

---

**Graviti**

**Graviti**

**Sep 27, 2022**



# GETTING STARTED

<b>1</b>	<b>What can Graviti SDK do?</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	10 Minutes to Graviti . . . . .	3
1.3	Upload Dataset . . . . .	6
1.4	Dataset Management . . . . .	8
1.5	Version Control . . . . .	9
1.6	Sheet Management . . . . .	17
1.7	graviti . . . . .	39
	<b>Python Module Index</b>	<b>199</b>
	<b>Index</b>	<b>201</b>



As a platform for unstructured data management, [Graviti](#) provides services in data hosting, version control, data visualization, and collaboration. Users can also integrate Graviti Data Platform into your own pipeline using developer tools.



## WHAT CAN GRAVITI SDK DO?

Graviti Python SDK is a python library to access Graviti workspace and manage datasets. It provides a pythonic way to access datasets by Graviti OpenAPI.

### 1.1 Installation

Graviti SDK can be installed with pip or from source.

#### 1.1.1 Pip

To install Graviti SDK with pip, run the following command:

```
$ pip3 install graviti
```

#### 1.1.2 Source

To install Graviti SDK from source, clone the repository from [github](https://github.com/Graviti-AI/graviti-python-sdk):

```
$ git clone https://github.com/Graviti-AI/graviti-python-sdk.git
$ cd graviti-python-sdk
$ pip install -e .
```

### 1.2 10 Minutes to Graviti

This is a simple introductory tutorial for beginners.

#### 1.2.1 Get an AccessKey

Before using Graviti SDK, please visit [Graviti Developer Tools](#) to get an AccessKey.

## 1.2.2 Dataset Preparation

This step is only for users who do not have datasets in their workspace. By running the code below, users can create a very simple dataset to experience Graviti SDK.

```
from graviti import DataFrame, Workspace
import graviti.portex as pt

ws = Workspace(f"{YOUR_ACCESSKEY}")
dataset = ws.datasets.create("Graviti-dataset-demo")

std = pt.build_package("https://github.com/Project-OpenBytes/portex-standard", "main")
box2ds = std.label.Box2D(
    categories=["boat", "car"],
    attributes={
        "difficult": pt.boolean(),
        "occluded": pt.boolean(),
    },
)
schema = pt.record(
    {
        "filename": pt.string(),
        "box2ds": pt.array(box2ds),
    }
)

filenames = ["a.jpg", "b.jpg", "c.jpg"]
data = []
for filename in filenames:
    row_data = {
        "filename": filename,
        "box2ds": [
            {
                "xmin": 1,
                "ymin": 1,
                "xmax": 4,
                "ymax": 5,
                "category": "boat",
                "attribute": {
                    "difficult": False,
                    "occluded": False,
                },
            },
        ],
    }
    data.append(row_data)
dataset["train"] = DataFrame(data=data, schema=schema)
dataset.commit("Commit-1")
```



### 1.2.3 Get a Dataset

Workspace initialization:

```
from graviti import Workspace
ws = Workspace(f"{YOUR_ACCESSKEY}")
```

List datasets on the workspace:

```
>>> ws.datasets.list()
LazyPagingList [
  Dataset("graviti-example/Graviti-dataset-demo")
]
```

Get one dataset:

```
>>> dataset = ws.datasets.get("Graviti-dataset-demo")
>>> dataset
Dataset("graviti-example/Graviti-dataset-demo")(
  (alias): '',
  (default_branch): 'main',
  (created_at): 2022-07-20 04:22:35+00:00,
  (updated_at): 2022-07-20 04:23:45+00:00,
  (is_public): False,
  (config): 'AmazonS3-us-west-1'
)
```

### 1.2.4 Get a Sheet

```
>>> dataset["train"]
  filename  box2ds
0  a.jpg    DataFrame(1, 6)
1  b.jpg    DataFrame(1, 6)
2  c.jpg    DataFrame(1, 6)
```

### 1.2.5 Get the Data

Get the DataFrame:

```
>>> df = dataset["train"]
>>> df
  filename  box2ds
0  a.jpg    DataFrame(1, 6)
1  b.jpg    DataFrame(1, 6)
2  c.jpg    DataFrame(1, 6)
```

View the schema of the sheet:

```
>>> df.schema
record(
  fields={
```

(continues on next page)

(continued from previous page)

```

'filename': string(),
'box2ds': array(
    items=label.Box2D(
        coords=float32(),
        categories=['boat', 'car'],
        attributes={
            'difficult': boolean(),
            'occluded': boolean(),
        },
    ),
),
},
)

```

Get the data by rows or columns:

```

>>> df.loc[0]
filename    a.jpg
box2ds      DataFrame(1, 6)

```

```

>>> df["box2ds"]
0    DataFrame(1, 6)
1    DataFrame(1, 6)
2    DataFrame(1, 6)

```

```

>>> df.loc[0]["box2ds"]
   xmin  ymin  xmax  ymax  category  attribute  difficult  occluded
0   1.0   1.0   4.0   5.0    boat      False      False      False

```

```

>>> df["box2ds"][0]
   xmin  ymin  xmax  ymax  category  attribute  difficult  occluded
0   1.0   1.0   4.0   5.0    boat      False      False      False

```

## 1.3 Upload Dataset

This is a simple guide to uploading a dataset.

### 1.3.1 Create or Get a Dataset

Create a new dataset:

```

from graviti import Workspace

ws = Workspace(f"{YOUR_ACCESSKEY}")
dataset = ws.datasets.create("Graviti-dataset-demo")

```

Or get an existing dataset:

```
dataset = ws.datasets.get("Graviti-dataset-demo")
```

### 1.3.2 DataFrame Preparation

Users need to organize the data into *DataFrame* format with *Schema*. In addition, SDK also supports uploading different kinds of *Binary Files*, such as image, audio, etc.

```
from graviti import DataFrame
from graviti.file import Image
import graviti.portex as pt

std = pt.build_package("https://github.com/Project-OpenBytes/portex-standard", "main")
box2ds = std.label.Box2D(
    categories=["boat", "car"],
    attributes={
        "difficult": pt.boolean(),
        "occluded": pt.boolean(),
    },
)
schema = pt.record(
    {
        "filename": pt.string(),
        "image": std.file.Image(),
        "box2ds": pt.array(box2ds),
    }
)

filenames = ["a.jpg", "b.jpg", "c.jpg"]
data = []
for filename in filenames:
    row_data = {
        "filename": filename,
        "image": Image(f"PATH/TO/{filename}"),
        "box2ds": [
            {
                "xmin": 1,
                "ymin": 1,
                "xmax": 4,
                "ymax": 5,
                "category": "boat",
                "attribute": {
                    "difficult": False,
                    "occluded": False,
                },
            },
        ],
    }
    data.append(row_data)
df = DataFrame(data=data, schema=schema)
```

### 1.3.3 Upload and Commit

Create or modify the sheet by uploading the dataframe, more details about sheet can be viewed in [Sheet Management](#).

```
dataset["train"] = df
dataset.commit("Commit-1")
```

The method `commit()` actually includes creating, uploading and committing the draft.

Interested users can learn more about dataset version management in [Version Control](#), which can also help troubleshooting when uploads fail.

## 1.4 Dataset Management

*Dataset* is the most basic concept in Graviti SDK. Almost all operations require a dataset first. Of course, it is necessary to initialize a *Workspace* before managing the dataset:

```
from graviti import Workspace
ws = Workspace(f"{YOUR_ACCESSKEY}")
```

### 1.4.1 Create a Dataset

SDK provides method `create()` to support creating a dataset based on the given name:

```
ws.datasets.create(f"{DATASET_NAME}")
```

In addition to name, alias and storage config can also be specified:

```
ws.datasets.create(f"{DATASET_NAME}", f"{DATASET_ALIAS}", f"{STORAGE_CONFIG}")
```

---

**Note:** Unlike the operation on the web page, here SDK will not create an empty draft after creating the dataset.

---

### 1.4.2 List Datasets

SDK provides method `list()` to support listing datasets on the workspace:

```
ws.datasets.list()
```

### 1.4.3 Get a Dataset

SDK provides method `get()` to support getting a dataset by name:

```
ws.datasets.get(f"{DATASET_NAME}")
```

### 1.4.4 Delete a Dataset

SDK provides method `delete()` to support deleting a dataset by name:

```
ws.datasets.delete(f"{DATASET_NAME}")
```

### 1.4.5 Edit the Dataset

SDK provides method `edit()` to support changing the name, alias and default branch of the dataset.

```
dataset = ws.datasets.get(f"{DATASET_NAME}")
dataset.edit(
    name=f"{NEW_DATASET_NAME}",
    alias=f"{NEW_ALIAS}",
    default_branch=f"{NEW_DEFAULT_BRANCH}"
)
```

## 1.5 Version Control

Version control is one of the important features of Graviti. It can help teams or individual users develop datasets in parallel and trace the history of the data.

### 1.5.1 HEAD

SDK supports viewing the version of the dataset through property `HEAD`:

```
>>> from graviti import Workspace
>>> ws = Workspace(f"{YOUR_ACCESSKEY}")
>>> dataset = ws.datasets.get("Graviti-dataset-demo")
>>> dataset.HEAD
Branch("main")(
  (commit_id): '47293b32f28c4008bc0f25b847b97d6f',
  (parent): None,
  (title): 'Commit-1',
  (committer): 'graviti-example',
  (committed_at): 2022-07-20 04:23:45+00:00
)
>>> dataset.HEAD.name
"main"
>>> dataset.HEAD.commit_id
"47293b32f28c4008bc0f25b847b97d6f"
```

## 1.5.2 Checkout

SDK provides method `checkout()` to support switching the version of the dataset. This method will modify the HEAD of the dataset and discard the previous modification on the dataset.

Checkout the Branch("dev"):

```
>>> from graviti import Workspace
>>> ws = Workspace(f"{YOUR_ACCESSKEY}")
>>> dataset = ws.datasets.get("Graviti-dataset-demo")
>>> dataset.HEAD
Branch("main")(
  (commit_id): '47293b32f28c4008bc0f25b847b97d6f',
  (parent): None,
  (title): 'Commit-1',
  (committer): 'graviti-example',
  (committed_at): 2022-07-20 04:23:45+00:00
)
>>> dataset.checkout("dev")
Branch("dev")(
  (commit_id): '781007a41d1641859c87cb00f8e32bf3',
  (parent): Commit("3db73ac2876a42c0bf43a0489ce1756a"),
  (title): 'commit-5',
  (committer): 'graviti-example',
  (committed_at): 2022-07-19 04:23:45+00:00
)
```

Checkout the Tag("v1.0"):

```
>>> dataset.checkout("v1.0")
Tag("v1.0")(
  (commit_id): '2cd44960e0bf486c950536f7eeebc482',
  (parent): Commit("e8dc893eb2344b9a98bddce71a1c0eab"),
  (title): 'commit-7',
  (committer): 'graviti-example',
  (committed_at): 2022-07-19 04:25:45+00:00
)
```

Checkout the Commit("2cd4496"):

```
>>> dataset.checkout("2cd44960e0bf486c950536f7eeebc482")
Commit("2cd44960e0bf486c950536f7eeebc482")(
  (parent): Commit("e8dc893eb2344b9a98bddce71a1c0eab"),
  (title): 'commit-7',
  (committer): 'graviti-example',
  (committed_at): 2022-07-19 04:25:45+00:00
)
```

### 1.5.3 More Details

More details about the version control methods are as follows:

#### Commit

*Commit* is the basic element of Graviti version control system. Each commit of the dataset represents a **read-only** version.

The following sections will introduce the operations related to commits in the SDK. First of all, it is necessary to get a dataset:

```
from graviti import Workspace

ws = Workspace(f"{YOUR_ACCESSKEY}")
dataset = ws.datasets.get(f"{DATASET_NAME}")
```

#### List Commits

SDK provides method *list()* to support listing commits preceding the given revision. The revision can be one commit ID:

```
dataset.commits.list(f"{COMMIT_ID}")
```

Or the branch name:

```
dataset.commits.list(f"{BRANCH_NAME}")
```

Or the tag name:

```
dataset.commits.list(f"{TAG_NAME}")
```

If no revision is specified, all commits preceding the current commit will be returned:

```
dataset.commits.list()
```

#### Get a Commit

SDK provides method *get()* to support getting a commit by revision. The revision can be one commit ID:

```
dataset.commits.get(f"{COMMIT_ID}")
```

Or the branch name:

```
dataset.commits.get(f"{BRANCH_NAME}")
```

Or the tag name:

```
dataset.commits.get(f"{TAG_NAME}")
```

If no revision is specified, the current commit of dataset will be returned:

```
dataset.commits.get()
```

### Checkout Commit

SDK provides method `checkout()` to support switching the version of the dataset by commits:

```
dataset.checkout(f"{COMMIT_ID}")
```

### Branch

Each dataset is created with a default branch `main`. When getting a dataset through the SDK, its version is the latest commit of the default branch.

The following sections will introduce the operations and precautions related to branches in the SDK. Of course, it is necessary to get a dataset first:

```
from graviti import Workspace

ws = Workspace(f"{YOUR_ACCESSKEY}")
dataset = ws.datasets.get(f"{DATASET_NAME}")
```

---

**Note:** In most cases, a branch can be thought of as a named commit. But if there is a dataset that has just been created and has no commit history, the commit id of its default branch is `None`.

---

### Create a Branch

SDK provides method `create()` to support creating a branch based on a revision. The revision can be one commit ID:

```
dataset.branches.create(f"{BRANCH_NAME}", f"{COMMIT_ID}")
```

The revision can also be the branch name. In this situation, the new branch will be created based on the latest commit of the source branch:

```
dataset.branches.create(f"{BRANCH_NAME}", f"{SOURCE_BRANCH_NAME}")
```

The revision can also be the tag name:

```
dataset.branches.create(f"{BRANCH_NAME}", f"{TAG_NAME}")
```

If no revision is specified, the created branch will be based on the current commit of the dataset:

```
dataset.branches.create(f"{BRANCH_NAME}")
```



## List Branches

SDK provides method `list()` to support listing branches:

```
dataset.branches.list()
```

## Get a Branch

SDK provides method `get()` to support getting a branch by name:

```
dataset.branches.get(f"{BRANCH_NAME}")
```

## Delete a Branch

SDK provides method `delete()` to support deleting a branch by name:

```
dataset.branches.delete(f"{BRANCH_NAME}")
```

## Checkout Branch

SDK provides method `checkout()` to support switching the version of the dataset by branches:

```
dataset.checkout(f"{BRANCH_NAME}")
dataset.HEAD # Check whether the dataset version is correct.
```

In addition, this checkout method is often used to update the version of the local dataset without getting the dataset again, for example:

```
# Other users committed a draft on the Branch("main").
dataset.checkout("main") # Update the version of the dataset.
```

## Draft

All operations related to modifying data require a draft first. *Draft* can be viewed as a writable dataset in SDK.

The following sections will introduce the operations and precautions related to drafts in the SDK. Of course, it is necessary to get a dataset first:

```
from graviti import Workspace

ws = Workspace(f"{YOUR_ACCESSKEY}")
dataset = ws.datasets.get(f"{DATASET_NAME}")
```

### Create a Draft

SDK provides method `create()` to support creating a draft based on a branch:

```
dataset.drafts.create(f"{DRAFT_TITLE}", branch=f"{BRANCH_NAME}")
```

If no branch is specified, the created draft will be based on the current branch of the dataset:

```
dataset.drafts.create(f"{DRAFT_TITLE}")
```

In addition to title, it is also allowed to add description to the draft:

```
dataset.drafts.create(f"{DRAFT_TITLE}", f"{DRAFT_DESCRIPTION}")
```

---

**Important:** Graviti use number, not title, to uniquely identify drafts.

---

### List Drafts

SDK provides method `list()` to support listing drafts. Drafts can be filtered by state including OPEN, COMMITTED, CLOSED and ALL:

```
dataset.drafts.list(f"{STATE}")
```

Drafts can also be filtered by the branch name:

```
dataset.drafts.list(branch=f"{BRANCH_NAME}")
```

If neither the state nor the branch name is specified, then all open drafts on all branches will be returned:

```
dataset.drafts.list()
```

### Get a Draft

SDK provides method `get()` to support getting a draft by number:

```
dataset.drafts.get(DRAFT_NUMBER)
```

### Edit the Draft

SDK provides method `edit()` to support changing the title and description of the draft:

```
draft = dataset.drafts.get(DRAFT_NUMBER)
draft.edit(f"{NEW_TITLE}", f"{NEW_DESCRIPTION}")
```

## Upload the Draft

SDK provides method `upload()` to support uploading the local draft to Graviti. This step is essential if the user wants to save changes to the sheet of the dataset.

```
draft = dataset.drafts.get(DRAFT_NUMBER)
del draft["train"]
draft.upload()
```

**Note:** SDK supports specifying the max workers in multi-thread upload. The default is 8.

## Commit the Draft

SDK provides method `commit()` to support committing a draft. This action means that a new commit will be created and all the changes from the draft will be saved into this commit. Furthermore, it is not allowed to read or upload data on a committed draft:

```
draft = dataset.drafts.get(DRAFT_NUMBER)
draft.commit(f"{COMMIT_TITLE}")
```

In addition to title, it is also allowed to add description to the commit:

```
draft.commit(f"{COMMIT_TITLE}", f"{COMMIT_DESCRIPTION}")
```

In this case, SDK will automatically update the version of the dataset after committing the draft. And all modifications on the dataset will be lost.

```
>>> dataset = ws.datasets.get("Graviti-dataset-demo")
>>> dataset.HEAD.name # The version of the dataset is Branch("main").
"main"
>>> dataset.HEAD.commit_id
"524d110ecae7474eaec9471f4a6c28b0"
>>> draft = dataset.drafts.create("draft-4", branch="dev")
>>> draft.commit("commit-4")
Branch("dev")(
  (commit_id): '3db73ac2876a42c0bf43a0489ce1756a',
  (parent): Commit('1b21a40f03ab4cec814ec47ee0d10b24'),
  (title): 'commit-4',
  (committer): 'graviti-example',
  (committed_at): 2022-07-21 04:23:45+00:00
)
>>> dataset.HEAD.name # The version of the dataset has been updated to Branch("dev").
"dev"
>>> dataset.HEAD.commit_id
"3db73ac2876a42c0bf43a0489ce1756a"
```

Users can avoid the automatic update by setting `update_dataset_head` to `False`:

```
>>> dataset = ws.datasets.get("Graviti-dataset-demo")
>>> dataset.HEAD.name # The version of the dataset is Branch("main").
"main"
```

(continues on next page)

(continued from previous page)

```

>>> dataset.HEAD.commit_id
"524d110ecae7474eac9471f4a6c28b0"
>>> draft = dataset.drafts.create("draft-5", branch="dev")
>>> draft.commit("commit-5", update_dataset_head=False)
Branch("dev")(
  (commit_id): '781007a41d1641859c87cb00f8e32bf3',
  (parent): Commit("3db73ac2876a42c0bf43a0489ce1756a"),
  (title): 'commit-5',
  (committer): 'graviti-example',
  (committed_at): 2022-07-21 04:24:45+00:00
)
>>> dataset.HEAD.name # The version of the dataset has not been updated.
"main"
>>> dataset.HEAD.commit_id
"524d110ecae7474eac9471f4a6c28b0"

```

## Close the Draft

SDK provides method `close()` to support closing a draft: This action means that all changes made on this draft will be dropped. And the closed draft cannot be reopened. Furthermore, it is not allowed to read or upload data on a closed draft:

```

draft = dataset.drafts.get(DRAFT_NUMBER)
draft.close()

```

## Tag

Graviti supports tagging specific commits in a dataset's history as being important, for example, to mark release revisions (v1.0, v2.0 and so on).

