulator',
'notes': 'The code files used to generate the plots in the paper: https://arxiv.org/abs/2103.00011',
'num_resources': 1,
'num_tags': 1,
'organization': {'id': '141b74c4-e606-480d-abd1-d2eb1b96fdbe',
'name': 'institut-fur-theoretische-physik',
'title': 'Institut für Theoretische Physik',
'type': 'organization',
'description': 'Appelstraße 2, 30167 Hannover Telefon +49 511 762-3267, -2244 Fax + 49 511 762-3023 E-Mail office@itp.uni-hannover.de\r\n\r\nhttps://www.itp.uni-hannover.de/',
'image_url': '',
'created': '2021-07-23T10:16:29.539258',
'is_organization': True,
'approval_status': 'approved',
'state': 'active'},
'owner_org': '141b74c4-e606-480d-abd1-d2eb1b96fdbe',
'private': False,
'state': 'active',
'title': 'Code: Error-correction properties of an topological insulator',
'type': 'dataset',
'url': '',
'version': '',
'resources': [{'cache_last_updated': None,
'cache_url': None,
'created': '2021-07-23T10:19:34.613278',
```

```

'datastore_active': False,
'description': 'Code used to generate the data in the paper: https://arxiv.org/abs/2103.00011',
'format': 'ZIP',
'hash': '',
'id': '90e50731-8458-4c83-b045-a53563486457',
'last_modified': None,
'metadata_modified': '2021-07-23T10:19:34.753938',
'mimetype': 'application/zip',
'mimetype_inner': None,
'name': 'code.zip',
'package_id': 'a28770ef-17da-456c-886c-1e2f11985e17',
'position': 0,
'resource_type': None,
'size': None,
'state': 'active',
'url': 'https://data.uni-hannover.de/dataset/c81d3262-b3f5-41ef-8dfa-1527c731888e/resource/6716d783-c5d9-47a3-bae2-162d513ac13e/download/code.zip',
'url_type': None}],
'tags': [{'display_name': 'Code',
'id': 'b727a6c5-a351-41e4-8a49-787a9db5c9fc',
'name': 'Code',
'state': 'active',
'vocabulary_id': None}],
'extras': [],
'groups': [],
'relationships_as_subject': [],
'relationships_as_object': []

```

Example 5: Search Datasets

List Datasets performing a query:

```

In [15]: API_URL = API_Base_URL + "package_search"
print("API URL is: ", API_URL)
print("Method: GET")

```

```

API URL is: https://service.tib.eu/ldmservice/api/3/action/package_search
Method: GET

```

```

In [16]: # Example: Searching datasets belonging to TIB organization,
#         ordering results by Organization name alphabetical,
#         and limiting the results to 2 rows
params = {"fq": "type:dataset +organization:tib",
          "sort": "organization asc,title_string asc",
          "rows": 2}

try:
    response = requests.get(API_URL, params = params)
except requests.exceptions.RequestException as e:
    print("ERROR ACCESSING API: ", API_URL, e.__str__())

```

Showing results:

```
In [17]: search_result = response.json().get('result')
search_result
```

```
Out[17]: {'count': 12,
'facets': {},
'results': [{'author': '',
'author_email': '',
'creator_user_id': '17755db4-395a-4b3b-ac09-e8e3484ca700',
'id': '54920aae-f322-4fca-bd09-cd091946632c',
'isopen': True,
'license_id': 'cc-by',
'license_title': 'Creative Commons Attribution',
'license_url': 'http://www.opendefinition.org/licenses/cc-by',
'maintainer': '',
'maintainer_email': '',
'metadata_created': '2017-11-24T13:42:19.407543',
'metadata_modified': '2021-03-03T10:15:41.845072',
'name': 'example-video-2',
'notes': 'Video about auto combustion reactions of STF50 with EDTA+CA: varying ph
i.',
'num_resources': 1,
'num_tags': 7,
'organization': {'id': '0c5362f5-b99e-41db-8256-3d0d7549bf4d',
'name': 'tib',
'title': 'TIB',
'type': 'organization',
'description': 'The German National Library of Science and Technology, abbreviate
d TIB, is the national library of the Federal Republic of Germany for all fields of e
ngineering, technology, and the natural sciences.',
'image_url': 'https://www.tib.eu/typo3conf/ext/tib_tmpl_bootstrap/Resources/Publi
c/images/TIB_Logo_en.png',
'created': '2017-11-23T17:30:37.757128',
'is_organization': True,
'approval_status': 'approved',
'state': 'active'},
'owner_org': '0c5362f5-b99e-41db-8256-3d0d7549bf4d',
'private': False,
'state': 'active',
'title': 'Autocombustion reactions STF50 video',
'type': 'dataset',
'url': '',
'version': '',
'resources': [{'cache_last_updated': None,
'cache_url': None,
'created': '2017-11-24T13:42:36.237930',
'datastore_active': False,
'description': '',
'format': 'video/mp4',
'hash': '',
'id': '8649545f-f1d0-49d2-b9cd-88f2593ec059',
'last_modified': '2017-12-01T16:35:53.307078',
'metadata_modified': None,
'mimetype': 'video/mp4',
'mimetype_inner': None,
'name': 'STF50 autocombustions with varying Phi',
'package_id': '54920aae-f322-4fca-bd09-cd091946632c',
'position': 0,
```