Before operating tags, a dataset with existing commits is needed:

```

from graviti import Workspace

ws = Workspace(f"{YOUR_ACCESSKEY}")
dataset = ws.datasets.get(f"{DATASET_NAME}")

```

## Create a Tag

SDK provides method `create()` to support creating a tag based on a revision. The revision can be one commit ID:

```
dataset.tags.create(f"{TAG_NAME}", f"{COMMIT_ID}")
```

The revision can also be the branch name. In this situation, the tag will be created based on the latest commit of the branch:

```
dataset.tags.create(f"{TAG_NAME}", f"{BRANCH_NAME}")
```

The revision can also be the tag name. SDK supports creating multiple tags based on the same commit:

```
dataset.tags.create(f"{TAG_NAME}", f"{SOURCE_TAG_NAME}")
```

If no tag is specified, the created tag will be based on the current commit of the dataset:

```
dataset.tags.create(f"{TAG_NAME}")
```

## List Tags

SDK provides method `list()` to support listing tags:

```
dataset.tags.list()
```

## Get a Tag

SDK provides method `get()` to support getting a tag by name:

```
dataset.tags.get(f"{TAG_NAME}")
```

## Delete a Tag

SDK provides method `delete()` to support deleting a tag by name:

```
dataset.tags.delete(f"{TAG_NAME}")
```

## Checkout Tag

SDK provides method `checkout()` to support switching the version of the dataset by tags:

```
dataset.checkout(f"{TAG_NAME}")
# Check whether the dataset version is correct.
dataset.HEAD
```

# 1.6 Sheet Management

In the Graviti SDK, Sheet and DataFrame are interpretations of the same thing at different levels.

Sheet refers to the form of data organization that is one level lower than the dataset. One dataset can have many different sheets, such as train, test, or frame-by-frame pictures from different videos. Each sheet has its own schema.

Graviti SDK organizes the data of a sheet into a DataFrame format, which makes it more convenient and intuitive to get and modify the data.

More details about the DataFrame, Binary Files, Schema and Search are as follows:

## 1.6.1 DataFrame

*DataFrame* is an integrated data structure with an easy-to-use API for simplifying data processing in Dataset. A Graviti DataFrame contains 2-dimensional tabular data and a Protex schema describing the names and types of each column.

### Initialize a DataFrame

Initialize a DataFrame from a list of dicts:

```
>>> from graviti import DataFrame
>>> data = [
...     {"filename": "a.jpg"},
...     {"filename": "b.jpg"},
...     {"filename": "c.jpg"},
... ]
>>> df = DataFrame(data)
>>> df
  filename
0  a.jpg
1  b.jpg
2  c.jpg
```

Initialize a DataFrame with multi-level column names:

```
>>> from graviti import DataFrame
>>> data = [
...     {"attribute": {"weather": "sunny", "color": "red"}},
...     {"attribute": {"weather": "rainy", "color": "black"}},
...     {"attribute": {"weather": "sunny", "color": "white"}},
... ]
>>> df = DataFrame(data)
>>> df
  attribute
  color    weather
0  red      sunny
1  black    rainy
2  white    sunny
```

Initialize a DataFrame with nested DataFrame construction:

```
>>> from graviti import DataFrame
>>> data = [
...     {"points": [{"xmin": 1, "ymin": 3}, {"xmin": 5, "ymin": 8}]},
...     {"points": [{"xmin": 6, "ymin": 10}]},
...     {"points": [{"xmin": 1, "ymin": 3}, {"xmin": 5, "ymin": 8}, {"xmin": 1, "ymin": 9}
...     ]},
... ]
>>> df = DataFrame(data)
>>> df
  points
0  DataFrame(2, 2)
1  DataFrame(1, 2)
2  DataFrame(3, 2)
```

(continues on next page)

(continued from previous page)

```
>>> df["points"][0]
      xmin  ymin
0      1     3
1      5     8
```

## Read the DataFrame

Read data by row:

```
df.loc[0]
```

Read data by column:

```
df[f"{COLUMN_NAME}"]
```

Read a DataFrame cell:

```
df.loc[0][f"{COLUMN_NAME}"]
df[f"{COLUMN_NAME}"][0]
```

## Edit the DataFrame

### Edit Rows

Edit one row:

```
df.loc[0] = {"filename": "d.jpg"}
```

Edit multiple rows:

```
df.loc[0:2] = [{"filename": "d.jpg"}, {"filename": "e.jpg"}]
```

### Edit the Items of Column

Edit one item:

```
df[f"{COLUMN_NAME}"][0] = "d.jpg"
```

Edit multiple items:

```
df[f"{COLUMN_NAME}"][0:2] = ["d.jpg", "e.jpg"]
```

## Delete Rows

Delete one row:

```
del df.loc[0]
```

Delete multiple rows:

```
del df.loc[0:2]
```

## Extend Rows

DataFrame supports method `extend()`.

Extend rows to the end of the DataFrame:

```
df.extend([{"filename": "a.jpg"}])
```

Extend another Dataframe to the end of the DataFrame:

```
df1 = DataFrame([{"filename": "a.jpg"}])  
df.extend(df1)
```

## Add Columns

DataFrame supports adding columns by `setitem`:

```
>>> from graviti import DataFrame  
>>> data = [  
...     {"filename": "a.jpg"},  
...     {"filename": "b.jpg"},  
...     {"filename": "c.jpg"},  
... ]  
>>> df = DataFrame(data)  
>>> df  
  filename  
0  a.jpg  
1  b.jpg  
2  c.jpg  
>>> df["caption"] = ["a", "b", "c"]  
>>> df  
  filename  caption  
0  a.jpg      a  
1  b.jpg      b  
2  c.jpg      c  
>>> df.schema  
record(  
  fields={  
    'filename': string(),  
    'caption': string(),  
  },  
)
```



The above example shows adding a column of data with no specified type, and the schema of the column will be inferred. In this case, the column schema can only be Portex *Primitive Types*.

If specific Portex type is required, please add a Series as the column to the DataFrame.

```
>>> from graviti import DataFrame, Series
>>> data = [
...     {"filename": "a.jpg"},
...     {"filename": "b.jpg"},
...     {"filename": "c.jpg"},
... ]
>>> df = DataFrame(data)
>>> df
  filename
0  a.jpg
1  b.jpg
2  c.jpg
>>> df["category"] = Series(["cat", "dog", "cat"], pt.enum(["cat", "dog"]))
>>> df
  filename  category
0  a.jpg      cat
1  b.jpg      dog
2  c.jpg      cat
>>> df.schema
record(
  fields={
    'filename': string(),
    'category': enum(
      values=['cat', 'dog'],
    ),
  },
)
```

Note that not all DataFrame can be modified. Only if the fields of the schema are from given arguments, the DataFrame can be changed, like the above example. If the fields are defined in a template, the DataFrame cannot be changed, and `TypeError` will be raised:

```
>>> from graviti import DataFrame, Workspace
>>> import graviti.portex as pt

>>> std = pt.build_package("https://github.com/Project-OpenBytes/portex-standard", "main
↪")
>>> box2ds = std.label.Box2D(
...     categories=["boat", "car"],
...     attributes={
...         "difficult": pt.boolean(),
...         "occluded": pt.boolean(),
...     },
... )
>>> df = DataFrame(
...     [
...         {
...             "xmin": 1,
...             "ymin": 1,
```

(continues on next page)

(continued from previous page)

```

...         "xmax": 4,
...         "ymax": 5,
...         "category": "boat",
...         "attribute": {
...             "difficult": False,
...             "occluded": False,
...         },
...     },
... ],
...     schema=box2ds
... )
>>> df
   xmin  ymin  xmax  ymax  category  attribute
      0  1.0  1.0  4.0  5.0    boat      False      False
>>> df["caption"] = ["a"]
TypeError: Cannot set item 'caption' in ImmutableFields

```

## 1.6.2 Binary Files

Graviti SDK use the *File* and *RemoteFile* to represent a specific file.

In addition, SDK also provides several commonly used file formats, including *Image*, *Audio* and *PointCloud*.

### File

SDK supports all various binary files including video files and text files by *File*.

Load the local text files to DataFrame:

```

import graviti.portex as pt
from graviti import DataFrame
from graviti.file import File

std = pt.build_package("https://github.com/Project-OpenBytes/portex-standard", "main")
schema = pt.record(
    {
        "filename": pt.string(),
        "file": std.file.File(),
    }
)

data = [
    {
        "filename": "EXAMPLE1.txt",
        "file": File("PATH/TO/YOUR/EXAMPLE1.txt")
    },
    {
        "filename": "EXAMPLE2.txt",
        "file": File("PATH/TO/YOUR/EXAMPLE2.txt")
    },
]

```

(continues on next page)

(continued from previous page)

```
]

df = DataFrame(data, schema)
```

Read the text in DataFrame:

```
text = df["text"][0]
with text.open() as fp:
    fp.read().decode("utf-8")
```

For all binary files, SDK supports viewing their basic information, including extension, size and checksum:

```
text.extension
text.size
text.get_checksum()
```

## Image

Load the local image into DataFrame:

```
import graviti.portex as pt
from graviti import DataFrame
from graviti.file import Image

std = pt.build_package("https://github.com/Project-OpenBytes/portex-standard", "main")
schema = pt.record(
    {
        "filename": pt.string(),
        "image": std.file.Image(),
    }
)

data = [
    {
        "filename": "EXAMPLE1.png",
        "image": Image("PATH/TO/YOUR/EXAMPLE1.png")
    },
    {
        "filename": "EXAMPLE2.png",
        "image": Image("PATH/TO/YOUR/EXAMPLE2.png")
    },
]

df = DataFrame(data, schema)
```

Read the image in DataFrame:

```
import PIL

image = df["image"][0]
with image.open() as fp:
    PIL.Image.open(fp)
```

For image files, SDK supports viewing their height and width:

```
image.height  
image.width
```

### Audio

Load the local audio into DataFrame:

```
import graviti.portex as pt  
from graviti import DataFrame  
from graviti.file import Audio  
  
std = pt.build_package("https://github.com/Project-OpenBytes/portex-standard", "main")  
schema = pt.record(  
    {  
        "filename": pt.string(),  
        "audio": std.file.Audio(),  
    }  
)  
  
data = [  
    {  
        "filename": "EXAMPLE1.wav",  
        "audio": Audio("PATH/TO/YOUR/EXAMPLE1.wav")  
    },  
    {  
        "filename": "EXAMPLE2.wav",  
        "audio": Audio("PATH/TO/YOUR/EXAMPLE2.wav")  
    },  
]  
  
df = DataFrame(data, schema)
```

Read the audio in DataFrame:

```
audio = df["audio"][0]  
with audio.open() as fp:  
    fp.read()
```

### Point Cloud

Load the local point\_cloud into DataFrame:

```
import graviti.portex as pt  
from graviti import DataFrame  
from graviti.file import PointCloud  
  
std = pt.build_package("https://github.com/Project-OpenBytes/portex-standard", "main")  
schema = pt.record(  
    {  
        "filename": pt.string(),  
    }
```

(continues on next page)

(continued from previous page)

```

        "point_cloud": std.file.PointCloud(),
    }
)

data = [
    {
        "filename": "EXAMPLE1.pcd",
        "point_cloud": PointCloud("PATH/TO/YOUR/EXAMPLE1.pcd")
    },
    {
        "filename": "EXAMPLE2.pcd",
        "point_cloud": PointCloud("PATH/TO/YOUR/EXAMPLE2.pcd")
    },
]

df = DataFrame(data, schema)

```

Read the point\_cloud in DataFrame:

```

point_cloud = df["point_cloud"][0]
with point_cloud.open() as fp:
    fp.read()

```

### 1.6.3 Schema

Each sheet has a record type schema to describe the name and the type of each column. Graviti use [Portex](#) schema language to define the schema, please refer to its documentation for the syntax.

Graviti SDK supports the python interaction with the Portex schema.

#### Primitive Types

Graviti SDK provides classes to initialize Portex primitive types:

boolean

binary

string

int32

int64

float32

float64

```

>>> import graviti.portex as pt
>>> pt.boolean()
boolean()

```

```
>>> import graviti.portex as pt
>>> pt.binary()
binary()
```

```
>>> import graviti.portex as pt
>>> pt.string()
string()
```

```
>>> import graviti.portex as pt
>>> pt.int32()
int32()
```

```
>>> import graviti.portex as pt
>>> pt.int64()
int64()
```

```
>>> import graviti.portex as pt
>>> pt.float32()
float32()
```

```
>>> import graviti.portex as pt
>>> pt.float64()
float64()
```

## Complex Types

Graviti SDK provides classes to initialize Portex complex types.

### enum

An enum type can be created by giving the values:

```
>>> import graviti.portex as pt
>>> enum = pt.enum(values=["a", "b", "c"])
>>> enum.values
['a', 'b', 'c']
```

### array

The array type can be created by giving the item type. Param `length` can be specified to fix the length of array.

```
>>> import graviti.portex as pt
>>> array = pt.array(pt.int32())
>>> array
array(
  items=int32(),
)
>>> array = pt.array(pt.boolean(), length=2)
```

(continues on next page)

(continued from previous page)

```
>>> array.items
boolean()
>>> array.length
2
```

## record

The record type can be created by giving the fields, including names and types. The record type is used to describe the name and type of each column of the tabular data, and all the primitive and complex types mentioned above can be used here for each column type.

The names and types can be accessed by `fields`, which acts like a dict whose key is the column name and the value is column type.

Init record with list

Init record with dict

```
>>> import graviti.portex as pt
>>> record = pt.record(
...     [
...         ("x", pt.int32()),
...         ("y", pt.int32()),
...         ("categories", pt.enum(values=["cat", "dog"]))
...     ]
... )
>>> record
record(
  fields={
    'x': int32(),
    'y': int32(),
    'categories': enum(
      values=['cat', 'dog'],
    ),
  },
)
>>> record.fields
{
  'x': int32(),
  'y': int32(),
  'categories': enum(
    values=['cat', 'dog'],
  ),
}
```

```
>>> import graviti.portex as pt
>>> record = pt.record(
...     {
...         "x": pt.int32(),
...         "y": pt.int32(),
...         "categories": pt.enum(values=["cat", "dog"]),
...     }
... )
```

(continues on next page)

(continued from previous page)

```
... )
>>> record
record(
  fields={
    'x': int32(),
    'y': int32(),
    'categories': enum(
      values=['cat', 'dog'],
    ),
  },
)
>>> record.fields
{
  'x': int32(),
  'y': int32(),
  'categories': enum(
    values=['cat', 'dog'],
  ),
}
```

## Template Type

The template type can be created by giving the parameters and the declaration. And the type can be instantiated by giving the arguments.

```
>>> import graviti.portex as pt
>>> vector_template = {
...     "type": "template",
...     "parameters": [
...         {
...             "name": "coords",
...             "default": {"type": "int32"},
...         },
...         {
...             "name": "labels",
...             "default": None,
...         },
...     ],
...     "declaration": {
...         "type": "record",
...         "fields": [
...             {
...                 "name": "x",
...                 "+": "$coords",
...             },
...             {
...                 "name": "y",
...                 "+": "$coords",
...             },
...             {
...                 "name": "label",
```

(continues on next page)



(continued from previous page)

```

...         "exist_if": "$labels",
...         "type": "enum",
...         "values": "$labels",
...     },
... ],
... },
... }
>>> Vector = pt.template.template("Vector", vector_template)
>>> Vector
<class 'graviti.portex.template.Vector'>
>>> vector = Vector(coords=pt.float32(), labels=["cat", "dog"])
>>> vector
Vector(
  coords=float32(),
  labels=['cat', 'dog'],
)

```

## Schema Files

Graviti SDK provides `read_yaml()` and `read_json()` to read the Portex type from a yaml or a json file.

YAML File

JSON File

Take the following `schema.yaml` file as an example:

```

---
type: record
fields:
  - name: filename
    type: string

  - name: category
    type: int32

  - name: attribute
    type: record
    fields:
      - name: weather
        type: enum
        values: ["sunny", "rainy", "windy"]

      - name: distorted
        type: boolean

```

```

>>> import graviti.portex as pt
>>> schema = pt.read_yaml("schema.yaml")
>>> schema
record(
  fields={
    'filename': string(),

```

(continues on next page)

(continued from previous page)

```

'category': int32(),
'attribute': record(
    fields={
        'weather': enum(
            values=['sunny', 'rainy', 'windy'],
        ),
        'distorted': boolean(),
    },
),
},
)

```

Take the following `schema.json` file as an example:

```

{
  "type": "record",
  "fields": [
    {
      "name": "filename",
      "type": "string"
    },
    {
      "name": "category",
      "type": "int32"
    },
    {
      "name": "attribute",
      "type": "record",
      "fields": [
        {
          "name": "weather",
          "type": "enum",
          "values": [
            "sunny",
            "rainy",
            "windy"
          ]
        },
        {
          "name": "distorted",
          "type": "boolean"
        }
      ]
    }
  ]
}

```

```

>>> import graviti.portex as pt
>>> schema = pt.read_json("schema.json")
>>> schema
record(
  fields={

```

(continues on next page)

(continued from previous page)

```

'filename': string(),
'category': int32(),
'attribute': record(
    fields={
        'weather': enum(
            values=['sunny', 'rainy', 'windy'],
        ),
        'distorted': boolean(),
    },
),
},
)

```

## Schema Package

Graviti SDK supports to use external packages defined under a repo. By giving the repo and revision, the package can be initialized and used locally.