```
    'resource_type': None,
    'size': 71194509,
    'state': 'active',
    'url': 'https://github.com/guillermobet/files/raw/master/STF50_autocombustions_w
ith_varying_phi_v2_HD.mp4',
    'url_type': ''}],
  'tags': [{ 'display_name': 'CA',
    'id': '5df7cf26-78df-4382-b27d-fad8237cf180',
    'name': 'CA',
    'state': 'active',
    'vocabulary_id': None},
  { 'display_name': 'Combustion',
    'id': '23f7f291-52c1-4942-aa23-008a9b23a5e1',
    'name': 'Combustion',
    'state': 'active',
    'vocabulary_id': None},
  { 'display_name': 'EDTA',
    'id': '9d0587af-aad0-4352-ab8f-fc7b90f7430b',
    'name': 'EDTA',
    'state': 'active',
    'vocabulary_id': None},
  { 'display_name': 'Experiment',
    'id': 'a6bbc1be-05c4-406c-8d13-b9e2018b311a',
    'name': 'Experiment',
    'state': 'active',
    'vocabulary_id': None},
  { 'display_name': 'Reactions',
    'id': 'a292a3c1-b272-4c02-bfb2-385e12ff6b66',
    'name': 'Reactions',
    'state': 'active',
    'vocabulary_id': None},
  { 'display_name': 'STF50',
    'id': '53b4f8bd-5778-4ece-b3ac-78e8a60be011',
    'name': 'STF50',
    'state': 'active',
    'vocabulary_id': None},
  { 'display_name': 'Video',
    'id': '7d945dfc-6203-4ef8-8369-90704d7498ac',
    'name': 'Video',
    'state': 'active',
    'vocabulary_id': None}],
  'extras': [],
  'groups': [],
  'relationships_as_subject': [],
  'relationships_as_object': [],
  { 'author': 'Brunet Mauricio',
    'author_email': '',
    'creator_user_id': '17755db4-395a-4b3b-ac09-e8e3484ca700',
    'id': 'c31c9ff2-0ea9-43d8-8f93-2af0eaade9d4',
    'isopen': True,
    'license_id': 'cc-by',
    'license_title': 'Creative Commons Attribution',
    'license_url': 'http://www.opendefinition.org/licenses/cc-by',
    'maintainer': '',
    'maintainer_email': '',
    'metadata_created': '2022-05-10T06:06:18.809145',
    'metadata_modified': '2022-05-10T06:11:58.326588',
    'name': 'example-cad-visualizations',
    'notes': 'Example usage of CAD visualization in 2D and 3D using CKAN Views.',
```



```
'num_resources': 3,
'num_tags': 5,
'orcid': '',
'organization': {'id': '0c5362f5-b99e-41db-8256-3d0d7549bf4d',
'name': 'tib',
'title': 'TIB',
'type': 'organization',
'description': 'The German National Library of Science and Technology, abbreviated TIB, is the national library of the Federal Republic of Germany for all fields of engineering, technology, and the natural sciences.',
'image_url': 'https://www.tib.eu/typo3conf/ext/tib_tpl_bootstrap/Resources/Public/images/TIB_Logo_en.png',
'created': '2017-11-23T17:30:37.757128',
'is_organization': True,
'approval_status': 'approved',
'state': 'active'},
'owner_org': '0c5362f5-b99e-41db-8256-3d0d7549bf4d',
'private': False,
'services_used_list': '',
'state': 'active',
'title': 'Example CAD Visualizations',
'type': 'dataset',
'url': '',
'version': '',
'extra_authors': [{'extra_author': '', 'orcid': ''}],
'extras': [{'__extras': {'id': '93cf96d0-a85f-4b35-b631-e7cd89209e95',
'package_id': 'c31c9ff2-0ea9-43d8-8f93-2af0eaade9d4',
'state': 'active'},
'key': '',
'value': ''}],
'resources': [{'auto_update': 'No',
'auto_update_url': '',
'cache_last_updated': None,
'cache_url': None,
'created': '2022-05-10T06:08:39.012405',
'datastore_active': False,
'description': '',
'format': '',
'hash': '',
'id': '4c0009c0-69f2-49ec-831c-181f97d624a5',
'last_modified': '2022-05-10T06:08:38.988476',
'metadata_modified': '2022-05-10T06:09:15.436821',
'mimetype': None,
'mimetype_inner': None,
'name': 'Example 2D .dwg file',
'package_id': 'c31c9ff2-0ea9-43d8-8f93-2af0eaade9d4',
'position': 0,
'resource_type': None,
'size': 169807,
'state': 'active',
'url': 'https://service.tib.eu/ldmservice/dataset/c31c9ff2-0ea9-43d8-8f93-2af0eaade9d4/resource/4c0009c0-69f2-49ec-831c-181f97d624a5/download/drive_shaft.dwg',
'url_type': 'upload'},
{'auto_update': 'No',
'auto_update_url': '',
'cache_last_updated': None,
'cache_url': None,
'created': '2022-05-10T06:09:15.447478',
'datastore_active': False,
```

```
'description': '',
'format': '',
'hash': '',
'id': '6b12f482-7693-4e36-9f47-fa0d4e03a5dc',
'last_modified': '2022-05-10T06:09:15.426782',
'metadata_modified': '2022-05-10T06:11:58.154051',
'mimetype': None,
'mimetype_inner': None,
'name': 'Example 3D .dwg file',
'package_id': 'c31c9ff2-0ea9-43d8-8f93-2af0eaade9d4',
'position': 1,
'resource_type': None,
'size': 733036,
'state': 'active',
'url': 'https://service.tib.eu/ldmservice/dataset/c31c9ff2-0ea9-43d8-8f93-2af0eaade9d4/resource/6b12f482-7693-4e36-9f47-fa0d4e03a5dc/download/visualization_-_aerial.dwg',
'url_type': 'upload'},
{'auto_update': 'No',
'auto_update_url': '',
'cache_last_updated': None,
'cache_url': None,
'created': '2022-05-10T06:11:58.167565',
'datastore_active': False,
'description': 'Example usage of CAD using Ckan View with information provided by PANGAEA.',
'format': '',
'hash': '',
'id': '3f429389-27fa-4bac-a677-1c989882b06f',
'last_modified': '2022-05-10T06:11:58.142911',
'metadata_modified': '2022-05-10T06:11:58.329896',
'mimetype': None,
'mimetype_inner': None,
'name': 'PANGEA CAD example',
'package_id': 'c31c9ff2-0ea9-43d8-8f93-2af0eaade9d4',
'position': 2,
'resource_type': None,
'size': 942674,
'state': 'active',
'url': 'https://service.tib.eu/ldmservice/dataset/c31c9ff2-0ea9-43d8-8f93-2af0eaade9d4/resource/3f429389-27fa-4bac-a677-1c989882b06f/download/g12_0020_100_5_5_grosskaste.dwg',
'url_type': 'upload'}],
'tags': [{'display_name': '2D',
'id': 'aa5643c3-51ea-4233-a672-6f5a2a7b174e',
'name': '2D',
'state': 'active',
'vocabulary_id': None},
{'display_name': '3D',
'id': 'c98a3ca2-e5c9-4173-93fb-420e0b48e9d8',
'name': '3D',
'state': 'active',
'vocabulary_id': None},
{'display_name': 'CAD',
'id': '80b88538-5f29-4c5f-af29-895228232a10',
'name': 'CAD',
'state': 'active',
'vocabulary_id': None},
{'display_name': 'dwg',
```