SDK provides `build_package()` to build an external Portex type package from the repo. Take `standard` as an example, which is used as the standard external package by Graviti.

```

>>> import graviti.portex as pt
>>> std = pt.build_package("https://github.com/Project-OpenBytes/portex-standard", "main
↳ ")
Cloning repo 'https://github.com/Project-OpenBytes/portex-standard@main'
Cloned to '/tmp/portex/2a656e669aea0b88dca87784a3963215'
>>> std
ExternalPackage {
  'calibration.Intrinsic': <class 'graviti.portex.builder.calibration.Intrinsic'>,
  'calibration.Extrinsic': <class 'graviti.portex.builder.calibration.Extrinsic'>,
  'geometry.Vector3D': <class 'graviti.portex.builder.geometry.Vector3D'>,
  'geometry.Quaternion': <class 'graviti.portex.builder.geometry.Quaternion'>,
  'geometry.Keypoint2D': <class 'graviti.portex.builder.geometry.Keypoint2D'>,
  'geometry.Vector2D': <class 'graviti.portex.builder.geometry.Vector2D'>,
  'geometry.PointList2D': <class 'graviti.portex.builder.geometry.PointList2D'>,
  'label.file.SemanticMask': <class 'graviti.portex.builder.label.file.SemanticMask'>,
  'label.file.InstanceMask': <class 'graviti.portex.builder.label.file.InstanceMask'>,
  'label.file.RemoteInstanceMask': <class 'graviti.portex.builder.label.file.
↳ RemoteInstanceMask'>,
  'label.file.PanopticMask': <class 'graviti.portex.builder.label.file.PanopticMask'>,
  'label.file.RemoteSemanticMask': <class 'graviti.portex.builder.label.file.
↳ RemoteSemanticMask'>,
  'label.tensor.SemanticMask': <class 'graviti.portex.builder.label.tensor.SemanticMask'>
↳ ,
  'label.tensor.InstanceMask': <class 'graviti.portex.builder.label.tensor.InstanceMask'>
↳ ,
  ... (25 items are folded),
  'tensor.Image': <class 'graviti.portex.builder.tensor.Image'>
}
>>> box2d = std.label.Box2D(categories=["cat", "dog"])
>>> box2d

```

(continues on next page)

(continued from previous page)

```
label.Box2D(
    coords=float32(),
    categories=['cat', 'dog'],
)
```

**Note:** Using branch as the revision to build the external package is unstable, since the latest commit may change.

Tag name or commit ID as revision is recommended.

## Binary Files

SDK supports adding and uploading binary files, whose schema must be of type `file.File`, `file.Image`, `file.Audio` or `file.PointCloud` in [standard](#) package.

```
>>> import graviti.portex as pt
>>> std = pt.build_package("https://github.com/Project-OpenBytes/portex-standard", "main
↳")
Cloning repo 'https://github.com/Project-OpenBytes/portex-standard@main'
Cloned to '/tmp/portex/2a656e669aea0b88dca87784a3963215'
>>> record = pt.record(
...     {
...         "filename": pt.string(),
...         "image": std.file.Image,
...     }
... )
```

When using the record in the above example as the schema of a DataFrame, the column of “image” stores image files. Please see [Binary Files](#) for more details about data.

## Schema Methods

### Convert

PortexType provides methods to convert to or init from python object, json string and yaml string. Take the following schema as an example:

```
>>> import graviti.portex as pt
>>> schema = pt.record(
...     {
...         "x": pt.int32(),
...         "y": pt.int32(),
...         "categories": pt.enum(values=["cat", "dog"]),
...     }
... )
>>> schema
record(
  fields={
    'x': int32(),
    'y': int32(),
```

(continues on next page)

(continued from previous page)

```

    'categories': enum(
        values=['cat', 'dog'],
    ),
},
)

```

Python Object

JSON String

YAML String

```

>>> pyobj = schema.to_pyobj()
>>> pyobj
{'type': 'record',
 'fields': [{'name': 'x', 'type': 'int32'},
            {'name': 'y', 'type': 'int32'},
            {'name': 'categories', 'type': 'enum', 'values': ['cat', 'dog']}]}

>>> pt.PortexType.from_pyobj(pyobj)
record(
  fields={
    'x': int32(),
    'y': int32(),
    'categories': enum(
      values=['cat', 'dog'],
    ),
  },
)

```

```

>>> json_string = schema.to_json()
>>> json_string
'{"type": "record", "fields": [{"name": "x", "type": "int32"}, {"name": "y", "type":
↪ "int32"}, {"name": "categories", "type": "enum", "values": ["cat", "dog"]}]}

>>> pt.PortexType.from_json(json_string)
record(
  fields={
    'x': int32(),
    'y': int32(),
    'categories': enum(
      values=['cat', 'dog'],
    ),
  },
)

```

```

>>> yaml_string = schema.to_yaml()
>>> yaml_string
'type: record\nfields:\n- name: x\n  type: int32\n- name: y\n  type: int32\n- name: ↪
↪ categories\n  type: enum\n  values:\n    - cat\n    - dog\n'

>>> pt.PortexType.from_yaml(yaml_string)
record(

```

(continues on next page)

(continued from previous page)

```
fields={
  'x': int32(),
  'y': int32(),
  'categories': enum(
    values=['cat', 'dog'],
  ),
},
)
```

## Expand

For better comprehension and operations, SDK provides methods to expand external Portex type to builtin types:

```
>>> import graviti.portex as pt
>>> std = pt.build_package("https://github.com/Project-OpenBytes/portex-standard", "main
↪")
>>> box2d = std.label.Box2D(categories=["cat", "dog"])
# Expand the first layer of the external type
>>> box2d.internal_type
label._Label(
  geometry={
    'xmin': float32(),
    'ymin': float32(),
    'xmax': float32(),
    'ymax': float32(),
  },
  categories=['cat', 'dog'],
)
# Expand the top level of the external type to internal type
>>> box2d.to_builtin()
record(
  fields={
    'xmin': float32(),
    'ymin': float32(),
    'xmax': float32(),
    'ymax': float32(),
    'category': label.Category(
      categories=['cat', 'dog'],
    ),
  },
)
```

### 1.6.4 Search

This topic describes DataFrame search methods:

- `query()`
- `apply()`

### Dataset Preparation

Take the following DataFrame as an example:

```
from graviti import DataFrame
import graviti.portex as pt

points_schema = pt.array(
    pt.record(
        {
            "xmin": pt.int32(),
            "ymin": pt.int32(),
            "category": pt.enum(["boat", "car"]),
        }
    )
)
schema = pt.record(
    {
        "filename": pt.string(),
        "points": points_schema,
    }
)
data = []
for filename in ("a.jpg", "b.jpg", "c.jpg"):
    row_data = {
        "filename": filename,
        "points": [
            {
                "xmin": 1,
                "ymin": 1,
                "category": "boat",
            },
            {
                "xmin": 100,
                "ymin": 100,
                "category": "car" if filename == "a.jpg" else "boat",
            },
        ],
    }
    data.append(row_data)
df = DataFrame(data, schema)
```

```
>>> df
  filename  points
0  a.jpg      DataFrame(2, 3)
```

(continues on next page)

(continued from previous page)

```
1 b.jpg      DataFrame(2, 3)
2 c.jpg      DataFrame(2, 3)
```

Upload the DataFrame:

```
from graviti import Workspace
ws = Workspace(f"{YOUR_ACCESSKEY}")
dataset = ws.datasets.create("Graviti-dataset-demo")
draft = dataset.drafts.create("Draft-1")
draft["train"] = DataFrame(data=data, schema=schema)
draft.upload()
draft.commit("Commit-1")
```

Get the uploaded DataFrame:

```
df = dataset["train"]
```

## Query

The query operation will use the lambda function to evaluate each rows, and return the True rows. The lambda function must return a boolean value.

SDK uses the `engine.online()` to start online searching. For example, search for all rows with filename as “a.jpg”:

```
>>> from graviti import engine
>>> with engine.online():
...     result = df.query(lambda x: x["filename"] == "a.jpg")
>>> result
      filename  points
0  a.jpg      DataFrame(2, 3)
```

SDK use `any()` to match points in rows where at least one category is boat:

```
>>> from graviti import engine
>>> with engine.online():
...     result = df.query(lambda x: (x["points"]["category"]=="boat").any())
>>> result
      filename  points
0  a.jpg      DataFrame(2, 3)
1  b.jpg      DataFrame(2, 3)
2  c.jpg      DataFrame(2, 3)
```

SDK use `all()` to match points in rows whose category are all boat:

```
>>> from graviti import engine
>>> with engine.online():
...     result = df.query(lambda x: (x["points"]["category"]=="boat").all())
>>> result
      filename  points
0  b.jpg      DataFrame(2, 3)
1  c.jpg      DataFrame(2, 3)
```



## Apply

The apply operation will apply the lambda function to DataFrame row by row.

Search all points with the categories of “car”:

```
>>> from graviti import engine
>>> with engine.online():
...     result = df.apply(lambda x: x["points"].query(lambda y: y["category"]=="car"))
>>> result
0 DataFrame(1, 3)
1 DataFrame(0, 3)
2 DataFrame(0, 3)
```

## Query & Apply

SDK also supports calling apply() after the query().

Search all rows with the points category has “car” and remove null rows:

```
>>> from graviti import engine
>>> with engine.online():
...     result = df.query(lambda x: (x["points"]["category"] == "car").any()).apply(
...         lambda x: x["points"].query(lambda y: y["category"] == "car")
...     )
>>> result
0 DataFrame(1, 3)
```

In the Graviti SDK, the relationship between sheet name, DataFrame and dataset/draft is like the relationship between key, value and dict. Thus, SDK supports managing sheets like manipulating the dict in python.

The following will introduce more details about the sheet management methods in the SDK. First of all, it is necessary to get a dataset:

```
from graviti import Workspace

ws = Workspace(f"{YOUR_ACCESSKEY}")
dataset = ws.datasets.get(f"{DATASET_NAME}")
```

## 1.6.5 Create a Sheet

SDK supports managing sheets of open drafts. Thus, it is necessary to create or get an open draft:

```
draft = dataset.drafts.create(f"{DRAFT_TITLE}")
```

Then users need to *Initialize a DataFrame* like:

```
>>> from graviti import DataFrame
>>> data = [
...     {"filename": "a.jpg"},
...     {"filename": "b.jpg"},
...     {"filename": "c.jpg"},
... ]
```

(continues on next page)

(continued from previous page)

```
>>> df1 = DataFrame(data)
>>> df1
  filename
0  a.jpg
1  b.jpg
2  c.jpg
```

Next, users can create a new sheet:

```
draft[f"{SHEET_NAME}"] = df1
draft.upload()
```

---

**Note:** Only changes made on the open draft can be synchronized to Graviti via the method `upload()`.

---

Or replacing the old sheet:

```
draft["train"] = df2
draft.upload()
```

## 1.6.6 List Sheets

List sheets for the specified version of the dataset:

```
dataset.checkout(f"{COMMIT_ID}")
list(dataset.keys())
```

List sheets of an open draft:

```
draft = dataset.drafts.get({DRAFT_NUMBER})
list(draft.keys())
```

## 1.6.7 Get a Sheet

Get a sheet for the specified version of the dataset by name:

```
dataset.checkout(f"{REVISION}")
dataset[f"{SHEET_NAME}"]
```

Get a sheet of an open draft by name:

```
draft = dataset.drafts.get({DRAFT_NUMBER})
draft[f"{SHEET_NAME}"]
```

## 1.6.8 Delete a Sheet

Delete a sheet of the open draft by name:

```
del draft[f"{SHEET_NAME}"]
draft.upload()
```

## 1.7 graviti

Graviti Python SDK.

### 1.7.1 Subpackages

`graviti.dataframe`

Dataframe module.

#### Subpackages

`graviti.dataframe.column`

Column Series module.

#### Submodules

`graviti.dataframe.column.indexing`

The implementation of the Graviti indexing related class.

### Module Contents

#### Classes

<code>ColumnSeriesIlocIndexer</code>	Index class for ColumnSeries.iloc.
<code>ColumnSeriesLocIndexer</code>	Index class for ColumnSeries.loc.

**class** `graviti.dataframe.column.indexing.ColumnSeriesIlocIndexer(obj)`  
Index class for ColumnSeries.iloc.

**Parameters** `obj` (`graviti.dataframe.column.series.SeriesBase`) –

**class** `graviti.dataframe.column.indexing.ColumnSeriesLocIndexer(obj)`  
Index class for ColumnSeries.loc.

**Parameters** `obj` (`graviti.dataframe.column.series.SeriesBase`) –

## `graviti.dataframe.column.series`

The implementation of the Graviti Series.

## Module Contents

### Classes

<a href="#"><i>SeriesBase</i></a>	One-dimensional array.
<a href="#"><i>Series</i></a>	One-dimensional array.
<a href="#"><i>ArraySeries</i></a>	One-dimensional array for portex builtin type array.
<a href="#"><i>FileSeries</i></a>	One-dimensional array for file.
<a href="#"><i>EnumSeries</i></a>	One-dimensional array for portex builtin type enum.

**class** `graviti.dataframe.column.series.SeriesBase`

Bases: [\*graviti.dataframe.container.Container\*](#)

One-dimensional array.

#### Parameters

- **data** – The data that needs to be stored in Series. Could be ndarray or Iterable.
- **schema** – Data type to force. If None, will be inferred from data.

### Examples

Constructing Series from a list.

```
>>> d = [1,2,3,4]
>>> series = Series(data=d)
>>> series
0 1
1 2
2 3
3 4
```

**classmethod** `from_pyarrow(cls, array, schema=None)`

Instantiate a Series backed by an pyarrow array.

#### Parameters

- **array** (*pyarrow.Array*) – The input pyarrow array.
- **schema** (*Optional[graviti.portex.PortexType]*) – The schema of the series. If None, will be inferred from *array*.
- **cls** (*Type[\_SB]*) –

**Raises** **TypeError** – When the schema is mismatched with the pyarrow array type.

**Returns** The loaded Series instance.

**Return type** `_SB`

**property** `iloc(self)`

Purely integer-location based indexing for selection by position.

Allowed inputs are:

- An integer, e.g. 5.
- A list or array of integers, e.g. [4, 3, 0].
- A slice object with ints, e.g. 1:7.
- A boolean array of the same length as the axis being sliced.

**Returns** The instance of the `ILocIndexer`.

**Return type** *graviti.dataframe.column.indexing.ColumnSeriesILocIndexer*

**Examples**

```
>>> series = Series([1, 2, 3])
>>> series.loc[0]
1
>>> df.loc[[0]]
0    1
dtype: int64
```

**property** `loc(self)`

Access a group of rows and columns by indexes or a boolean array.

Allowed inputs are:

- A single index, e.g. 5.
- A list or array of indexes, e.g. [4, 3, 0].
- A slice object with indexes, e.g. 1:7.
- A boolean array of the same length as the axis being sliced.

**Returns** The instance of the `LocIndexer`.

**Return type** *graviti.dataframe.column.indexing.ColumnSeriesLocIndexer*

**Examples**

```
>>> series = Series([1, 2, 3])
>>> series.loc[0]
1
>>> df.loc[[0]]
0    1
dtype: int64
```

**class** `graviti.dataframe.column.series.Series`

Bases: *SeriesBase*

One-dimensional array.

**Parameters**

- **data** – The data that needs to be stored in Series. Could be ndarray or Iterable.
- **schema** – Data type to force. If None, will be inferred from data.

## Examples

Constructing Series from a list.

```
>>> d = [1,2,3,4]
>>> series = Series(data=d)
>>> series
0 1
1 2
2 3
3 4
```

**to\_pylist**(self)

Convert the Series to a python list.

**Returns** The python list representing the Series.

**Return type** List[*Any*]

**class** graviti.dataframe.column.series.**ArraySeries**

Bases: [SeriesBase](#)

One-dimensional array for portex builtin type array.

**to\_pylist**(self)

Convert the Series to a python list.

**Returns** The python list representing the Series.

**Return type** List[*Any*]

**class** graviti.dataframe.column.series.**FileSeries**

Bases: [SeriesBase](#)

One-dimensional array for file.

**to\_pylist**(self)

Convert the BinaryFileSeries to python list.

**Returns** The python list.

**Return type** List[graviti.file.FileBase]

**class** graviti.dataframe.column.series.**EnumSeries**

Bases: [Series](#)

One-dimensional array for portex builtin type enum.

**to\_pylist**(self)

Convert the Series to a python list.

**Returns** The python list representing the Series.

**Return type** List[*Any*]

## `graviti.dataframe.row`

Row Series module.

### Submodules

#### `graviti.dataframe.row.indexing`

The implementation of the Graviti indexing related class.

### Module Contents

#### Classes

<code>RowSeriesILocIndexer</code>	Index class for RowSeries.iloc.
<code>RowSeriesLocIndexer</code>	Index class for RowSeries.loc.

**class** `graviti.dataframe.row.indexing.RowSeriesILocIndexer(obj)`  
 Index class for RowSeries.iloc.

**Parameters** `obj` (`graviti.dataframe.row.series.Series`) –

**class** `graviti.dataframe.row.indexing.RowSeriesLocIndexer(obj)`  
 Index class for RowSeries.loc.

**Parameters** `obj` (`graviti.dataframe.row.series.Series`) –

#### `graviti.dataframe.row.series`

The implementation of the Graviti Series.

### Module Contents

#### Classes

<code>Series</code>	One-dimensional array.
---------------------	------------------------

**class** `graviti.dataframe.row.series.Series(data=None, schema=None, index=None)`  
 One-dimensional array.

#### Parameters

- **data** (`Optional[Dict[str, Any]]`) – The data that needs to be stored in Series. Could be ndarray or Iterable.
- **schema** (`Any`) – Data type to force. If None, will be inferred from data.
- **index** (`Optional[List[str]]`) – Index of the data.

## Examples

Constructing Series from a list.

```
>>> d = {"filename": "a.jpg", "attributes": {"color": "red", "pose": "frontal"}}
>>> series = Series(data=d)
>>> series
filename          a.jpg
attributes color      red
              pose frontal
```

### **property** `iloc(self)`

Purely integer-location based indexing for selection by position.

Allowed inputs are:

- An integer, e.g. 5.
- A list or array of integers, e.g. [4, 3, 0].
- A slice object with ints, e.g. 1:7.
- A boolean array of the same length as the axis being sliced.

**Returns** The instance of the `ILocIndexer`.

**Return type** *graviti.dataframe.row.indexing.RowSeriesILocIndexer*

## Examples

```
>>> series = Series([1, 2, 3])
>>> series.loc[0]
1
>>> df.loc[[0]]
0    1
dtype: int64
```

### **property** `loc(self)`

Access a group of rows and columns by indexes or a boolean array.

Allowed inputs are:

- A single index, e.g. 5.
- A list or array of indexes, e.g. [4, 3, 0].
- A slice object with indexes, e.g. 1:7.
- A boolean array of the same length as the axis being sliced.

**Returns** The instance of the `LocIndexer`.

**Return type** *graviti.dataframe.row.indexing.RowSeriesLocIndexer*



## Examples

```
>>> series = Series([1, 2, 3])
>>> series.loc[0]
1
>>> df.loc[[0]]
0    1
dtype: int64
```

## graviti.dataframe.sql

search module.

## Submodules

### graviti.dataframe.sql.array

The implementation of the search related array.

## Module Contents

### Classes

<i>LogicalOperatorsMixin</i>	A mixin for dynamically implementing logical operators.
<i>ScalarArray</i>	One-dimensional array for portex builtin type array.
<i>BooleanScalarArray</i>	One-dimensional array for portex builtin type array with the boolean items.
<i>StringScalarArray</i>	One-dimensional array for portex builtin type array with the string and enum items.
<i>EnumScalarArray</i>	One-dimensional array for portex builtin type array with the string and enum items.
<i>NumberScalarArray</i>	One-dimensional array for portex builtin type array with the numerical items.
<i>DataFrame</i>	The Two-dimensional array for the search.
<i>ArrayDistributor</i>	A distributor to instance DataFrame, ArrayScalar by different array items.
<i>ArraySeries</i>	The One-dimensional array for the search.

### **class** graviti.dataframe.sql.array.LogicalOperatorsMixin

A mixin for dynamically implementing logical operators.

### **class** graviti.dataframe.sql.array.ScalarArray(expr, schema, upper\_expr)

Bases: *graviti.dataframe.sql.container.SearchContainer*

One-dimensional array for portex builtin type array.

#### Parameters

- **expr** (*\_E*) –

- **schema** (*graviti.portex.PortexType*) –
- **upper\_expr** (*\_E*) –

**query**(*self, func*)

Query the data of an ArraySeries with a lambda function.

**Parameters** **func** (*Callable[[Any], Any]*) – The query function.

**Returns** The ArraySeries with the query expression.

**Return type** *ScalarArray*

**any**(*self*)

Whether any element is True.

**Returns** The BooleanSeries with the any expression.

**Return type** *graviti.dataframe.sql.scalar.BooleanScalar*

**all**(*self*)

Whether all elements are True.

**Returns** The BooleanSeries with the all expression.

**Return type** *graviti.dataframe.sql.scalar.BooleanScalar*

**class** *graviti.dataframe.sql.array.BooleanScalarArray*(*expr, schema, upper\_expr*)

Bases: *ScalarArray, LogicalOperatorsMixin*

One-dimensional array for portex builtin type array with the boolean items.

**Parameters**

- **expr** (*\_E*) –
- **schema** (*graviti.portex.PortexType*) –
- **upper\_expr** (*\_E*) –

**class** *graviti.dataframe.sql.array.StringScalarArray*(*expr, schema, upper\_expr*)

Bases: *ScalarArray, LogicalOperatorsMixin*

One-dimensional array for portex builtin type array with the string and enum items.

**Parameters**

- **expr** (*\_E*) –
- **schema** (*graviti.portex.PortexType*) –
- **upper\_expr** (*\_E*) –

**class** *graviti.dataframe.sql.array.EnumScalarArray*(*expr, schema, upper\_expr*)

Bases: *ScalarArray*

One-dimensional array for portex builtin type array with the string and enum items.

**Parameters**

- **expr** (*\_E*) –
- **schema** (*graviti.portex.PortexType*) –
- **upper\_expr** (*\_E*) –

**class** graviti.dataframe.sql.array.**NumberScalarArray**(*expr, schema, upper\_expr*)  
 Bases: *ScalarArray, graviti.dataframe.sql.scalar.ComparisonArithmeticOperatorsMixin*

One-dimensional array for portex builtin type array with the numerical items.

**Parameters**

- **expr** (*\_E*) –
- **schema** (*graviti.portex.PortexType*) –
- **upper\_expr** (*\_E*) –

**size**(*self*)

Get the length of array series.

**Returns** The NumberScalar with the size expression.

**Return type** *graviti.dataframe.sql.scalar.NumberScalar*

**max**(*self*)

Get the max value of array series.

**Returns** The NumberScalar with the max expression.

**Return type** *graviti.dataframe.sql.scalar.NumberScalar*

**min**(*self*)

Get the min value of array series.

**Returns** The NumberScalar with the min expression.

**Return type** *graviti.dataframe.sql.scalar.NumberScalar*

**sum**(*self*)

Get the sum of array series.

**Returns** The NumberScalar with the sum expression.

**Return type** *graviti.dataframe.sql.scalar.NumberScalar*

**class** graviti.dataframe.sql.array.**DataFrame**(*expr, schema, upper\_expr*)

Bases: *graviti.dataframe.sql.container.SearchContainer*

The Two-dimensional array for the search.

**Parameters**

- **expr** (*\_E*) –
- **schema** (*graviti.portex.PortexType*) –
- **upper\_expr** (*\_E*) –

**query**(*self, func*)

Query the data of an ArraySeries with a lambda function.

**Parameters** **func** (*Callable[[Any], Any]*) – The query function.

**Returns** The DataFrame with the query expression.

**Return type** *DataFrame*

**class** graviti.dataframe.sql.array.**ArrayDistributor**(*expr, schema, upper\_expr*)

Bases: *graviti.dataframe.sql.container.SearchContainer*

A distributor to instance DataFrame, ArrayScalar by different array items.

**Parameters**

- **expr** (*\_E*) –
- **schema** (*graviti.portex.PortexType*) –
- **upper\_expr** (*\_E*) –

**classmethod from\_upper**(*cls, expr, schema*)

Instantiate a Search object from the upper level.

#### Parameters

- **expr** (*Union[str, Dict[str, Any]]*) – The expression of the search.
- **schema** (*graviti.portex.PortexType*) – The schema of the series.

**Returns** The loaded object.

**Return type** *graviti.dataframe.sql.container.SearchContainer*

**class** *graviti.dataframe.sql.array.ArraySeries*(*expr, schema, upper\_expr*)

Bases: *graviti.dataframe.sql.container.SearchContainer*

The One-dimensional array for the search.

#### Parameters

- **expr** (*\_E*) –
- **schema** (*graviti.portex.PortexType*) –
- **upper\_expr** (*\_E*) –

*graviti.dataframe.sql.container*

The search container and register.

## Module Contents

### Classes

<i>SearchScalarContainer</i>	The base class for the search container.
<i>SearchContainer</i>	The base class for the search array container.
<i>SearchContainerRegister</i>	The class decorator to connect portex type and the search array container.

**class** *graviti.dataframe.sql.container.SearchScalarContainer*(*expr, schema*)

The base class for the search container.

#### Parameters

- **expr** (*\_E*) – The expression of the search.
- **schema** (*graviti.portex.PortexType*) – The schema of the series.

**classmethod from\_upper**(*cls, expr, schema*)

Instantiate a Search object from the upper level.

#### Parameters

- **expr** (*\_E*) – The expression of the search.

- **schema** (*graviti.portex.PortexType*) – The schema of the series.
- **cls** (*Type[\_S]*) –

**Returns** The loaded object.

**Return type** *\_S*

**class** *graviti.dataframe.sql.container.SearchContainer*(*expr, schema, upper\_expr*)

The base class for the search array container.

#### Parameters

- **expr** (*\_E*) – The expression of the search.
- **schema** (*graviti.portex.PortexType*) – The schema of the series.
- **upper\_expr** (*\_E*) – The expression of the search.

**classmethod** **from\_upper**(*cls, expr, schema*)

Instantiate a Search object from the upper level.

#### Parameters

- **expr** (*\_E*) – The upper expression of the search.
- **schema** (*graviti.portex.PortexType*) – The schema of the series.

**Returns** The loaded object.

**Return type** *SearchContainer*

**class** *graviti.dataframe.sql.container.SearchContainerRegister*(\**portex\_types*)

The class decorator to connect portex type and the search array container.

**Parameters** **portex\_types** (*Type[graviti.portex.base.PortexType]*) – The portex types needs to be connected.

## *graviti.dataframe.sql.scalar*

The implementation of the search related Scalar.

## Module Contents

### Classes

<i>LogicalOperatorsMixin</i>	A mixin for dynamically implementing logical operators.
<i>ComparisonArithmeticOperatorsMixin</i>	A mixin for dynamically implementing comparison and arithmetic operators.
<i>NumberScalar</i>	One-dimensional array for numerical portex builtin type.
<i>BooleanScalar</i>	One-dimensional array for portex builtin type boolean.
<i>StringScalar</i>	One-dimensional array for portex builtin type string.
<i>EnumScalar</i>	One-dimensional array for portex builtin type enum.
<i>RowSeries</i>	The One-dimensional array for the search.

## Attributes

---

*NUMERICAL\_PRIORITY*

---

`graviti.dataframe.sql.scalar.NUMERICAL_PRIORITY`

**class** `graviti.dataframe.sql.scalar.LogicalOperatorsMixin`

A mixin for dynamically implementing logical operators.

**class** `graviti.dataframe.sql.scalar.ComparisonArithmeticOperatorsMixin`

A mixin for dynamically implementing comparison and arithmetic operators.

**check\_type\_for\_other**(*self, other, opt*)

Check whether the other series is the same type as self series.

**Parameters**

- **other** (`graviti.dataframe.sql.container.SearchScalarContainer`) – The needed check series.
- **opt** (*str*) – Name of opt.

**Raises** **TypeError** – When the right series type is different from left.

**Return type** `None`

**class** `graviti.dataframe.sql.scalar.NumberScalar(expr, schema)`

Bases: `graviti.dataframe.sql.container.SearchScalarContainer`, `LogicalOperatorsMixin`, `ComparisonArithmeticOperatorsMixin`

One-dimensional array for numerical portex builtin type.

**Parameters**

- **expr** (*\_E*) –
- **schema** (`graviti.portex.PortexType`) –

**class** `graviti.dataframe.sql.scalar.BooleanScalar(expr)`

Bases: `graviti.dataframe.sql.container.SearchScalarContainer`, `LogicalOperatorsMixin`

One-dimensional array for portex builtin type boolean.

**Parameters** **expr** (`Union[str, Dict[str, Any]]`) –

**class** `graviti.dataframe.sql.scalar.StringScalar(expr, schema)`

Bases: `graviti.dataframe.sql.container.SearchScalarContainer`, `LogicalOperatorsMixin`

One-dimensional array for portex builtin type string.

**Parameters**

- **expr** (*\_E*) –
- **schema** (`graviti.portex.PortexType`) –

**class** `graviti.dataframe.sql.scalar.EnumScalar(expr, schema)`

Bases: `graviti.dataframe.sql.container.SearchScalarContainer`

One-dimensional array for portex builtin type enum.

**Parameters**

- **expr** (*\_E*) –

- **schema** (*graviti.portex.PortexType*) –

**class** *graviti.dataframe.sql.scalar.RowSeries*(*schema*)

Bases: *graviti.dataframe.sql.container.SearchScalarContainer*

The One-dimensional array for the search.

**Parameters** **schema** (*graviti.portex.PortexRecordBase*) –

## Submodules

**graviti.dataframe.container**

The table-structured data container related classes.

## Module Contents

### Classes

<i>Container</i>	The base class for the table-structured data container.
------------------	---

### Attributes

<i>RECORD_KEY</i>
-------------------

*graviti.dataframe.container.RECORD\_KEY* = *\_\_record\_key*

**class** *graviti.dataframe.container.Container*

The base class for the table-structured data container.

**property** *iloc*(*self*)

Purely integer-location based indexing for selection by position.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** *Any*

**property** *loc*(*self*)

Access the row by index.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** *Any*

**abstract** *to\_pylist*(*self*)

Convert the container to a python list.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** *List[Any]*

**copy**(*self*)

Get a copy of the container.

**Returns** A copy of the container.

**Parameters** `self (_T)` –

**Return type** `_T`

## `graviti.dataframe.frame`

The implementation of the Graviti DataFrame.

## Module Contents

### Classes

---

<code>DataFrame</code>	Two-dimensional, size-mutable, potentially heterogeneous tabular data.
------------------------	--

---

### Attributes

---

<code>APPLY_KEY</code>
------------------------

---

```
graviti.dataframe.frame.APPLY_KEY = apply_result
```

```
class graviti.dataframe.frame.DataFrame
```

Bases: `graviti.dataframe.container.Container`

Two-dimensional, size-mutable, potentially heterogeneous tabular data.

#### Parameters

- **data** – The data that needs to be stored in DataFrame.
- **schema** – The schema of the DataFrame. If None, will be inferred from *data*.
- **columns** – Column labels to use for resulting frame when data does not have them, defaulting to `RangeIndex(0, 1, 2, ..., n)`. If data contains column labels, will perform column selection instead.

## Examples

Constructing DataFrame from list.

```
>>> df = DataFrame(  
...     [  
...         {"filename": "a.jpg", "box2ds": {"x": 1, "y": 1}},  
...         {"filename": "b.jpg", "box2ds": {"x": 2, "y": 2}},  
...         {"filename": "c.jpg", "box2ds": {"x": 3, "y": 3}},  
...     ]  
... )  
>>> df
```

(continues on next page)



(continued from previous page)

	filename	box2ds	
		x	y
0	a.jpg	1	1
1	b.jpg	2	2
2	c.jpg	3	3

**classmethod** `from_pyarrow(cls, array, schema=None)`

Create DataFrame with pyarrow struct array.

#### Parameters

- **array** (*pyarrow.StructArray*) – The input pyarrow struct array.
- **schema** (*Optional[graviti.portex.PortexRecordBase]*) – The schema of the DataFrame.
- **cls** (*Type[\_T]*) –

**Raises** **TypeError** – When the given schema is mismatched with the pyarrow array type.

**Returns** The loaded DataFrame instance.

**Return type** `_T`

**property** `iloc(self)`

Purely integer-location based indexing for selection by position.

Allowed inputs are:

- An integer, e.g. 5.
- A tuple, e.g. (5, "COLUMN\_NAME")

**Returns** The instance of the ILocIndexer.

**Return type** *graviti.dataframe.indexing.DataFrameILocIndexer*

## Examples

```
>>> df = DataFrame({"col1": [1, 2], "col2": [3, 4]})
>>> df.iloc[0]
col1    1
col2    3
Name: 0, dtype: int64
>>> df.iloc[0, "col1"]
1
```

**property** `loc(self)`

Access the row by indexes.

Allowed inputs are:

- A single index, e.g. 5.
- A tuple, e.g. (5, "COLUMN\_NAME")

**Returns** The instance of the LocIndexer.

**Return type** *graviti.dataframe.indexing.DataFrameLocIndexer*

## Examples

```
>>> df = DataFrame({"col1": [1, 2], "col2": [3, 4]})
>>> df.loc[0]
col1    1
col2    3
Name: 0, dtype: int64
>>> df.loc[0, "col1"]
1
```

### property `shape(self)`

Return a tuple representing the dimensionality of the DataFrame.

**Returns** Shape of the DataFrame.

**Return type** Tuple[int, int]

## Examples

```
>>> df = DataFrame(
...     [
...         {"filename": "a.jpg", "box2ds": {"x": 1, "y": 1}},
...         {"filename": "b.jpg", "box2ds": {"x": 2, "y": 2}},
...         {"filename": "c.jpg", "box2ds": {"x": 3, "y": 3}},
...     ]
... )
>>> df
  filename box2ds
0  a.jpg    x    y
1  b.jpg    2    2
2  c.jpg    3    3
>>> df.shape
(3, 2)
```

### property `size(self)`

Return an int representing the number of elements in this object.

**Returns** Size of the DataFrame.

**Return type** int

## Examples

```
>>> df = DataFrame({"col1": [1, 2], "col2": [3, 4]})
>>> df.size
4
```

### method `head(self, n=5)`

Return the first  $n$  rows.

#### Parameters

- **n** (*int*) – Number of rows to select.

- `self(_T)` –

**Returns** The first  $n$  rows.

**Return type** `_T`

### Examples

```
>>> df = DataFrame(
...     [
...         {"animal": "alligator"},
...         {"animal": "bee"},
...         {"animal": "falcon"},
...         {"animal": "lion"},
...         {"animal": "monkey"},
...         {"animal": "parrot"},
...         {"animal": "shark"},
...         {"animal": "whale"},
...         {"animal": "zebra"},
...     ]
... )
>>> df
   animal
0 alligator
1      bee
2   falcon
3     lion
4   monkey
5   parrot
6    shark
7    whale
8    zebra
```

Viewing the first  $n$  lines (three in this case)

```
>>> df.head(3)
   animal
0 alligator
1      bee
2   falcon
```

For negative values of  $n$

```
>>> df.head(-3)
   animal
0 alligator
1      bee
2   falcon
3     lion
4   monkey
5   parrot
```

**tail**(*self*, *n*=5)

Return the last  $n$  rows.

**Parameters**

- **n** (*int*) – Number of rows to select.
- **self** (*\_T*) –

**Returns** The last *n* rows.

**Return type** *\_T*

**Examples**

```
>>> df = DataFrame(  
...     [  
...         {"animal": "alligator"},  
...         {"animal": "bee"},  
...         {"animal": "falcon"},  
...         {"animal": "lion"},  
...         {"animal": "monkey"},  
...         {"animal": "parrot"},  
...         {"animal": "shark"},  
...         {"animal": "whale"},  
...         {"animal": "zebra"},  
...     ]  
... )  
>>> df  
   animal  
0 alligator  
1      bee  
2    falcon  
3      lion  
4    monkey  
5    parrot  
6     shark  
7     whale  
8     zebra
```

Viewing the last 5 lines

```
>>> df.tail()  
   animal  
0  monkey  
1  parrot  
2   shark  
3   whale  
4   zebra
```

Viewing the last *n* lines (three in this case)

```
>>> df.tail(3)  
   animal  
0   shark  
1   whale  
2   zebra
```

**extend(self, values)**

Extend Sequence object or DataFrame to itself row by row.