```

'id': '675a1366-8d81-4e07-ab30-8c492c34b91d',
'name': 'dwg',
'state': 'active',
'vocabulary_id': None},
{'display_name': 'visualization',
'id': '7ffd8f1d-b342-4349-ae9-a1d5aae5d2bd',
'name': 'visualization',
'state': 'active',
'vocabulary_id': None}],
'groups': [],
'relationships_as_subject': [],
'relationships_as_object': []}],
'sort': 'organization asc,title_string asc',

```

Example 6: Setting user token to perform privileged operations (Example: Create Organization)

Create a new Organization. Only system's administrators are allowed to create Organizations, for that a user token for a valid system administrator user should be provided to the API.

The token can be created in LDM into User's account details, tab "API Tokens" and must be copied and saved at the moment of creation. After that is not visible any more.

In [18]:

```

API_URL = API_Base_URL + "organization_create"
print("API URL is: ", API_URL)
print("Method: POST")

```

API URL is: https://service.tib.eu/ldmservice/api/3/action/organization_create
Method: POST

In [19]:

```
org_dict = {
    "description": "API Example Organization",
    "image_url": "https://www.tib.eu/typo3conf/ext/tib_tmpl_bootstrap/Resources/P
    "name": "api_example1",
}

try:
    response = requests.post(API_URL, data = org_dict)
except requests.exceptions.RequestException as e:
    print("ERROR ACCESSING API: ", API_URL, e.__str__())
```

Showing results (Must fail!):

In [20]:

```
# This operation should fall because we are trying
# to create an Organization without Authorization
organization_create_result = response.json()
organization_create_result
```

Out[20]:

```
{'help': 'https://service.tib.eu/ldmservice/api/3/action/help_show?name=organization_
create',
 'error': {'__type': 'Authorization Error',
 'message': 'Access denied: Action organization_create requires an authenticated use
r'},
 'success': False}
```

Authorizing the user with token:

Notice: this code can't be run live due to security. The authorization token can't be shared. Selectable code bellow.

```
In [ ]: user_token = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJkZFBkZWlrd0xuZ2FGNnV1VENCS2tsceZCNTZlZ2RpaXh5MXh0eS01SR2Nta1JBeFdpPe
try:
    response = requests.post(API_URL, data = org_dict, headers={'Authorization': user_token})
except requests.exceptions.RequestException as e:
    print("ERROR ACCESSING API: ", API_URL, e.__str__())

organization_create_result = response.json()
organization_create_result
```

```
Out[ ]: {'help': 'http://localhost:5000/api/3/action/help_show?name=organization_create',
'success': True,
'result': {'approval_status': 'approved',
'created': '2022-11-03T10:12:29.654384',
'description': 'API Example Organization',
'display_name': 'api_example1',
'id': '418c44ac-f36a-498d-bed8-7d049b2a22e4',
'image_display_url': 'https://www.tib.eu/typo3conf/ext/tib_tmpl_bootstrap/Resources/Public/gfx/logos/tib-full.svg',
'image_url': 'https://www.tib.eu/typo3conf/ext/tib_tmpl_bootstrap/Resources/Public/gfx/logos/tib-full.svg',
'is_organization': True,
'name': 'api_example1',
'num_followers': 0,
'package_count': 0,
'state': 'active',
'title': '',
'type': 'organization',
'users': [{'about': None,
'activity_streams_email_notifications': False,
'capacity': 'admin',
'created': '2017-08-08T16:45:41.109676',
'display_name': 'admin',
'email_hash': '59235f35e4763abb0b547bd093562f6e',
'fullname': None,
'id': '17755db4-395a-4b3b-ac09-e8e3484ca700',
'image_display_url': None,
'image_url': None,
'name': 'admin',
'number_created_packages': 9,
'state': 'active',
'sysadmin': True}],
'extras': [],
'tags': [],
'groups': []}]}
```

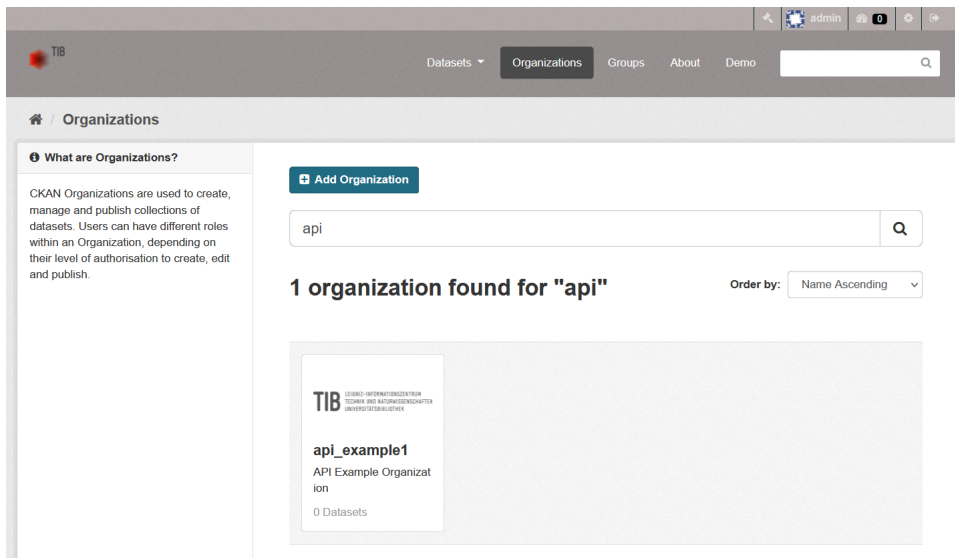
In [21]:

```
# Here you have the code selectable!  
"""  
user_token = 'zQ20Dgx0X0.5bABYIL-_46nLy64m5amx8GB25lqSYi5SLGYRmGDzi8'  
  
try:  
    response = requests.post(API_URL, data = org_dict, headers={'Authorization': user  
except requests.exceptions.RequestException as e:  
    print("ERROR ACCESSING API: ", API_URL, e.__str__())  
  
organization_create_result = response.json()  
organization_create_result  
"""  
"""
```

Out[21]:

```
''
```

The Organization just created in LDM



Example 7: Create Dataset

Create a new Dataset. Only authorized users are allowed to create Datasets, for that a user token for a valid user should be provided to the API.

Notice: The token can be created in LDM into User's account details, tab "API Tokens" and must be copied and saved at the moment of creation. After that is not visible any more.

In [22]:

```
API_URL = API_Base_URL + "package_create"  
print("API URL is: ", API_URL)  
print("Method: POST")
```

```
API URL is: https://service.tib.eu/ldmservice/api/3/action/package_create  
Method: POST
```

In [23]:

```
# Notice: we are creating a private dataset for the user identified by the token
dataset_dict = {
    "name": "api_examples_dataset",
    "title": "API Examples Dataset",
    "private": True,
    "author": "John Doe",
    "author_email": "johndoe@mail.com",
    "notes": "This is the description of the Dataset",
    "owner_org": "tib"}
```

Notice: this code can't be run live due to security. The authorization token can't be shared. Selectable code bellow.

```
In [ ]: user_token = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJEM0paelhkcnZlWVNDNHREdheHoyayVpRG85VVF6NUxuc1JtQWYwTktsMHRZdEgtRk'

try:
    response = requests.post(API_URL, data = dataset_dict, headers={'Authorization': user_token})
except requests.exceptions.RequestException as e:
    print("ERROR ACCESSING API: ", API_URL, e.__str__())
```

In []:

```
dataset_create_result = response.json()
dataset_create_result
```

```
Out[ ]: {'help': 'http://localhost:5000/api/3/action/help_show?name=package_create',
'success': True,
'result': {'author': 'John Doe',
'author_email': 'johndoe@mail.com',
'creator_user_id': '17755db4-395a-4b3b-ac09-e8e3484ca700',
'id': '64ba61cf-d93a-46fe-aacc-f975f03f1f62',
'isopen': False,
'license_title': None,
'metadata_created': '2022-11-03T12:14:58.727042',
'metadata_modified': '2022-11-03T12:14:58.727049',
'name': 'api_examples_dataset',
'notes': 'This is the description of the Dataset',
'num_resources': 0,
'num_tags': 0,
'organization': {'id': '0c5362f5-b99e-41db-8256-3d0d7549bf4d',
'name': 'tib',
'title': 'TIB',
'type': 'organization',
'description': 'The German National Library of Science and Technology, abbreviated TIB, is the national library of the Federal Republic of Germany for all fields of engineering, technology, and the natural sciences.',
'image_url': 'https://www.tib.eu/typo3conf/ext/tib_tpl_bootstrap/Resources/Public/images/TIB_Logo_en.png',
'image_url1': 'https://www.tib.eu/typo3conf/ext/tib_tpl_bootstrap/Resources/Public/images/TIB_Logo_en.png',
'created': '2017-11-23T17:30:37.757128',
'is_organization': True,
'approval_status': 'approved',
'state': 'active'},
'owner_org': '0c5362f5-b99e-41db-8256-3d0d7549bf4d',
'private': True,
'services_used_list': '',
'state': 'active',
'title': 'API Examples Dataset',
'type': 'dataset',
'resources': [],
'tags': [],
'extras': [],
'groups': [],
'relationships_as_subject': [],
'relationships_as_object': [],
'doi': '10.23680/2z9wcpvo',
'doi_status': False,
'domain': 'localhost:5000',
'doi_date_published': None,
'doi_publisher': 'TIB'}}
```

In [24]:

```
# Here you have the code selectable!
"""
user_token = 'iqoULnP29L8fIhD94TyZ0gtnlxQ'

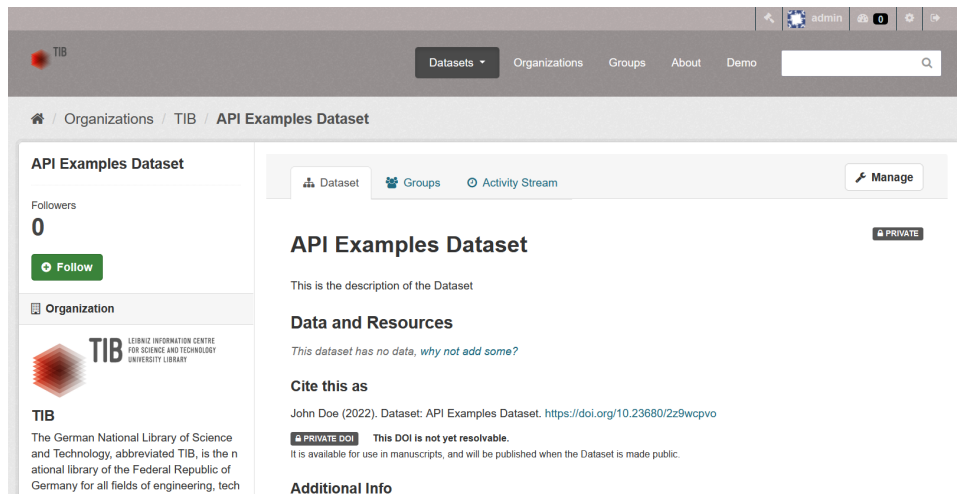
try:
    response = requests.post(API_URL, data = dataset_dict, headers={'Authorization':
except requests.exceptions.RequestException as e:
    print("ERROR ACCESSING API: ", API_URL, e.__str__())

dataset_create_result = response.json()
dataset_create_result
"""
"""
```

Out[24]:

```
''
```

The Dataset just created in LDM



The screenshot shows the TIB LDM interface. The top navigation bar includes 'Datasets', 'Organizations', 'Groups', 'About', and 'Demo'. The main content area is titled 'API Examples Dataset' and is marked as 'PRIVATE'. It features a 'Follow' button, an organization profile for TIB (Leibniz Information Centre for Science and Technology), and a 'Data and Resources' section with a note: 'This dataset has no data, why not add some?'. A 'Cite this as' section provides a citation for John Doe (2022) and a note that the DOI is not yet resolvable. The page also includes a 'Manage' button and tabs for 'Dataset', 'Groups', and 'Activity Stream'.

Example 8: Show Private Dataset's data

List metadata from a particular private Dataset by its name (ID). We are using the same API call used in "Example 3", but now using a valid user token for showing private Datasets.

In [25]:

```
API_URL = API_Base_URL + "package_show"
print("API URL is: ", API_URL)
print("Method: GET")
```

```
API URL is: https://service.tib.eu/ldmservice/api/3/action/package_show
Method: GET
```

In [26]:

```
# We use the Dataset created in Example 7
dataset_ID = "api_examples_dataset"
dataset_ID
```

Out[26]: 'api_examples_dataset'

In [27]:

```
params = {"id": dataset_ID}
try:
    response = requests.get(API_URL, params = params)
except requests.exceptions.RequestException as e:
    print("ERROR ACCESSING API: ", API_URL, e.__str__())
```

Showing results (Must fail): Without the authorization token the Dataset is not accessible

In [28]:

```
dataset = response.json()
dataset
```

```
Out[28]: {'help': 'https://service.tib.eu/ldmservice/api/3/action/help_show?name=package_show',
          'error': {'__type': 'Not Found Error', 'message': 'Not found'},
          'success': False}
```

Now defining the user token.

Notice: this code can't be run live due to security. The authorization token can't be shared. Selectable code bellow.

```
In [ ]: user_token = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJqdGkiOiIxTWxXZ01VeGtoX3Q3TzhkY10zM0hHal96RU0eFFIY01N2Y0enF3a0JyYi13THhLSm'

params = {"id": dataset_ID}
try:
    response = requests.get(API_URL, params = params, headers={'Authorization': user_token})
except requests.exceptions.RequestException as e:
    print("ERROR ACCESSING API: ", API_URL, e.__str__())
```

```
In [ ]: dataset = response.json().get('result')
dataset
```

```
Out[ ]: {'author': 'John Doe',
         'author_email': 'johndoe@mail.com',
         'creator_user_id': '17755db4-395a-4b3b-ac09-e8e3484ca700',
         'id': '64ba61cf-d93a-46fe-aacc-f975f03f1f62',
         'isopen': False,
         'license_title': None,
         'metadata_created': '2022-11-03T12:14:58.727042',
         'metadata_modified': '2022-11-03T12:14:58.727049',
         'name': 'api_examples_dataset',
         'notes': 'This is the description of the Dataset',
         'num_resources': 0,
         'num_tags': 0,
         'organization': {'id': '0c5362f5-b99e-41db-8256-3d0d7549bf4d',
                          'name': 'tib',
                          'title': 'TIB',
                          'type': 'organization',
                          'description': 'The German National Library of Science and Technology, abbreviated TIB, is the national library of the Federal Republic of Germany for all fields of engineering, technology, and the natural sciences.',
                          'image_url': 'https://www.tib.eu/typo3conf/ext/tib_tpl_bootstrap/Resources/Public/images/TIB_Logo_en.png',
                          'created': '2017-11-23T17:30:37.757128',
                          'is_organization': True,
                          'approval_status': 'approved',
                          'state': 'active'},
         'owner_org': '0c5362f5-b99e-41db-8256-3d0d7549bf4d',
         'private': True,
         'services_used_list': '',
         'state': 'active',
         'title': 'API Examples Dataset',
         'type': 'dataset',
         'resources': [],
         'tags': [],
         'extras': [],
         'groups': [],
         'relationships_as_subject': [],
         'relationships_as_object': [],
         'doi': '10.23680/Zz9wcpvo',
         'doi_status': False,
         'domain': 'localhost:5000',
         'doi_date_published': None,
         'doi_publisher': 'TIB'}
```



```
In [29]: # Here you have the code selectable!
        """
        user_token = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJqdGkiOiIwIiwiaWF0IjoiYXZ0eGtoX3Q3TzhkYjYi
        params = {"id": dataset_ID}
        try:
            response = requests.get(API_URL, params = params, headers={'Authorization': user_
        except requests.exceptions.RequestException as e:
            print("ERROR ACCESSING API: ", API_URL, e.__str__())

        dataset = response.json().get('result')
        dataset
        """
        ""
```

Out[29]: ''

Example 9: Create User

Create a new User. Only authorized users are allowed to create Users, for that a user token for a valid authorized user should be provided to the API.

The token can be created in LDM into User's account details, tab "API Tokens" and must be copied and saved at the moment of creation. After that is not visible any more.

```
In [30]: API_URL = API_Base_URL + "user_create"
        print("API URL is: ", API_URL)
        print("Method: POST")
```

```
API URL is: https://service.tib.eu/ldmservice/api/3/action/user_create
Method: POST
```

```
In [31]: user_dict = {
        "name": "john_doe",
        "email": "johndoe@mail.com",
        "password": "password123",
        "fullname": "John Doe"
        }
```

Showing results:

Notice: this code can't be run live due to security. The authorization token can't be shared. Selectable code bellow.