**Parameters** **values** (*Union[Iterable[Dict[str, Any]], DataFrame]*) – A sequence object or DataFrame.

**Raises**

- **TypeError** – When the self is the member of another Dataframe.
- **TypeError** – When the given Dataframe mismatched with the self schema.

**Return type** None

**Examples**

```
>>> df = DataFrame([
...     {"filename": "a.jpg", "box2ds": {"x": 1, "y": 1}},
...     {"filename": "b.jpg", "box2ds": {"x": 2, "y": 2}},
... ])
```

Extended by another list.

```
>>> df.extend([{"filename": "c.jpg", "box2ds": {"x": 3, "y": 3}}])
>>> df
  filename box2ds
        x      y
0  a.jpg      1      1
1  b.jpg      2      2
2  c.jpg      3      3
```

Extended by another DataFrame.

```
>>> df2 = DataFrame([{"filename": "d.jpg", "box2ds": {"x": 4, "y": 4}}])
>>> df.extend(df2)
>>> df
  filename box2ds
        x      y
0  a.jpg      1      1
1  b.jpg      2      2
2  d.jpg      4      4
```

**to\_pylist(self)**

Convert the DataFrame to a python list.

**Returns** The python list representing the DataFrame.

**Return type** List[Dict[str, Any]]

**query(self, func)**

Query the columns of a DataFrame with a lambda function.

**Parameters** **func** (*Callable[[Any], Any]*) – The query function.

**Returns** The query result DataFrame.

**Raises** **TypeError** – When the DataFrame is not in a Commit.

**Return type** *DataFrame*

## Examples

```
>>> df = DataFrame([
...     {"filename": "a.jpg", "box2ds": {"x": 1, "y": 1}},
...     {"filename": "b.jpg", "box2ds": {"x": 2, "y": 2}},
... ])
>>> df.query(lambda x: x["filename"] == "a.jpg")
   filename box2ds
0    a.jpg      1    1
```

### **apply**(self, func)

Apply a function to the DataFrame row by row.

**Parameters** **func** (*Callable[[Any], Any]*) – Function to apply to each row.

**Returns** The apply result DataFrame or Series.

**Raises** **TypeError** – When the DataFrame is not in a Commit.

**Return type** *graviti.dataframe.container.Container*

## Examples

```
>>> df = DataFrame([
...     {"filename": "a.jpg", "box2ds": {"x": 1, "y": 1}},
...     {"filename": "b.jpg", "box2ds": {"x": 2, "y": 2}},
... ])
>>> df.apply(lambda x: x["box2ds"]["x"] + 1)
   filename box2ds
0    a.jpg      2    1
1    b.jpg      3    2
```

## graviti.dataframe.indexing

The implementation of the Graviti indexing related class.

## Module Contents

### Classes

<i>DataFrameIlocIndexer</i>	Index class for DataFrame.iloc.
<i>DataFrameLocIndexer</i>	Index class for DataFrame.loc.

**class** graviti.dataframe.indexing.**DataFrameIlocIndexer**(obj)

Index class for DataFrame.iloc.

**Parameters** **obj** (*graviti.dataframe.DataFrame*) –

**class** graviti.dataframe.indexing.**DataFrameLocIndexer**(obj)

Index class for DataFrame.loc.

**Parameters** `obj` (*graviti.dataframe.DataFrame*) –

## graviti.file

File module.

## Submodules

### graviti.file.audio

Graviti audio file class.

## Module Contents

### Classes

<i>Audio</i>	This class represents local audio files.
<i>RemoteAudio</i>	This class represents remote audio files.

**class** `graviti.file.audio.Audio(path)`

Bases: *graviti.file.base.File*

This class represents local audio files.

**Parameters** `path` (*str*) –

**class** `graviti.file.audio.RemoteAudio(key, extension, size, object_policy_manager)`

Bases: *graviti.file.base.RemoteFile*

This class represents remote audio files.

**Parameters**

- **key** (*str*) –
- **extension** (*str*) –
- **size** (*int*) –
- **object\_policy\_manager** (*graviti.manager.ObjectPolicyManager*) –

### graviti.file.base

Graviti basic file class.

## Module Contents

### Classes

<i>FileBase</i>	This class represents the file in a DataFrame.
<i>File</i>	This class represents local files.
<i>RemoteFile</i>	This class represents the file on Graviti platform.

#### **class** graviti.file.base.FileBase

Bases: *graviti.utility.repr.ReprMixin*

This class represents the file in a DataFrame.

##### **property** key(*self*)

Get the key of the file.

**Returns** The key of the file.

**Return type** str

##### **property** extension(*self*)

Get the extension of the file.

**Returns** The extension of the file.

**Return type** str

##### **property** size(*self*)

Get the size of the file.

**Returns** The size of the file.

**Return type** int

##### **abstract** open(*self*)

Return the binary file pointer of this file.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** Union[*graviti.utility.requests.UserResponse*, *\_io.BufferedReader*]

#### **class** graviti.file.base.File(*path*)

Bases: *FileBase*

This class represents local files.

**Parameters** **path** (*str*) – The local path of the file.

##### **property** path(*self*)

Get the path of the file.

**Returns** The path of the file.

**Return type** *pathlib.Path*

##### **property** key(*self*)

Get the key of the file.

**Returns** The key of the file.

**Return type** str

##### **property** extension(*self*)

Get the extension of the file.



**Returns** The extension of the file.

**Return type** str

**property** `size(self)`

Get the size of the file.

**Returns** The size of the file.

**Return type** int

**get\_checksum(self)**

Get the sha1 checksum of the local file.

**Returns** The sha1 checksum of the local file.

**Return type** str

**open(self)**

Return the binary file pointer of this file.

**Returns** The local file pointer.

**Return type** `_io.BufferedReader`

**class** `graviti.file.base.RemoteFile(key, extension, size, object_policy_manager)`

Bases: [FileBase](#)

This class represents the file on Graviti platform.

#### Parameters

- **key** (str) – The key of the file.
- **extension** (str) – The extension of the file.
- **size** (int) – The size of the file.
- **object\_policy\_manager** (`graviti.manager.ObjectPolicyManager`) – The policy to access the file.

**open(self)**

Return the binary file pointer of this file.

**Returns** The remote file pointer.

**Return type** `graviti.utility.requests.UserResponse`

## `graviti.file.image`

Graviti image file class.

## Module Contents

### Classes

<a href="#"><i>Image</i></a>	This class represents local image files.
<a href="#"><i>RemoteImage</i></a>	This class represents remote image files.

**class** graviti.file.image.**Image**(*path*)

Bases: [graviti.file.base.File](#)

This class represents local image files.

**Parameters** *path* (*str*) – The local path of the image.

**property** *height*(*self*)

Get the height of the image.

**Returns** The height of the image.

**Return type** *int*

**property** *width*(*self*)

Get the width of the image.

**Returns** The width of the image.

**Return type** *int*

**class** graviti.file.image.**RemoteImage**(*key*, *extension*, *size*, *height*, *width*, *object\_policy\_manager*)

Bases: [graviti.file.base.RemoteFile](#)

This class represents remote image files.

**Parameters**

- **key** (*str*) – The key of the image file.
- **extension** (*str*) – The extension of the image file.
- **size** (*int*) – The size of the image file.
- **height** (*int*) – The height of the image.
- **width** (*int*) – The width of the image.
- **object\_policy\_manager** (*graviti.manager.ObjectPolicyManager*) – The policy to access the file.

**property** *height*(*self*)

Get the height of the image.

**Returns** The height of the image.

**Return type** *int*

**property** *width*(*self*)

Get the width of the image.

**Returns** The width of the image.

**Return type** *int*

**graviti.file.image\_size**

Functions to get image size.

**Module Contents****Classes**

<i>ImageFormatBase</i>	The base class for different image formats.
<i>JPEG</i>	The class for JPEG image format.
<i>PNG</i>	The class for PNG image format.
<i>OldPNG</i>	The class for an older version of PNG image format.
<i>GIF</i>	The class for GIF image format.
<i>JPEG2000</i>	The class for JPEG 2000 image format.
<i>BMP</i>	The class for BMP image format.
<i>TIFF</i>	The class for TIFF image format.
<i>ICO</i>	The class for ICO image format.
<i>WebP</i>	The class for WebP image format.
<i>FLIF</i>	The class for Flif image format.

**Functions**

<i>get_image_size(path)</i>	Get the height and width of the input image file.
-----------------------------	---

**Attributes**

<i>Image</i>
--------------

**graviti.file.image\_size.Image**

**graviti.file.image\_size.get\_image\_size(path)**

Get the height and width of the input image file.

**Parameters** *path* (*pathlib.Path*) – The path of the image.

**Returns** The height and width of the input image.

**Return type** Tuple[int, int]

**class graviti.file.image\_size.ImageFormatBase**

The base class for different image formats.

**classmethod check(cls, header, size)**

Check if the input header fits the current image format.

**Parameters**

- **header** (*bytes*) – The header of the image.
- **size** (*int*) – The size of the image.

**Returns** Whether if the input header fits the current image format.

**Return type** bool

**classmethod** `get_image_size(cls, header, fp)`

Get the height and width through the input data.

**Parameters**

- **header** (*bytes*) – The header of the image or the entire image.
- **fp** (*\_io.BufferedReader*) – The image file pointer.

**Returns** The height and width of the image.

**Raises** *ImageDecodeError* – When the input image file is invalid.

**Return type** Tuple[int, int]

**class** `graviti.file.image_size.JPEG`

Bases: *ImageFormatBase*

The class for JPEG image format.

**class** `graviti.file.image_size.PNG`

Bases: *ImageFormatBase*

The class for PNG image format.

**class** `graviti.file.image_size.OldPNG`

Bases: *ImageFormatBase*

The class for an older version of PNG image format.

**class** `graviti.file.image_size.GIF`

Bases: *ImageFormatBase*

The class for GIF image format.

**class** `graviti.file.image_size.JPEG2000`

Bases: *ImageFormatBase*

The class for JPEG 2000 image format.

**class** `graviti.file.image_size.BMP`

Bases: *ImageFormatBase*

The class for BMP image format.

**class** `graviti.file.image_size.TIFF`

Bases: *ImageFormatBase*

The class for TIFF image format.

**class** `graviti.file.image_size.ICO`

Bases: *ImageFormatBase*

The class for ICO image format.

**class** `graviti.file.image_size.WebP`

Bases: *ImageFormatBase*

The class for WebP image format.

**class** `graviti.file.image_size.FLIF`

Bases: *ImageFormatBase*

The class for Flif image format.

## graviti.file.point\_cloud

Graviti point cloud file class.

### Module Contents

#### Classes

<i>PointCloud</i>	This class represents local point cloud files.
<i>RemotePointCloud</i>	This class represents remote point cloud files.

**class** graviti.file.point\_cloud.PointCloud(*path*)

Bases: *graviti.file.base.File*

This class represents local point cloud files.

**Parameters** *path* (*str*) –

**class** graviti.file.point\_cloud.RemotePointCloud(*key*, *extension*, *size*, *object\_policy\_manager*)

Bases: *graviti.file.base.RemoteFile*

This class represents remote point cloud files.

**Parameters**

- **key** (*str*) –
- **extension** (*str*) –
- **size** (*int*) –
- **object\_policy\_manager** (*graviti.manager.ObjectPolicyManager*) –

## graviti.manager

Manager module.

### Submodules

## graviti.manager.branch

The implementation of the Branch and BranchManager.

### Module Contents

#### Classes

<i>Branch</i>	This class defines the structure of a branch.
<i>BranchManager</i>	This class defines the operations on the branch in Graviti.

**class** graviti.manager.branch.**Branch**(dataset, name, commit\_id)

Bases: [graviti.manager.commit.NamedCommit](#)

This class defines the structure of a branch.

**Parameters**

- **dataset** ([graviti.manager.dataset.Dataset](#)) – Class `Dataset` instance.
- **name** (*str*) – The name of the branch.
- **commit\_id** (*Optional[str]*) – The commit id.

**class** graviti.manager.branch.**BranchManager**(dataset)

This class defines the operations on the branch in Graviti.

**Parameters** **dataset** ([graviti.manager.dataset.Dataset](#)) – `Dataset` instance.

**create**(self, name, revision=`CURRENT_COMMIT`)

Create a branch.

**Parameters**

- **name** (*str*) – The branch name.
- **revision** (*str*) – The information to locate the specific commit, which can be the commit id, the branch name, or the tag name. The default value is the current commit of the dataset.

**Raises** [NoCommitsError](#) – When create branches on default branch without commit history.

**Returns** The [Branch](#) instance with the given name.

**Return type** [Branch](#)

**get**(self, name)

Get the branch with the given name.

**Parameters** **name** (*str*) – The required branch name.

**Raises** [ResourceNameError](#) – When the given name is an empty string.

**Returns** The [Branch](#) instance with the given name.

**Return type** [Branch](#)

**list**(self)

List the information of branches.

**Returns** The `LazyPagingList` of [branches](#) instances.

**Return type** [graviti.manager.lazy.LazyPagingList\[Branch\]](#)

**delete**(self, name)

Delete a branch.

**Parameters** **name** (*str*) – The name of the branch to be deleted.

**Raises** [ResourceNameError](#) – When the given name is an empty string.

**Return type** `None`

## graviti.manager.commit

The implementation of the Commit and CommitManager.

## Module Contents

### Classes

<i>Commit</i>	This class defines the structure of a commit.
<i>NamedCommit</i>	This class defines the structure of a named commit.
<i>CommitManager</i>	This class defines the operations on the commit in Graviti.

**class** graviti.manager.commit.**Commit**(dataset, commit\_id)

Bases: [graviti.manager.sheets.Sheets](#)

This class defines the structure of a commit.

#### Parameters

- **dataset** ([graviti.manager.dataset.Dataset](#)) – Class Dataset instance.
- **commit\_id** (*Optional[str]*) – The commit id.

**classmethod** **from\_response**(cls, dataset, contents)

Create a [Commit](#) instance from python dict.

#### Parameters

- **dataset** ([graviti.manager.dataset.Dataset](#)) – The dataset of the commit.
- **contents** (*Dict[str, Any]*) – A python dict containing all the information of the commit:

```
{
    "commit_id": <str>
    "parent_commit_id": <Optional[str]>
    "title": <str>
    "description": <str>
    "committer": <str>
    "committed_at": <str>
}
```

- **cls** (*Type[\_C]*) –

**Returns** A [Commit](#) instance created from the input python dict.

**Return type** *\_C*

**property** **parent**(self)

Return the parent of the commit.

**Returns** The parent of the commit.

**Return type** *Optional[Commit]*

**property** **title**(self)

Return the title of the commit.

**Returns** The title of the commit.

**Return type** str

**property description**(*self*)

Return the description of the commit.

**Returns** The description of the commit.

**Return type** str

**property committer**(*self*)

Return the committer of the commit.

**Returns** The committer of the commit.

**Return type** str

**property committed\_at**(*self*)

Return the time when the draft is committed.

**Returns** The time when the draft is committed.

**Return type** datetime.datetime

**search**(*self, sheet, criteria, schema=None*)

Create a search.

**Parameters**

- **sheet** (*str*) – The sheet name.
- **criteria** (*Dict[str, Any]*) – The criteria of search.
- **schema** (*Optional[graviti.portex.PortexRecordBase]*) – The schema of the search result DataFrame.

**Raises** *NoCommitsError* – When there is no commit on the current branch.

**Returns** The created DataFrame instance.

**Return type** graviti.dataframe.DataFrame

**class** graviti.manager.commit.**NamedCommit**(*dataset, name, commit\_id*)

Bases: *Commit*

This class defines the structure of a named commit.

*NamedCommit* is the base class of *Branch* and *Tag*.

**Parameters**

- **dataset** (*graviti.manager.dataset.Dataset*) – Class *Dataset* instance.
- **name** (*str*) – The name of the named commit.
- **commit\_id** (*Optional[str]*) – The commit id.

**classmethod from\_response**(*cls, dataset, contents*)

Create a *NamedCommit* instance from python dict.

**Parameters**

- **dataset** (*graviti.manager.dataset.Dataset*) – The dataset of the NamedCommit.
- **contents** (*Dict[str, Any]*) – A python dict containing all the information of the NamedCommit:



```
{
  "name": <str>
  "commit_id": <Optional[str]>
  "parent_commit_id": <Optional[str]>
  "title": <str>
  "description": <str>
  "committer": <str>
  "committed_at": <str>
}
```

- **cls** (*Type[\_NC]*) –

**Returns** A *NamedCommit* instance created from the input python dict.

**Return type** *\_NC*

**class** `graviti.manager.commit.CommitManager(dataset)`

This class defines the operations on the commit in Graviti.

**Parameters** **dataset** (`graviti.manager.dataset.Dataset`) – *Dataset* instance.

**get**(*self*, *revision=CURRENT\_COMMIT*)

Get the certain commit with the given revision.

**Parameters** **revision** (*str*) – The information to locate the specific commit, which can be the commit id, the branch name, or the tag name. The default value is the current commit of the dataset.

**Raises** *NoCommitsError* – When revision is not given and the commit id of current dataset is None, or when the given branch has no commit history yet.

**Returns** The *Commit* instance with the given revision.

**Return type** *Commit*

**list**(*self*, *revision=CURRENT\_COMMIT*)

List the commits.

**Parameters** **revision** (*str*) – The information to locate the specific commit, which can be the commit id, the branch name, or the tag name. If it is given, list the commits before the given commit. If it is not given, list the commits before the current commit.

**Returns** The LazyPagingList of *commits* instances.

**Return type** Union[*graviti.manager.lazy.LazyPagingList[Commit]*, List[Optional[*Commit*]]]

**graviti.manager.common**

Common tools.

## Module Contents

### Classes

---

<i>DefaultValue</i>	This class defines the default value of parameters from methods in manager.
---------------------	---

---

### Functions

---

<i>check_head_status</i> (head, remote_revision, remote_commit_id)	Check if the commit for the HEAD of the current dataset is up-to-date.
--	--

---

### Attributes

---

<i>LIMIT</i>
<i>CURRENT_COMMIT</i>
<i>CURRENT_BRANCH</i>
<i>ALL_BRANCHES</i>

---

`graviti.manager.common.LIMIT = 128`

**class** `graviti.manager.common.DefaultValue(name)`

This class defines the default value of parameters from methods in manager.