```
In [ ]: user_token = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJqdGkiOiIeTlVlVGVtoX3Q3TzhkYi0zM0hHa196RU00eFFiY011N2Y0enF3a0JyYi13THhLSm
try:
    response = requests.post(API_URL, data = user_dict, headers={'Authorization': user_token})
except requests.exceptions.RequestException as e:
    print("ERROR ACCESSING API: ", API_URL, e.__str__())

user_create_result = response.json()
user_create_result
```

In [32]:

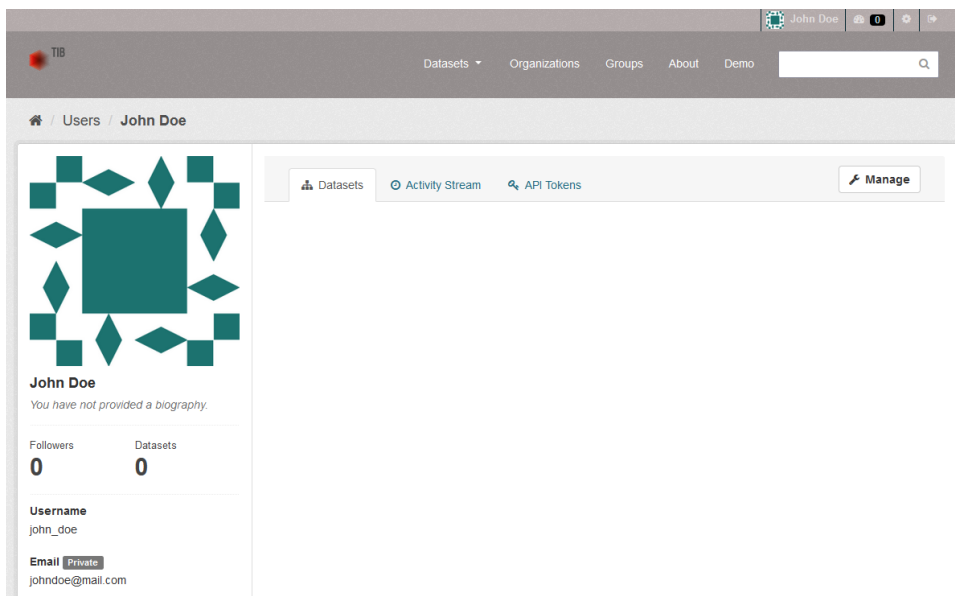
```
# Here you have the code selectable!
"""
user_token = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJqdGkiOiIeTlVlVGVtoX3Q3TzhkYi0zM0hHa196RU00eFFiY011N2Y0enF3a0JyYi13THhLSm

try:
    response = requests.post(API_URL, data = user_dict, headers={'Authorization': user
except requests.exceptions.RequestException as e:
    print("ERROR ACCESSING API: ", API_URL, e.__str__())

user_create_result = response.json()
user_create_result
"""
"""
```

Out[32]:

The User just created in LDM



Example 10: Upload a Dataset

On this context "Upload a Dataset" means the insertion of a new Dataset with metadata and resources into LDM using the API.

The procedure is similar to "Example 7: Create Dataset" but in this example more details are explained.

Only authorized users are allowed to create Datasets, for that a user token for a valid user should be provided to the API.

Notice: The token can be created in LDM into User's account details, tab "API Tokens" and must be copied and saved at the moment of creation. After that is not visible any more.

In [33]:

```
API_URL = API_Base_URL + "package_create"
print("API URL is: ", API_URL)
print("Method: POST")
```

API URL is: https://service.tib.eu/ldmservice/api/3/action/package_create
Method: POST

This is a basic Dataset dictionary with the mandatory metadata needed for the insertion, but it can be extended adding more metadata. The result of the API operation "package_show" (see response in "Example 4: Show Dataset's data") can be used as reference for the addition of more fields to the dictionary.

The metadata fields are defined in LDM's metadata schema and only fields listed in "package_show" operation are allowed.

In [34]:

```
# Notice: we are creating a private dataset for the user identified later by the token
# The following are the required fields

dataset_dict = {
    "name": "api-examples-dataset-2",
    "title": "API Examples Dataset-2",
    "private": True,
    "author": "John Doe",
    "author_email": "johndoe@mail.com",
    "notes": "This is the description of the Dataset",
    "owner_org": "api_example1",
}
```

Name:

Notice the name should be created parsing the "title" to lowercase and replacing whitespaces with scores (-).

```
"name": "api-examples-dataset-2"
```

Public or Private:

Notice the Dataset could be created as public or private for the user authorizing the operation setting "private" to False or True respectively. On this example the Dataset is "private".

```
"private": True
```

Owner Organization:

Notice the Dataset created will belong to the organization described in "owner_org" by the Organization's "ID" or Organization's "name" (see response in "Example 3: Show Organization's data" for reference).

The Organization must already exist in LDM. Otherwise should be created following the instructions on "Example 6: Setting user token to perform privileged operations (Example: Create Organization) before inserting the Dataset".

```
"owner_org": "api-example-organization"
```

In case the Organization doesn't exist and is referenced in a "package_create" operation the API will notice an error:

```
{'help': 'http://localhost:5000/api/3/action/help_show?name=package_create', 'error':
```



```
In [ ]: 1
2 user_token = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJxUkdrRExJdy00VU5CVTdJbk1zeVBSanNmOHdwYjRQQWRScm50UUtDN29MMXYxU
3
4 try:
5     response = requests.post(API_URL, data = resource_dict, headers={'Authorization': user_token})
6 except requests.exceptions.RequestException as e:
7     print("ERROR ACCESSING API: ", API_URL, e.__str__())
8
9 resource_insert_result = response.json()
10 resource_insert_result
```

```
Out[ ]: {'help': 'http://localhost:5000/api/3/action/help_show?name=resource_create',
'success': True,
'result': {'cache_last_updated': None,
'cache_url': None,
'created': '2022-12-06T12:35:39.686019',
'description': 'This is a sample resource for demonstration purposes.',
'format': 'SVG',
'hash': '',
'id': '26bd95b3-48d0-46ed-9cd5-40316b5e5110',
'last_modified': None,
'metadata_modified': '2022-12-06T12:35:39.682249',
'mimetype': 'image/svg+xml',
'mimetype_inner': None,
'name': 'TIB logo',
'package_id': 'ab2c9d5d-0109-4c3a-a61a-34d0a1dd022b',
'position': 0,
'resource_type': None,
'size': None,
'state': 'active',
```

Adding the second resource:

The Dataset in which the Resource will be inserted is defined by Dataset's "name" .

With "url_type" as "upload" the resource will be inserted as a file uploaded into the dataset and copied from "url".

In this case the file must be sent in the request to the API as "multipart/form-data". On Python this can be made sending the opened file as "file" parameter to the request call as we can see below:

In [38]:

```
resource_dict = {"package_id": "api-examples-dataset-2",
                 "description": "This is a second sample resource for demonstration",
                 "format": "svg",
                 "name": "TIB logo - Uploaded",
                 "url": "https://www.tib.eu/typo3conf/ext/tib_tpl_bootstrap/Resource",
                 "url_type": "upload"}
```

Inserting the New Resource:

Notice: this code can't be run live due to security. The authorization token can't be shared. Selectable code below.