**Parameters** `name` (*str*) – The name of the default value.

`graviti.manager.common.CURRENT_COMMIT :Any`

`graviti.manager.common.CURRENT_BRANCH :Any`

`graviti.manager.common.ALL_BRANCHES :Any`

**exception** `graviti.manager.common.StatusWarning(revision_type, name)`

Bases: `Warning`

This class defines the warning that the commit of dataset branch or tag is not up-to-date.

**Parameters**

- **revision\_type** (*str*) – The type of current dataset revision.
- **name** (*str*) – The name of current dataset revision.

`graviti.manager.common.check_head_status(head, remote_revision, remote_commit_id)`

Check if the commit for the HEAD of the current dataset is up-to-date.

**Parameters**

- **head** (`graviti.manager.commit.Commit`) – The current revision of the local dataset.
- **remote\_revision** (*str*) – The revision of the remote dataset in server.

- **remote\_commit\_id** (*Optional[str]*) – The commit id of the remote dataset in server.

**Return type** None

## graviti.manager.dataset

The implementation of the Dataset and DatasetManager.

## Module Contents

### Classes

<i>RevisionType</i>	RevisionType is an enumeration type including "BRANCH", "COMMIT" and "TAG".
<i>ObjectPolicyManagerType</i>	ObjectPolicyManagerType is an enumeration type including "OSS", "S3" and "AZURE".
<i>Dataset</i>	This class defines the basic concept of the dataset on Graviti.
<i>DatasetManager</i>	This class defines the operations on the dataset on Graviti.

### Attributes

<i>logger</i>
<i>handler</i>

graviti.manager.dataset.**logger**

graviti.manager.dataset.**handler**

**class** graviti.manager.dataset.**RevisionType**

Bases: `enum.Enum`

RevisionType is an enumeration type including "BRANCH", "COMMIT" and "TAG".

**class** graviti.manager.dataset.**ObjectPolicyManagerType**

Bases: `enum.Enum`

ObjectPolicyManagerType is an enumeration type including "OSS", "S3" and "AZURE".

**class** graviti.manager.dataset.**Dataset** (*access\_key, url, response*)

Bases: `graviti.utility.UserMutableMapping[str, graviti.dataframe.DataFrame]`, `graviti.utility.ReprMixin`

This class defines the basic concept of the dataset on Graviti.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.

- **response** (*Dict[str, Any]*) – The response of the OpenAPI associated with the dataset:

```
{
  "id": <str>
  "name": <str>
  "alias": <str>
  "default_branch": <str>
  "commit_id": <Optional[str]>
  "cover_url": <str>
  "created_at": <str>
  "updated_at": <str>
  "owner": <str>
  "is_public": <bool>
  "config": <str>
  "backend_type": <str>
}
```

**dataset\_id**

Dataset ID.

**name**

The name of the dataset, unique for a user.

**alias**

Dataset alias.

**default\_branch**

The default branch of dataset.

**commit\_id**

The commit ID of the dataset.

**created\_at**

The time when the dataset was created.

**updated\_at**

The time when the dataset was last modified.

**owner**

The owner of the dataset.

**is\_public**

Whether the dataset is public.

**config**

The config of dataset.

**backend\_type**

The backend type of dataset storage.

**property HEAD(*self*)**

Return the current branch or commit.

**Returns** The current branch or commit.

**Return type** *graviti.manager.commit.Commit*

**property branches(*self*)**

Get class *BranchManager* instance.

**Returns** Required *BranchManager* instance.

**Return type** *graviti.manager.branch.BranchManager*

**property drafts(self)**

Get class *DraftManager* instance.

**Returns** Required *DraftManager* instance.

**Return type** *graviti.manager.draft.DraftManager*

**property commits(self)**

Get class *CommitManager* instance.

**Returns** Required *CommitManager* instance.

**Return type** *graviti.manager.commit.CommitManager*

**property tags(self)**

Get class *TagManager* instance.

**Returns** Required *TagManager* instance.

**Return type** *graviti.manager.tag.TagManager*

**checkout(self, revision)**

Checkout to a commit.

**Parameters** **revision** (*str*) – The information to locate the specific commit, which can be the commit id, the branch, or the tag.

**Return type** None

**edit(self, \*, name=None, alias=None, default\_branch=None)**

Update the meta data of the dataset.

**Parameters**

- **name** (*Optional[str]*) – The new name of the dataset.
- **alias** (*Optional[str]*) – The new alias of the dataset.
- **default\_branch** (*Optional[str]*) – The new default branch of the dataset.

**Return type** None

**commit(self, title, description=None, jobs=8, quiet=False)**

Create, upload and commit the draft to push the local dataset to Graviti.

**Parameters**

- **title** (*str*) – The commit title.
- **description** (*Optional[str]*) – The commit description.
- **jobs** (*int*) – The number of the max workers in multi-thread upload, the default is 8.
- **quiet** (*bool*) – Set to True to stop showing the upload process bar.

**Raises**

- **StatusError** – When the HEAD of the dataset is not a branch.
- **StatusError** – When the dataset has no modifications.

**Return type** None

**class** *graviti.manager.dataset.DatasetManager*(*access\_key, url, owner*)

This class defines the operations on the dataset on Graviti.

**Parameters**

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) –

**create**(*self*, *name*, *alias*="", *config*=None)  
Create a Graviti dataset with given name.

## Parameters

- **name** (*str*) – The name of the dataset, unique for a user.
- **alias** (*str*) – Alias of the dataset, default is “”.
- **config** (*Optional[str]*) – The auth storage config name.

**Returns** The created *Dataset* instance.

**Return type** *Dataset*

```
get(self, dataset)
```

Get a Graviti dataset with given name.

**Parameters** **dataset** (*str*) – The name of the dataset, unique for a user.

**Returns** The requested *Dataset* instance.

**Raises** *ResourceNameError* – When the required dataset does not exist.

**Return type** *Dataset*

```
list(self)
```

List Graviti datasets.

**Returns** The LazyPagingList of *Dataset* instances.

**Return type** `graviti.manager.lazy.LazyPagingList[Dataset]`

```
delete(self, name)
```

### Delete a Graviti dataset with given name.

**Parameters** **name** (*str*) – The name of the dataset, unique for a user.

**Return type** None

graviti.manager.draft

### The implementation of the Draft and DraftManager.

## Module Contents

## Classes

<i>Draft</i>	The basic structure of the Graviti draft.
<i>DraftManager</i>	This class defines the operations on the draft in Graviti.

```
class graviti.manager.draft.Draft(dataset, number, *, title, branch, state, parent_commit_id, creator,  
                                created_at, updated_at, description="")
```

Bases: `graviti.manager.sheets.Sheets`

The basic structure of the Graviti draft.

#### Parameters

- **dataset** (`graviti.manager.dataset.Dataset`) – Class `Dataset` instance.
- **number** (`int`) – The number of the draft.
- **title** (`str`) – The title of the draft.
- **branch** (`str`) – The based branch of the draft.
- **state** (`str`) – The draft state which includes “OPEN”, “CLOSED”, “COMMITTED”.
- **parent\_commit\_id** (`Optional[str]`) – The parent commit id.
- **creator** (`str`) – The creator of the draft.
- **created\_at** (`str`) – The time when the draft is created.
- **updated\_at** (`str`) – The time of last update.
- **description** (`str`) – The draft description.

#### parent

The parent of the draft.

**edit**(*self*, title=None, description=None)

Update title and description of the draft.

#### Parameters

- **title** (`Optional[str]`) – The title of the draft.
- **description** (`Optional[str]`) – The description of the draft.

**Return type** None

**close**(*self*)

Close the draft.

**Return type** None

**commit**(*self*, title, description=None, update\_dataset\_head=True)

Commit the current draft.

#### Parameters

- **title** (`str`) – The commit title.
- **description** (`Optional[str]`) – The commit description.
- **update\_dataset\_head** (`bool`) – Whether to update the dataset HEAD.
  - True (the default value): The dataset will be updated to the committed version. At this time, previous modifications to the dataset will be lost.
  - False: The HEAD of the dataset will not be updated. This can be set if the user needs to continue with some operations on the dataset.

**Returns** The `Branch` instance.

**Return type** `graviti.manager.branch.Branch`

## Examples

The default scenario: `update_dataset_head` is `True`.

```
>>> dataset = ws.datasets.get("Graviti-dataset-demo")
>>> dataset.HEAD.name # The version of the dataset is Branch("main").
"main"
>>> dataset.HEAD.commit_id
"524d110ecae7474eaec9471f4a6c28b0"
>>> draft = dataset.drafts.create("draft-4", branch="dev")
>>> draft.commit("commit-4")
Branch("dev")(
  (commit_id): '3db73ac2876a42c0bf43a0489ce1756a',
  (parent): Commit("1b21a40f03ab4cec814ec47ee0d10b24"),
  (title): 'commit-4',
  (committer): 'graviti-example',
  (committed_at): 2022-07-19 04:23:45+00:00
)
>>> dataset.HEAD.name # The version of the dataset has been updated to Branch(
↪ "dev").
"dev"
>>> dataset.HEAD.commit_id
"3db73ac2876a42c0bf43a0489ce1756a"
```

Set `update_dataset_head` to `False`.

```
>>> dataset = ws.datasets.get("Graviti-dataset-demo")
>>> dataset.HEAD.name # The version of the dataset is Branch("main").
"main"
>>> dataset.HEAD.commit_id
"524d110ecae7474eaec9471f4a6c28b0"
>>> draft = dataset.drafts.create("draft-5", branch="dev")
>>> draft.commit("commit-5", update_dataset_head=False)
Branch("dev")(
  (commit_id): '781007a41d1641859c87cb00f8e32bf3',
  (parent): Commit("3db73ac2876a42c0bf43a0489ce1756a"),
  (title): 'commit-5',
  (committer): 'graviti-example',
  (committed_at): 2022-07-19 04:25:45+00:00
)
>>> dataset.HEAD.name # The version of the dataset has not been updated.
"main"
>>> dataset.HEAD.commit_id
"524d110ecae7474eaec9471f4a6c28b0"
```

**upload**(*self*, *jobs*=8, *quiet*=False)

Upload the local dataset to Graviti.

### Parameters

- **jobs** (*int*) – The number of the max workers in multi-thread upload, the default is 8.
- **quiet** (*bool*) – Set to `True` to stop showing the upload process bar.

**Return type** `None`



**class** `graviti.manager.draft.DraftManager(dataset)`

This class defines the operations on the draft in Graviti.

**Parameters** `dataset` (`graviti.manager.dataset.Dataset`) – `Dataset` instance.

**create**(*self*, *title*, *description=None*, *branch=CURRENT\_BRANCH*)

Create a draft.

#### Parameters

- **title** (*str*) – The draft title.
- **description** (*Optional[str]*) – The draft description.
- **branch** (*str*) – The branch name. The default value is the current branch of the dataset.

**Returns** The `Draft` instance with the given title and description.

**Raises** `StatusError` – When creating the draft without basing on a branch.

**Return type** `Draft`

**get**(*self*, *draft\_number*)

Get the certain draft with the given draft number.

**Parameters** `draft_number` (*int*) – The required draft number.

**Returns** The `Draft` instance with the given number.

**Return type** `Draft`

**list**(*self*, *state='OPEN'*, *branch=ALL\_BRANCHES*)

List all the drafts.

#### Parameters

- **state** (*str*) – The draft state which includes “OPEN”, “CLOSED”, “COMMITTED”, “ALL”. The default value is “OPEN”.
- **branch** (*str*) – The branch name. The default value is all branches.

**Returns** The LazyPagingList of `drafts` instances.

**Return type** `graviti.manager.lazy.LazyPagingList[Draft]`

## `graviti.manager.lazy`

Related classes for the lazy evaluation.

## Module Contents

### Classes

<code>LazyItem</code>	In paging lazy evaluation system, a <code>LazyItem</code> instance represents an element in a pagination.
<code>ReturnGenerator</code>	<code>ReturnGenerator</code> is a generator wrap to get the return value easily.

continues on next page

Table 1.29 – continued from previous page

<i>LazyPage</i>	In paging lazy evaluation system, a LazyPage instance represents a page with elements.
<i>InitPage</i>	In paging lazy evaluation system, InitPage is the page to initialize <i>LazyPagingList</i> .
<i>LazyPagingList</i>	LazyPagingList is a wrap of web paging request.

## Attributes

### *PagingGenerator*

graviti.manager.lazy.**PagingGenerator**

**class** graviti.manager.lazy.**LazyItem**(*page, data*)

Bases: Generic[\_T]

In paging lazy evaluation system, a LazyItem instance represents an element in a pagination.

If user wants to access the element, LazyItem will trigger the paging request to pull a page of elements and return the required element. All the pulled elements will be stored in different LazyItem instances and will not be requested again.

#### Parameters

- **page** (*LazyPage[\_T]*) – The page the item belongs to.
- **data** (*\_T*) –

#### **page**

The parent *LazyPage* of this item.

#### **data**

The actual element stored in this item.

**classmethod** **from\_page**(*cls, page*)

Create a LazyItem instance from page.

**Parameters** **page** (*LazyPage[\_T]*) – The page of the element.

**Returns** The LazyItem instance which stores the input page.

**Return type** *LazyItem[\_T]*

**classmethod** **from\_data**(*cls, data*)

Create a LazyItem instance from data.

**Parameters** **data** (*\_T*) – The actual data needs to be stored in LazyItem.

**Returns** The LazyItem instance which stores the input data.

**Return type** *LazyItem[\_T]*

**get**(*self*)

Access the actual element represented by LazyItem.

If the element is already pulled from web, it will be return directly, otherwise this function will request for a page of elements to get the required element.

**Returns** The actual element this LazyItem instance represents.

**Return type** `_T`

**class** `graviti.manager.lazy.ReturnGenerator(generator)`

Bases: `Generic[_T, _R]`

`ReturnGenerator` is a generator wrap to get the return value easily.

**Parameters** `generator` (`Generator[_T, Any, _R]`) – The generator needs to be wrapped.

**value**

The return value of the input generator.

**class** `graviti.manager.lazy.LazyPage(offset, limit, func)`

Bases: `Generic[_T]`

In paging lazy evaluation system, a `LazyPage` instance represents a page with elements.

`LazyPage` is used for sending paging request to pull a page of elements and storing them in different `LazyItem` instances.

**Parameters**

- **offset** (`int`) – The offset of the page.
- **limit** (`int`) – The limit of the page.
- **func** (`PagingGenerator[_T]`) – A paging generator function, which takes `offset<int>` and `limit<int>` as inputs and returns a generator. The returned generator should yield the element user needs, and return the total count of the elements in the paging request.

**items**

The `LazyItem` list which represents a page of elements.

**classmethod** `from_items(cls, offset, limit, func, item_contents)`

Create a `LazyPage` instance with the given items and generator function.

**Parameters**

- **offset** (`int`) – The offset of the page.
- **limit** (`int`) – The limit of the page.
- **func** (`PagingGenerator[_T]`) – A paging generator function, which takes `offset<int>` and `limit<int>` as inputs and returns a generator. The returned generator should yield the element user needs, and return the total count of the elements in the paging request.
- **item\_contents** (`Iterable[_T]`) – The lazy item contents that need to be stored on this page.

**Returns** The `LazyPage` instance which stores the input items and function.

**Return type** `LazyPage[_T]`

**pull**(`self`)

Send paging request to pull a page of elements and store them in `LazyItem`.

**Return type** `None`

**class** `graviti.manager.lazy.InitPage(offset, limit, func)`

Bases: `LazyPage[_T]`

In paging lazy evaluation system, `InitPage` is the page to initialize `LazyPagingList`.

`InitPage` will send a paging request to pull a page of elements and storing them in different `LazyItem` instances when construction. And the `totalCount` of the page will also be stored in the instance.

**Parameters**

- **offset** (*int*) – The offset of the page.
- **limit** (*int*) – The limit of the page.
- **func** (*PagingGenerator[\_T]*) – A paging generator function, which takes offset<int> and limit<int> as inputs and returns a generator. The returned generator should yield the element user needs, and return the total count of the elements in the paging request.

**items**

The *LazyItem* list which represents a page of elements.

**total\_count**

The totalCount of the paging request.

**class** graviti.manager.lazy.**LazyPagingList**(*func, limit*)  
Bases: MutableSequence[\_T], graviti.utility.ReprMixin

LazyPagingList is a wrap of web paging request.

It follows the python MutableSequence protocol, which means it can be used like a python builtin list. And it provides features like lazy evaluation and cache.

**Parameters**

- **func** (*PagingGenerator[\_T]*) – A paging generator function, which takes offset<int> and limit<int> as inputs and returns a generator. The returned generator should yield the element user needs, and return the total count of the elements in the paging request.
- **limit** (*int*) – The page size of each paging request.

**insert**(*self, index, value*)

Insert object before index.

**Parameters**

- **index** (*int*) – Position of the LazyPagingList.
- **value** (*\_T*) – Element to be inserted into the LazyPagingList.

**Return type** None

**append**(*self, value*)

Append object to the end of the LazyPagingList.

**Parameters** **value** (*\_T*) – Element to be appended to the LazyPagingList.

**Return type** None

**reverse**(*self*)

Reverse the items of the LazyPagingList in place.

**Return type** None

**pop**(*self, index=-1*)

Return the item at index (default last) and remove it from the LazyPagingList.

**Parameters** **index** (*int*) – Position of the LazyPagingList.

**Returns** Element to be removed from the LazyPagingList.

**Return type** *\_T*

**index**(*self, value, start=0, stop=None*)

Return the first index of the value.

**Parameters**

- **value** (*Any*) – The value to be found.
- **start** (*int*) – The start index of the subsequence.
- **stop** (*Optional[int]*) – The end index of the subsequence.

**Raises** **ValueError** – When the value is not in the LazyPagingList

**Returns** The first index of the value.

**Return type** *int*

**count**(*self, value*)

Return the number of occurrences of value.

**Parameters** **value** (*Any*) – The value needs to be counted.

**Returns** The number of occurrences of value.

**Return type** *int*

**extend**(*self, values*)

Extend LazyPagingList by appending elements from the iterable.

**Parameters** **values** (*Iterable[\_T]*) – Elements to be extended into the LazyPagingList.

**Return type** *None*

## **graviti.manager.policy**

The implementation of the dataset policy.