```
In [ ]: 1 from urllib.request import urlopen
2
3 user_token = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJxUkdrRExJdy00VU5CVTdJbk1zeVBSanNmOHdwYjRQQWRScm50UUtDN29MMXYxU
4
5 files = [('upload', urlopen(resource_dict['url']))]
6
7 try:
8     response = requests.post(API_URL, data = resource_dict, headers={'Authorization': user_token}, files=files)
9 except requests.exceptions.RequestException as e:
10     print("ERROR ACCESSING API: ", API_URL, e.__str__())
11
12 resource_insert_result = response.json()
13 resource_insert_result
14
```

```

Out[ ]: {'help': 'http://localhost:5000/api/3/action/help_show?name=resource_create',
'success': True,
'result': {'cache_last_updated': None,
'cache_url': None,
'created': '2022-12-06T12:48:13.212965',
'description': 'This is a second sample resource for demonstration purposes.',
'format': 'SVG',
'hash': '',
'id': 'd28524d7-49ca-4c9d-a4dd-4h19hh03dh35'}.

```

The resources added into the Dataset:

The screenshot shows the TIB dataset page for 'API Examples Dataset-2'. The page is private and includes a sidebar with organization information (TIB Leibniz-Informationszentrum) and social media links. The main content area displays the dataset description, a list of data resources (TIB logo and TIB logo - Uploaded), and a table of additional information.

Field	Value
Created	December 6, 2022
Last update	December 6, 2022
Author	John Doe
Author Email	John.Doe

The first resource is of type URL:

The screenshot shows the 'Edit' page for the 'TIB logo' resource. The format is 'SVG'. The 'URL' field contains the path: 'https://www.tib.eu/typo3conf/ext/tib_tpl_bootstrap/Resources/Public/gfx/logos/tib-full.svg'. The 'Name' field contains 'TIB logo'.

The second resource is of type FILE:

The screenshot shows the 'Edit' page for the 'TIB logo - Uploaded' resource. The format is 'SVG'. The 'File' field contains the filename 'tib-full.svg'. The 'Name' field contains 'TIB logo - Uploaded'.

```
In [39]: # Here you have the code above selectable!
"""
from urllib.request import urlopen

user_token = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJxUkdrRExJdy00VU5CVTdJbl
files = [('upload', urlopen(resource_dict['url']))]

try:
    response = requests.post(API_URL, data = resource_dict, headers={'Authorization':
except requests.exceptions.RequestException as e:
    print("ERROR ACCESSING API: ", API_URL, e._str_()).

resource_insert_result = response.json()
resource_insert_result
"""
"""
"""
```

```
Out[39]: ''
```

Example 11: Download a Dataset

On this context "Download a Dataset" means obtaining the Dataset's metadata and Resource's metadata including the URLs for downloading the files using a web browser or programming code.

The procedure is similar to "Example 4: Show Dataset's data" but in this example more details are explained.

Only authorized users are allowed to access private Datasets, for that a user token for a valid user should be provided to the API in that case.

Notice: The token can be created in LDM into User's account details, tab "API Tokens" and must be copied and saved at the moment of creation. After that is not visible any more.

```
In [40]: API_URL = API Base URL + "package_show"
print("API URL is: ", API_URL)
print("Method: GET")
```

```
API URL is: https://service.tib.eu/ldmservice/api/3/action/package_show
Method: GET
```

In this example the Dataset created in previous example "Example 10: Upload a Dataset" is used:

```
In [41]: dataset_ID = "api-examples-dataset-2"
dataset_ID
```

```
Out[41]: 'api-examples-dataset-2'
```

Accessing Dataset's metadata:

Notice: this code can't be run live due to security. The authorization token can't be shared. Selectable code bellow.


```
In [ ]: 1 params = {"id": dataset_ID}
2 user_token = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJxUkdrRExJdy00VU5CVTdJbk1zeVBSanNmOmdWYjRQQRsdmpnUUtDN29MMXYxU
3
4 try:
5     response = requests.get(API_URL, params = params, headers={'Authorization': user_token})
6 except requests.exceptions.RequestException as e:
7     print("ERROR ACCESSING API: ", API_URL, e.__str__())
8
```

Showing Dataset and Resources metadata:

```
In [ ]: 1 dataset = response.json().get('result')
2 dataset
```

```
Out[ ]: {'author': 'John Doe',
'author_email': 'johndoe@mail.com',
'creator_user_id': '17755db4-395a-4b3b-ac09-e8e3484ca700',
'id': 'ab2c9d5d-0109-4c3a-a61a-34d0a1dd022b',
'isopen': False,
'license_title': None,
'metadata_created': '2022-12-06T11:38:41.255770',
'metadata_modified': '2022-12-06T12:49:02.802646',
'name': 'api-examples-dataset-2',
'notes': 'This is the description of the Dataset',
'num_resources': 2,
'num_tags': 0,
'organization': {'id': '418c44ac-f36a-498d-bed8-7d049b2a22e4',
'name': 'api_example1',
'title': '',
'type': 'organization',
'description': 'API Example Organization',
'image_url': 'https://www.tib.eu/typo3conf/ext/tib_tmpl_bootstrap/Resources/Public/gfx/logos/tib-full.svg',
'created': '2022-11-03T10:12:29.654384',
'is_organization': True,
'approval_status': 'approved',
'state': 'active'},
'owner_org': '418c44ac-f36a-498d-bed8-7d049b2a22e4',
'private': True,
'services_used_list': '',
'state': 'active',
'title': 'API Examples Dataset-2',
'type': 'dataset',
'resources': [{'cache_last_updated': None,
'cache_url': None,
'created': '2022-12-06T12:35:39.686019',
'description': 'This is a sample resource for demonstration purposes.',
'format': 'SVG',
'hash': '',
'id': '26bd95b3-48d0-46ed-9cd5-40316b5e5110',
'last_modified': None,
'metadata_modified': '2022-12-06T12:35:39.682249',
'mimetype': 'image/svg+xml',
'mimetype_inner': None,
'name': 'TIB logo',
'package_id': 'ab2c9d5d-0109-4c3a-a61a-34d0a1dd022b',
'position': 0,
'resource_type': None,
'size': None,
'state': 'active',
'url': 'https://www.tib.eu/typo3conf/ext/tib_tmpl_bootstrap/Resources/Public/gfx/logos/tib-full.svg',
'url_type': ''},
{'cache_last_updated': None,
'cache_url': None,
'created': '2022-12-06T12:48:13.212965',
'description': 'This is a second sample resource for demonstration purposes.',
'format': 'SVG',
'hash': '',
'id': 'd28524d2-49ca-4c9d-a4dd-4b19bb03db35',
'last_modified': None,
'metadata_modified': '2022-12-06T12:48:13.207682',
'mimetype': 'image/svg+xml',
'mimetype_inner': None,
'name': 'TIB logo - Uploaded',
'package_id': 'ab2c9d5d-0109-4c3a-a61a-34d0a1dd022b',
'position': 1,
'resource_type': None,
'size': None,
'state': 'active',
'url': 'http://localhost:5000/dataset/ab2c9d5d-0109-4c3a-a61a-34d0a1dd022b/resource/d28524d2-49ca-4c9d-a4dd-4b19bb03db35/download/tib-full.svg',
'url_type': 'upload'}],
'tags': []},
```

This is the response above:


```

'name': 'TIB logo - Uploaded',
'package_id': 'ab2c9d5d-0109-4c3a-a61a-34d0a1dd022b',
'position': 1,
'resource_type': None,
'size': None,
'state': 'active',
'url': 'http://localhost:5000/dataset/ab2c9d5d-0109-4c3a-a61a-34d0a1dd022b/resource/ab2c9d5d-0109-4c3a-a61a-34d0a1dd022b',
'url_type': 'upload'},
'tags': [],
'groups': [],
'relationships_as_subject': [],
'relationships_as_object': [],
'doi': '10.23680/hzpdhljp',
'doi_status': False,
'domain': 'localhost:5000',
'doi_date_published': None,
'doi_publisher': 'TIB'}

```

All metadata is contained in the dictionary, including Resource's metadata.

In [43]:

```

import pprint

# Amount of resources
print("Number of resources: ", dataset_result['num_resources']).

# Resource 1 metadata
print("Resource 1 metadata:")
pprint.pprint(dataset_result['resources'][0]).

# Resource 2 metadata
print("Resource 2 metadata:")
pprint.pprint(dataset_result['resources'][1]).

```

```

Number of resources: 2
Resource 1 metadata:
{'cache_last_updated': None,
 'cache_url': None,
 'created': '2022-12-06T12:35:39.686019',
 'description': 'This is a sample resource for demonstration purposes.',
 'format': 'SVG',
 'hash': '',
 'id': '26bd95b3-48d0-46ed-9cd5-40316b5e5110',
 'last_modified': None,
 'metadata_modified': '2022-12-06T12:35:39.682249',
 'mimetype': 'image/svg+xml',
 'mimetype_inner': None,
 'name': 'TIB logo',
 'package_id': 'ab2c9d5d-0109-4c3a-a61a-34d0a1dd022b',
 'position': 0,
 'resource_type': None,
 'size': None,
 'state': 'active',
 'url': 'https://www.tib.eu/typo3conf/ext/tib_tpl_bootstrap/Resources/Public/gfx/logos/tib-full.svg',
 'url_type': ''}
Resource 2 metadata:
{'cache_last_updated': None,

```

```
'cache_url': None,  
'created': '2022-12-06T12:48:13.212965',  
'description': 'This is a second sample resource for demonstration purposes.',  
'format': 'SVG',  
'hash': '',  
'id': 'd28524d2-49ca-4c9d-a4dd-4b19bb03db35',  
'last_modified': None,  
'metadata_modified': '2022-12-06T12:48:13.207682',  
'mimetype': 'image/svg+xml',  
'mimetype_inner': None,  
'name': 'TIB logo - Uploaded',  
'package_id': 'ab2c9d5d-0109-4c3a-a61a-34d0a1dd022b',  
'position': 1,  
'resource_type': None,  
'size': None,  
'state': 'active',  
'url': 'http://localhost:5000/dataset/ab2c9d5d-0109-4c3a-a61a-34d0a1dd022b/resource/  
d28524d2-49ca-4c9d-a4dd-4b19bb03db35/download/tib-full.svg',
```

Resource's URLs

In [44]:

```
# Resource 1 metadata  
print("Resource 1 URL:")  
print(dataset_result['resources'][0]['url'])  
  
# Resource 2 metadata  
print("Resource 2 URL:")  
print(dataset_result['resources'][1]['url'])
```

Resource 1 URL:

https://www.tib.eu/typo3conf/ext/tib_tpl_bootstrap/Resources/Public/gfx/logos/tib-full.svg

Resource 2 URL:

<http://localhost:5000/dataset/ab2c9d5d-0109-4c3a-a61a-34d0a1dd022b/resource/d28524d2-49ca-4c9d-a4dd-4b19bb03db35/download/tib-full.svg>

In [45]:

```
# Here you have the code above selectable!  
"""  
params = {"id": dataset_ID}  
user_token = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJxUkdrRExJdy00VU5CVTdJbl'  
  
try:  
    response = requests.get(API_URL, params = params, headers={'Authorization': user  
except requests.exceptions.RequestException as e:  
    print("ERROR ACCESSING API: ", API_URL, e.__str__())  
"""  
"""  
"""
```

Out[45]:

```
''
```