## **Module Contents**

### **Classes**

<i>ObjectPolicyManager</i>	The basic structure of the object policy of the dataset.
<i>OSSObjectPolicyManager</i>	The basic structure of the object policy of the dataset stored in OSS.
<i>AZUREObjectPolicyManager</i>	The basic structure of the object policy of the dataset stored in AZURE.
<i>S3ObjectPolicyManager</i>	The basic structure of the object policy of the dataset stored in S3.

**class** `graviti.manager.policy.ObjectPolicyManager(dataset)`

The basic structure of the object policy of the dataset.

**Parameters** **dataset** (*graviti.manager.Dataset*) – Class Dataset instance.

**property** **prefix**(*self*)

Return the prefix of the put policy.

**Returns** The prefix of the put policy.

**Return type** *str*

**abstract** `get_object(self, key, _allow_retry=True)`

Get the object from graviti.

**Parameters**

- **key** (*str*) – The key of the file.
- **\_allow\_retry** (*bool*) – Whether requesting the get policy again is allowed.

**Raises** `NotImplementedError` – The method of the base class should not be called.

**Return type** `graviti.utility.UserResponse`

**abstract** `put_object(self, key, path, _allow_retry=True)`

Put the object to OSS.

**Parameters**

- **key** (*str*) – The key of the file.
- **path** (*pathlib.Path*) – The path of the file.
- **\_allow\_retry** (*bool*) – Whether requesting the put policy again is allowed.

**Raises** `NotImplementedError` – The method of the base class should not be called.

**Return type** `None`

**class** `graviti.manager.policy.OSSObjectPolicyManager(dataset)`

Bases: `ObjectPolicyManager`

The basic structure of the object policy of the dataset stored in OSS.

**Parameters** `dataset (graviti.manager.Dataset)` –

**get\_object** (*self, key, \_allow\_retry=True*)

Get the object from OSS.

**Parameters**

- **key** (*str*) – The key of the file.
- **\_allow\_retry** (*bool*) – Whether requesting the get policy again is allowed.

**Raises** `ResponseError` – If post response error.

**Returns** The response of OSS get object API.

**Return type** `graviti.utility.UserResponse`

**put\_object** (*self, key, path, \_allow\_retry=True*)

Put the object to OSS.

**Parameters**

- **key** (*str*) – The key of the file.
- **path** (*pathlib.Path*) – The path of the file.
- **\_allow\_retry** (*bool*) – Whether requesting the put policy again is allowed.

**Raises** `ResponseError` – If post response error.

**Return type** `None`

**class** `graviti.manager.policy.AZUREObjectPolicyManager(dataset)`

Bases: `ObjectPolicyManager`

The basic structure of the object policy of the dataset stored in AZURE.

Parameters **dataset** (*graviti.manager.Dataset*) –

**get\_object**(*self, key, \_allow\_retry=True*)

Get the object from AZURE.

**Parameters**

- **key** (*str*) – The key of the file.
- **\_allow\_retry** (*bool*) – Whether requesting the get policy again is allowed.

**Raises** *ResponseError* – If post response error.

**Returns** The response of AZURE get object API.

**Return type** *graviti.utility.UserResponse*

**put\_object**(*self, key, path, \_allow\_retry=True*)

Put the object to AZURE.

**Parameters**

- **key** (*str*) – The key of the file.
- **path** (*pathlib.Path*) – The path of the file.
- **\_allow\_retry** (*bool*) – Whether requesting the put policy again is allowed.

**Raises** *ResponseError* – If post response error.

**Return type** *None*

**class** *graviti.manager.policy.S3ObjectPolicyManager*(*dataset*)

Bases: *ObjectPolicyManager*

The basic structure of the object policy of the dataset stored in S3.

Parameters **dataset** (*graviti.manager.Dataset*) –

**get\_object**(*self, key, \_allow\_retry=True*)

Get the object from S3.

**Parameters**

- **key** (*str*) – The key of the file.
- **\_allow\_retry** (*bool*) – Whether requesting the get policy again is allowed.

**Raises** *ResponseError* – If post response error.

**Returns** The response of S3 get object API.

**Return type** *graviti.utility.UserResponse*

**put\_object**(*self, key, path, \_allow\_retry=True*)

Put the object to OSS.

**Parameters**

- **key** (*str*) – The key of the file.
- **path** (*pathlib.Path*) – The path of the file.
- **\_allow\_retry** (*bool*) – Whether requesting the put policy again is allowed.

**Raises** *ResponseError* – If post response error.

**Return type** *None*

### `graviti.manager.sheets`

The implementation of the Sheets.

## Module Contents

### Classes

---

<i>Sheets</i>	The basic structure of the Graviti sheets.
---------------	--

---

#### **class** `graviti.manager.sheets.Sheets`

Bases: `MutableMapping[str, graviti.dataframe.DataFrame]`, `graviti.utility.ReprMixin`

The basic structure of the Graviti sheets.

##### **keys**(*self*)

Return a new view of the keys in sheets.

**Returns** The keys in dict.

**Return type** `KeysView[str]`

##### **values**(*self*)

Return a new view of the values in sheets.

**Returns** The values in dict.

**Return type** `ValuesView[graviti.dataframe.DataFrame]`

##### **items**(*self*)

Return a new view of the (key, value) pairs in sheets.

**Returns** The (key, value) pairs in dict.

**Return type** `ItemsView[str, graviti.dataframe.DataFrame]`

### `graviti.manager.tag`

The implementation of the Tag and TagManager.

## Module Contents

### Classes

---

<i>Tag</i>	This class defines the structure of the tag of a commit.
<i>TagManager</i>	This class defines the operations on the tag in Graviti.

---

#### **class** `graviti.manager.tag.Tag`(*dataset, name, commit\_id*)

Bases: `graviti.manager.commit.NamedCommit`

This class defines the structure of the tag of a commit.

##### **Parameters**



- **dataset** (`graviti.manager.dataset.Dataset`) – Class Dataset instance.
- **name** (`str`) – The name of the tag.
- **commit\_id** (`Optional[str]`) – The commit id.

**class** `graviti.manager.tag.TagManager(dataset)`

This class defines the operations on the tag in Graviti.

**Parameters** **dataset** (`graviti.manager.dataset.Dataset`) – *Dataset* instance.

**create**(`self, name, revision=CURRENT_COMMIT`)

Create a tag for a commit.

**Parameters**

- **name** (`str`) – The tag name to be created for the specific commit.
- **revision** (`str`) – The information to locate the specific commit, which can be the commit id, the branch name, or the tag name. The default value is the current commit of the dataset.

**Raises** *NoCommitsError* – When create tags on the default branch without commit history.

**Returns** The *Tag* instance with the given name.

**Return type** *Tag*

**get**(`self, name`)

Get the certain tag with the given name.

**Parameters** **name** (`str`) – The required tag name.

**Raises** *ResourceNameError* – When the name is an empty string.

**Returns** The *Tag* instance with the given name.

**Return type** *Tag*

**list**(`self`)

List the information of tags.

**Returns** The LazyPagingList of *tags* instances.

**Return type** `graviti.manager.lazy.LazyPagingList[Tag]`

**delete**(`self, name`)

Delete a tag.

**Parameters** **name** (`str`) – The tag name to be deleted for the specific commit.

**Raises** *ResourceNameError* – When the name is an empty string.

**Return type** `None`

## graviti.openapi

OpenAPI module.

## Submodules

### graviti.openapi.branch

Interfaces about the branch.

## Module Contents

### Functions

<code>create_branch</code> (access_key, url, owner, dataset, *, name, revision)	Execute the OpenAPI <i>POST</i> <code>/v2/datasets/{owner}/{dataset}/branches</code> .
<code>list_branches</code> (access_key, url, owner, dataset, *, offset = None, limit = None)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}/branches</code> .
<code>get_branch</code> (access_key, url, owner, dataset, *, branch)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}/branches/{branch}</code> .
<code>delete_branch</code> (access_key, url, owner, dataset, *, branch)	Execute the OpenAPI <i>DELETE</i> <code>/v2/datasets/{owner}/{dataset}/branches/{branch}</code> .

`graviti.openapi.branch.create_branch`(access\_key, url, owner, dataset, \*, name, revision)  
Execute the OpenAPI *POST* `/v2/datasets/{owner}/{dataset}/branches`.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **name** (*str*) – The name of the branch.
- **revision** (*str*) – The information to locate the specific commit, which can be the commit id, the branch name, or the tag name.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

## Examples

```
>>> create_branch(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     name="branch-1",
...     revision="main"
... )
{
  "name": "main",
  "commit_id": "fde63f357daf46088639e9f57fd81cad",
  "parent_commit_id": "f68b1375454f459b8a486b8d1f4d9ddb",
  "title": "first commit",
  "description": "desc",
  "committer": "graviti-example",
  "committed_at": "2021-03-03T18:58:10Z"
}
```

`graviti.openapi.branch.list_branches(access_key, url, owner, dataset, *, offset=None, limit=None)`  
Execute the OpenAPI `GET /v2/datasets/{owner}/{dataset}/branches`.

### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **offset** (*Optional[int]*) – The offset of the page. The default value of this param in OpenAPIv2 is 0.
- **limit** (*Optional[int]*) – The limit of the page. The default value of this param in OpenAPIv2 is 128.

**Returns** The response of OpenAPI.

**Return type** Dict[str, Any]

## Examples

```
>>> list_branches(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST"
... )
{
  "branches": [
    {
      "name": "main",
      "commit_id": "fde63f357daf46088639e9f57fd81cad",
```

(continues on next page)

(continued from previous page)

```

        "parent_commit_id": "f68b1375454f459b8a486b8d1f4d9ddb",
        "title": "first commit",
        "description": "desc",
        "committer": "graviti-example",
        "committed_at": "2021-03-03T18:58:10Z"
    }
],
"offset": 0,
"record_size": 1,
"total_count": 1
}

```

`graviti.openapi.branch.get_branch(access_key, url, owner, dataset, *, branch)`  
 Execute the OpenAPI *GET* `/v2/datasets/{owner}/{dataset}/branches/{branch}`.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **branch** (*str*) – The name of the branch.

**Returns** The response of OpenAPI.

**Return type** Dict[*str*, *Any*]

#### Examples

```

>>> get_branch(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     branch="main",
... )
{
    "name": "main",
    "commit_id": "fde63f357daf46088639e9f57fd81cad",
    "parent_commit_id": "f68b1375454f459b8a486b8d1f4d9ddb",
    "title": "first commit",
    "description": "desc",
    "committer": "graviti-example",
    "committed_at": "2021-03-03T18:58:10Z"
}

```

`graviti.openapi.branch.delete_branch(access_key, url, owner, dataset, *, branch)`  
 Execute the OpenAPI *DELETE* `/v2/datasets/{owner}/{dataset}/branches/{branch}`.

#### Parameters

- **access\_key** (*str*) – User's access key.

- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **branch** (*str*) – The name of the branch.

**Return type** None

## Examples

```
>>> delete_branch(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     branch="branch-1",
... )
```

## graviti.openapi.commit

Interfaces about the commit.

## Module Contents

### Functions

<a href="#"><i>commit_draft</i></a> ( <i>access_key</i> , <i>url</i> , <i>owner</i> , <i>dataset</i> , *, <i>draft_number</i> , <i>title</i> , <i>description</i> = None)	Execute the OpenAPI <i>POST</i> <i>/v2/datasets/{owner}/{dataset}/commits</i> .	<i>POST</i>
<a href="#"><i>list_commits</i></a> ( <i>access_key</i> , <i>url</i> , <i>owner</i> , <i>dataset</i> , *, <i>revision</i> = None, <i>offset</i> = None, <i>limit</i> = None)	Execute the OpenAPI <i>GET</i> <i>/v2/datasets/{owner}/{dataset}/commits</i> .	<i>GET</i>
<a href="#"><i>get_commit</i></a> ( <i>access_key</i> , <i>url</i> , <i>owner</i> , <i>dataset</i> , *, <i>commit_id</i> )	Execute the OpenAPI <i>GET</i> <i>/v2/datasets/{owner}/{dataset}/commits/{commit_id}</i> .	<i>GET</i>
<a href="#"><i>get_revision</i></a> ( <i>access_key</i> , <i>url</i> , <i>owner</i> , <i>dataset</i> , *, <i>revision</i> )	Execute the OpenAPI <i>GET</i> <i>/v2/datasets/{owner}/{dataset}/revisions/{revision}</i> .	<i>GET</i>

**graviti.openapi.commit.commit\_draft**(*access\_key*, *url*, *owner*, *dataset*, \*, *draft\_number*, *title*, *description*=None)

Execute the OpenAPI *POST* */v2/datasets/{owner}/{dataset}/commits*.

### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **title** (*str*) – The draft title.

- **description** (*Optional[str]*) – The draft description.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

### Examples

```
>>> commit_draft(  
...     "ACCESSKEY-*****",  
...     "https://api.graviti.com",  
...     "graviti-example",  
...     "MNIST",  
...     draft_number=2,  
...     title="commit-2",  
... )  
{  
    "commit_id": "85c57a7f03804ccc906632248dc8c359",  
    "parent_commit_id": "784ba0d3bf0a41f6a7bfd771d8c00fcb",  
    "title": "upload data",  
    "description": "",  
    "committer": "graviti-example",  
    "committed_at": "2021-03-03T18:58:10Z"  
}
```

`graviti.openapi.commit.list_commits(access_key, url, owner, dataset, *, revision=None, offset=None, limit=None)`

Execute the OpenAPI *GET /v2/datasets/{owner}/{dataset}/commits*.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **revision** (*Optional[str]*) – The information to locate the specific commit, which can be the commit id, the branch name, or the tag name.
- **offset** (*Optional[int]*) – The offset of the page. The default value of this param in OpenAPIv2 is 0.
- **limit** (*Optional[int]*) – The limit of the page. The default value of this param in OpenAPIv2 is 128.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

## Examples

```
>>> list_commits(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
... )
{
  "commits": [
    {
      "commit_id": "85c57a7f03804ccc906632248dc8c359",
      "parent_commitId": "784ba0d3bf0a41f6a7bfd771d8c00fcb",
      "title": "upload data",
      "description": "",
      "committer": "graviti-example",
      "committed_at": "2021-03-03T18:58:10Z"
    }
  ],
  "offset": 0,
  "record_size": 1,
  "total_count": 1
}
```

`graviti.openapi.commit.get_commit(access_key, url, owner, dataset, *, commit_id)`  
Execute the OpenAPI *GET* `/v2/datasets/{owner}/{dataset}/commits/{commit_id}`.

### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **commit\_id** (*str*) – The commit ID.

**Returns** The response of OpenAPI.

**Return type** Dict[*str*, *Any*]

## Examples

```
>>> get_commit(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     commit_id="85c57a7f03804ccc906632248dc8c359"
... )
{
  "commit_id": "85c57a7f03804ccc906632248dc8c359",
  "parent_commit_id": "784ba0d3bf0a41f6a7bfd771d8c00fcb",
  "title": "upload data",
```

(continues on next page)

(continued from previous page)

```

    "description": "",
    "committer": "graviti-example",
    "committed_at": "2021-03-03T18:58:10Z"
  }

```

**graviti.openapi.commit.get\_revision**(*access\_key*, *url*, *owner*, *dataset*, \*, *revision*)  
 Execute the OpenAPI *GET /v2/datasets/{owner}/{dataset}/revisions/{revision}*.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **revision** (*str*) – The information to locate the specific commit, which can be the commit id, the branch name, or the tag name.

**Returns** The response of OpenAPI.

**Return type** Dict[*str*, *Any*]

#### Examples

```

>>> get_revision(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "MNIST",
...     revision="branch-1"
... )
{
  "type": "BRANCH",
  "commit_id": "85c57a7f03804ccc906632248dc8c359",
  "parent_commit_id": "784ba0d3bf0a41f6a7bfd771d8c00fcb",
  "title": "upload data",
  "description": "",
  "committer": "graviti-example",
  "committed_at": "2021-03-03T18:58:10Z"
}

```

#### graviti.openapi.data

Interfaces about the data.



## Module Contents

### Functions

<code>list_draft_data</code> (access_key, url, owner, dataset, *, draft_number, sheet, columns = None, order_by = None, offset = None, limit = None)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}/data.</code>
<code>list_commit_data</code> (access_key, url, owner, dataset, *, commit_id, sheet, columns = None, order_by = None, offset = None, limit = None)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}/commits/{commit_id}/sheets/{sheet}/data.</code>
<code>update_data</code> (access_key, url, owner, dataset, *, draft_number, sheet, data)	Execute the OpenAPI <i>PATCH</i> <code>/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}/data.</code>
<code>add_data</code> (access_key, url, owner, dataset, *, draft_number, sheet, data, strategy_arguments = None)	Execute the OpenAPI <i>POST</i> <code>/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}/data.</code>
<code>delete_data</code> (access_key, url, owner, dataset, *, draft_number, sheet, record_keys)	Execute the OpenAPI <i>DELETE</i> <code>/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}/data.</code>
<code>get_policy</code> (access_key, url, owner, dataset, *, draft_number, sheet, is_internal = None, expired = None)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}/policy.</code>

`graviti.openapi.data.list_draft_data`(access\_key, url, owner, dataset, \*, draft\_number, sheet, columns=None, order\_by=None, offset=None, limit=None)  
Execute the OpenAPI *GET* `/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}/data.`

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The sheet name.
- **columns** (*Optional[str]*) – The string of column names separated by |. Multiple indexes can be expressed using .. None means to get all columns.
- **order\_by** (*Optional[str]*) – The string of column names separated by | whose order determines the precedence of the sort. The rest are sorted by `__record_key` first. Multiple indexes can be expressed using ..
- **offset** (*Optional[int]*) – The offset of the page. The default value of this param in OpenAPIv2 is 0.
- **limit** (*Optional[int]*) – The limit of the page. The default value of this param in OpenAPIv2 is 128.

**Returns** The response of OpenAPI.

**Return type** Dict[*str*, *Any*]

## Examples

```

>>> list_draft_data(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     draft_number = 1,
...     sheet = "train",
...     order_by = "filename|attribute.weather",
... )
{
  "data": [
    {
      "__record_key": "123750493121329585",
      "filename": "0000f77c-6257be58.jpg",
      "image": {
        "url": "https://content-store-prod-vers",
        "checksum": "dcc197970e607f7576d978972f6fb312911ce005"
      },
      "attribute": {
        "weather": "clear",
        "scene": "city street",
        "timeofday": "daytime"
      },
      "box2ds": [
        {
          "xmin": 1125.902264,
          "xmax": 1156.978645,
          "ymin": 133.184488,
          "ymax": 210.875445,
          "category": "traffic light",
          "attribute": {
            "occluded": false,
            "truncated": false,
            "trafficLightColor": "G"
          }
        },
        {
          "xmin": 1156.978645,
          "xmax": 1191.50796,
          "ymin": 136.637417,
          "ymax": 210.875443,
          "category": "traffic light",
          "attribute": {
            "occluded": false,
            "truncated": false,
            "trafficLightColor": "G"
          }
        }
      ],
      ...
    ]
  },

```

(continues on next page)

(continued from previous page)

```

        ...(total 128 items)
    ],
    "offset": 0,
    "record_size": 128,
    "total_count": 700000
}

```

`graviti.openapi.data.list_commit_data(access_key, url, owner, dataset, *, commit_id, sheet, columns=None, order_by=None, offset=None, limit=None)`

Execute the OpenAPI *GET* `/v2/datasets/{owner}/{dataset}/commits/{commit_id}/sheets/{sheet}/data`.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **commit\_id** (*str*) – The commit id.
- **sheet** (*str*) – The sheet name.
- **columns** (*Optional[str]*) – The string of column names separated by |. Multiple indexes can be expressed using .. None means to get all columns.
- **order\_by** (*Optional[str]*) – The string of column names separated by | whose order determines the precedence of the sort. The rest are sorted by `__record_key` first. Multiple indexes can be expressed using ..
- **offset** (*Optional[int]*) – The offset of the page. The default value of this param in OpenAPIv2 is 0.
- **limit** (*Optional[int]*) – The limit of the page. The default value of this param in OpenAPIv2 is 128.

**Returns** The response of OpenAPI.

**Return type** `Dict[str, Any]`

#### Examples

```

>>> list_commit_data(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     commit_id = "fde63f357daf46088639e9f57fd81cad",
...     sheet = "train",
...     order_by = "filename|attribute.weather",
... )
{
  "data": [
    {
      "__record_key": "123750493121329585",

```

(continues on next page)

(continued from previous page)

```

        "filename": "0000f77c-6257be58.jpg",
        "image": {
            "url": "https://content-store-prod-vers",
            "checksum": "dcc197970e607f7576d978972f6fb312911ce005"
        },
        "attribute": {
            "weather": "clear",
            "scene": "city street",
            "timeofday": "daytime"
        },
        "box2ds": [
            {
                "xmin": 1125.902264,
                "xmax": 1156.978645,
                "ymin": 133.184488,
                "ymax": 210.875445,
                "category": "traffic light",
                "attribute": {
                    "occluded": false,
                    "truncated": false,
                    "trafficLightColor": "G"
                }
            },
            {
                "xmin": 1156.978645,
                "xmax": 1191.50796,
                "ymin": 136.637417,
                "ymax": 210.875443,
                "category": "traffic light",
                "attribute": {
                    "occluded": false,
                    "truncated": false,
                    "trafficLightColor": "G"
                }
            }
        ],
        ...
    ],
    "offset": 0,
    "record_size": 128,
    "total_count": 70000
}

```

`graviti.openapi.data.update_data(access_key, url, owner, dataset, *, draft_number, sheet, data)`  
 Execute the OpenAPI *PATCH* `/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}/data`.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.

- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The sheet name.
- **data** (*Union[List[Dict[str, Any]], Tuple[Dict[str, Any], Ellipsis]*) – The update data.

**Return type** None

## Examples

```
>>> update_data(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "OxfordIIITPet",
...     draft_number = 1,
...     sheet = "train",
...     data = [
...         {
...             "__record_key": "123750493121329585",
...             "filename": "0000f77c-6257be58.jpg",
...             "image": {
...                 "checksum": "dcc197970e607f7576d978972f6fb312911ce005"
...             },
...             "attribute": {
...                 "weather": "clear",
...                 "scene": "city street",
...                 "timeofday": "daytime"
...             },
...         },
...         {
...             "__record_key": "123750493121329585",
...             "filename": "0000f77c-62c2a288.jpg",
...             "image": {
...                 "checksum": "dcc197970e607f7576d978972f6fb2a2881ce004"
...             },
...             "attribute": {
...                 "weather": "clear",
...                 "scene": "highway",
...                 "timeofday": "dawn/dusk"
...             },
...         },
...     ],
... )
```

`graviti.openapi.data.add_data(access_key, url, owner, dataset, *, draft_number, sheet, data, strategy_arguments=None)`

Execute the OpenAPI *POST* `/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}/data`.

### Parameters

- **access\_key** (*str*) – User's access key.

- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The sheet name.
- **data** (*Union[List[Dict[str, Any]], Tuple[Dict[str, Any], Ellipsis]*) – The update data.
- **strategy\_arguments** (*Optional[Dict[str, Any]]*) – Arguments required by the `__record_key` generation strategy of the sheet.

**Return type** None

### Examples

```
>>> add_data(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "OxfordIIITPet",
...     draft_number = 1,
...     sheet = "train",
...     data = [
...         {
...             "filename": "0000f77c-6257be58.jpg",
...             "image": {
...                 "checksum": "dcc197970e607f7576d978972f6fb312911ce005"
...             },
...             "attribute": {
...                 "weather": "clear",
...                 "scene": "city street",
...                 "timeofday": "daytime"
...             },
...         },
...         {
...             "filename": "0000f77c-62c2a288.jpg",
...             "image": {
...                 "checksum": "dcc197970e607f7576d978972f6fb2a2881ce004"
...             },
...             "attribute": {
...                 "weather": "clear",
...                 "scene": "highway",
...                 "timeofday": "dawn/dusk"
...             },
...         },
...     ],
... )
```

`graviti.openapi.data.delete_data(access_key, url, owner, dataset, *, draft_number, sheet, record_keys)`  
 Execute the OpenAPI `DELETE /v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}/data`.

### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The sheet name.
- **record\_keys** (*List[str]*) – The record keys of the data to be deleted.

**Return type** None

### Examples

```
>>> delete_data(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "OxfordIIITPet",
...     draft_number = 1,
...     sheet = "train",
...     record_keys = ["123750493121329585", "123750493121329586"],
... )
```

`graviti.openapi.data.get_policy(access_key, url, owner, dataset, *, draft_number, sheet, is_internal=None, expired=None)`

Execute the OpenAPI *GET /v2/datasets/{owner}/{dataset}/drafts/{draft\_number}/sheets/{sheet}/policy*.

### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The sheet name.
- **is\_internal** (*Optional[bool]*) – Whether to return the intranet upload address, the default value in the OpenAPI is False.
- **expired** (*Optional[int]*) – Token expiry time in seconds. It cannot be negative. The default value in the OpenAPI is 60. If it is greater than 300, it will be processed as 300.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

## Examples

```
>>> get_policy(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     draft_number = 1,
...     sheet = "train",
... )
{
  "result": {
    "OSSAccessKeyId": "OSSACCESSKEYID",
    "Signature": "QbkCDeZtX37gb2zoemel3VCxz3k=",
    "policy": "eyJjb25kaXRpb25zIjpbWyJzdGFydHMtd2l0aCIiIiw",
    "success_action_status": "200"
  },
  "extra": {
    "backend_type": "oss",
    "host": "https://content-store-fat-v",
    "object_prefix": ""
  },
  "expire_at": "2022-03-03T18:58:10Z"
}
```

## graviti.openapi.dataset

Interfaces about the dataset.

## Module Contents

## Functions

<code>create_dataset</code> (access_key, url, name, *, alias = "", config = None, with_draft = None)	Execute the OpenAPI <i>POST</i> <code>/v2/datasets</code> .
<code>get_dataset</code> (access_key, url, owner, dataset)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}</code> .
<code>list_datasets</code> (access_key, url, *, offset = None, limit = None)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets</code> .
<code>update_dataset</code> (access_key, url, owner, dataset, *, name = None, alias = None, default_branch = None)	Execute the OpenAPI <i>PATCH</i> <code>/v2/datasets/{owner}/{dataset}</code> .
<code>delete_dataset</code> (access_key, url, owner, dataset)	Execute the OpenAPI <i>DELETE</i> <code>/v2/datasets/{owner}/{dataset}</code> .

`graviti.openapi.dataset.create_dataset`(access\_key, url, name, \*, alias="", config=None, with\_draft=None)

Execute the OpenAPI *POST* `/v2/datasets`.

### Parameters



- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **name** (*str*) – Name of the dataset, unique for a user.
- **alias** (*str*) – Alias of the dataset, default is "".
- **config** (*Optional[str]*) – The auth storage config name.
- **with\_draft** (*Optional[bool]*) – Whether to create a draft after the dataset is created.  
The default value of this parameter in OpenAPIv2 is False.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

### Examples

```
>>> create_dataset(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "MNIST",
... )
{
  "id": "2bc95d506db2401b898067f1045d7f68",
  "name": "OxfordIIITPet",
  "alias": "",
  "default_branch": "main",
  "commit_id": None,
  "cover_url": "https://tutu.s3.cn-northwest-1.amazonaws.com.cn/",
  "created_at": "2021-03-03T18:58:10Z",
  "updated_at": "2021-03-03T18:58:10Z",
  "owner": "graviti-example",
  "is_public": false,
  "config": "exampleConfigName",
  "backend_type": "OSS"
}
```

`graviti.openapi.dataset.get_dataset(access_key, url, owner, dataset)`  
Execute the OpenAPI *GET /v2/datasets/{owner}/{dataset}*.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

## Examples

```
>>> get_dataset(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "OxfordIIITPet"
... )
{
  "id": "2bc95d506db2401b898067f1045d7f68",
  "name": "OxfordIIITPet",
  "alias": "Oxford-IIIT Pet",
  "default_branch": "main",
  "commit_id": "a0d4065872f245e4ad1d0d1186e3d397",
  "cover_url": "https://tutu.s3.cn-northwest-1.amazonaws.com.cn/",
  "created_at": "2021-03-03T18:58:10Z",
  "updated_at": "2021-03-03T18:58:10Z",
  "owner": "graviti-example",
  "is_public": false,
  "config": "exampleConfigName",
  "backend_type": "OSS"
}
```

`graviti.openapi.dataset.list_datasets(access_key, url, *, offset=None, limit=None)`  
Execute the OpenAPI *GET /v2/datasets*.

### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **offset** (*Optional[int]*) – The offset of the page. The default value of this param in OpenAPIv2 is 0.
- **limit** (*Optional[int]*) – The limit of the page. The default value of this param in OpenAPIv2 is 128.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

## Examples

```
>>> list_datasets("ACCESSKEY-*****", "https://api.graviti.com")
{
  "datasets": [
    {
      "id": "2bc95d506db2401b898067f1045d7f68",
      "name": "OxfordIIITPet",
      "alias": "Oxford-IIIT Pet",
      "default_branch": "main",
      "commit_id": "a0d4065872f245e4ad1d0d1186e3d397",
      "cover_url": "https://tutu.s3.cn-northwest-1.amazonaws.com.cn/",
      "created_at": "2021-03-03T18:58:10Z",
```

(continues on next page)

(continued from previous page)

```

        "updated_at": "2021-03-03T18:58:10Z",
        "owner": "graviti-example",
        "is_public": false,
        "config": "exampleConfigName",
        "backend_type": "OSS"
    }
],
"offset": 0,
"record_size": 1,
"total_count": 1
}

```

`graviti.openapi.dataset.update_dataset(access_key, url, owner, dataset, *, name=None, alias=None, default_branch=None)`

Execute the OpenAPI *PATCH* `/v2/datasets/{owner}/{dataset}`.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **name** (*Optional[str]*) – New name of the dataset, unique for a user.
- **alias** (*Optional[str]*) – New alias of the dataset.
- **default\_branch** (*Optional[str]*) – User's chosen branch.

**Returns** The response of OpenAPI.

**Return type** `Dict[str, Any]`

#### Examples

```

>>> update_dataset(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "OxfordIIITPet",
...     name="OxfordIIITPets",
...     alias="Oxford-IIIT Pet",
...     default_branch="main",
... )
{
  "id": "2bc95d506db2401b898067f1045d7f68",
  "name": "OxfordIIITPets",
  "alias": "Oxford-IIIT Pet",
  "default_branch": "main",
  "commit_id": "a0d4065872f245e4ad1d0d1186e3d397",
  "cover_url": "https://tutu.s3.cn-northwest-1.amazonaws.com.cn/",
  "created_at": "2021-03-03T18:58:10Z",
  "updated_at": "2021-03-04T18:58:10Z",

```

(continues on next page)

(continued from previous page)

```

"owner": "graviti-example",
"is_public": false,
"config": "exampleConfigName",
"backend_type": "OSS"
}

```

`graviti.openapi.dataset.delete_dataset(access_key, url, owner, dataset)`  
Execute the OpenAPI *DELETE* `/v2/datasets/{owner}/{dataset}`.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.

**Return type** None

#### Examples

```

>>> delete_dataset(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "OxfordIIITPet",
... )

```

### `graviti.openapi.draft`

Interfaces about the draft.

#### Module Contents

#### Functions

<code>create_draft</code> ( <i>access_key</i> , <i>url</i> , <i>owner</i> , <i>dataset</i> , *, <i>title</i> , <i>branch</i> = None, <i>description</i> = None)	Execute the OpenAPI <i>POST</i> <code>/v2/datasets/{owner}/{dataset}/drafts</code> .
<code>list_drafts</code> ( <i>access_key</i> , <i>url</i> , <i>owner</i> , <i>dataset</i> , *, <i>state</i> = None, <i>branch</i> = None, <i>offset</i> = None, <i>limit</i> = None)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}/drafts</code> .
<code>get_draft</code> ( <i>access_key</i> , <i>url</i> , <i>owner</i> , <i>dataset</i> , *, <i>draft_number</i> )	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}/drafts/{draft_number}</code> .
<code>update_draft</code> ( <i>access_key</i> , <i>url</i> , <i>owner</i> , <i>dataset</i> , *, <i>draft_number</i> , <i>state</i> = None, <i>title</i> = None, <i>description</i> = None)	Execute the OpenAPI <i>PATCH</i> <code>/v2/datasets/{owner}/{dataset}/drafts/{draft_number}</code> .

`graviti.openapi.draft.create_draft`(*access\_key*, *url*, *owner*, *dataset*, \*, *title*, *branch*=None, *description*=None)  
Execute the OpenAPI *POST* `/v2/datasets/{owner}/{dataset}/drafts`.

**Parameters**

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **title** (*str*) – The draft title.
- **branch** (*Optional[str]*) – The specified branch name. None means the default branch of the dataset.
- **description** (*Optional[str]*) – The draft description.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

**Examples**

```
>>> create_draft(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     title="draft-2",
...     branch="main",
... )
{
  "number": 2,
  "title": "draft-2",
  "description": "",
  "branch": "main",
  "state": "OPEN",
  "parent_commit_id": "85c57a7f03804ccc906632248dc8c359",
  "creator": "graviti-example",
  "created_at": "2021-03-03T18:58:10Z",
  "updated_at": "2021-03-03T18:58:10Z"
}
```

`graviti.openapi.draft.list_drafts(access_key, url, owner, dataset, *, state=None, branch=None, offset=None, limit=None)`

Execute the OpenAPI *GET /v2/datasets/{owner}/{dataset}/drafts*.

**Parameters**

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **state** (*Optional[str]*) – The draft state which includes “OPEN”, “CLOSED”, “COMMITTED”, “ALL” and None. None means listing open drafts.

- **branch** (*Optional[str]*) – The branch name. None means listing drafts from all branches.
- **offset** (*Optional[int]*) – The offset of the page. The default value of this param in OpenAPIv2 is 0.
- **limit** (*Optional[int]*) – The limit of the page. The default value of this param in OpenAPIv2 is 128.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

## Examples

```
>>> list_drafts(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
... )
{
  "drafts": [
    {
      "number": 2,
      "title": "draft-2",
      "description": "",
      "branch": "main",
      "state": "OPEN",
      "parent_commit_id": "85c57a7f03804ccc906632248dc8c359",
      "creator": "graviti-example",
      "created_at": "2021-03-03T18:58:10Z",
      "updated_at": "2021-03-03T18:58:10Z"
    }
  ],
  "offset": 0,
  "record_size": 1,
  "total_count": 1
}
```

`graviti.openapi.draft.get_draft(access_key, url, owner, dataset, *, draft_number)`

Execute the OpenAPI `GET /v2/datasets/{owner}/{dataset}/drafts/{draft_number}`.

### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – Number of the draft.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

## Examples

```
>>> get_draft(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "MNIST",
...     "graviti-example",
...     draft_number=2,
... )
{
  "number": 2,
  "title": "draft-2",
  "description": "",
  "branch": "main",
  "state": "OPEN",
  "parent_commit_id": "85c57a7f03804ccc906632248dc8c359",
  "creator": "graviti-example",
  "created_at": "2021-03-03T18:58:10Z",
  "updated_at": "2021-03-03T18:58:10Z"
}
```

`graviti.openapi.draft.update_draft(access_key, url, owner, dataset, *, draft_number, state=None, title=None, description=None)`

Execute the OpenAPI *PATCH* `/v2/datasets/{owner}/{dataset}/drafts/{draft_number}`.

### Parameters

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The updated draft number.
- **state** (*Optional[str]*) – The updated draft state which could be “CLOSED” or None. Where None means no change in state.
- **title** (*Optional[str]*) – The draft title.
- **description** (*Optional[str]*) – The draft description.

**Returns** The response of OpenAPI.

**Return type** `Dict[str, Any]`

## Examples

Update the title or description of the draft:

```
>>> update_draft(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "MNIST",
...     draft_number=2,
...     title="draft-3"
```

(continues on next page)

(continued from previous page)

```

... )
{
  "number": 2,
  "title": "draft-3",
  "description": "",
  "branch": "main",
  "state": "OPEN",
  "parent_commit_id": "85c57a7f03804ccc906632248dc8c359",
  "creator": "graviti-example",
  "created_at": "2021-03-03T18:58:10Z",
  "updated_at": "2021-03-04T18:58:10Z"
}

```

Close the draft:

```

>>> update_draft(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "MNIST",
...     draft_number=2,
...     state="CLOSED"
... )
{
  "number": 2,
  "title": "draft-3",
  "description": "",
  "branch": "main",
  "state": "CLOSED",
  "parent_commit_id": "85c57a7f03804ccc906632248dc8c359",
  "creator": "graviti-example",
  "created_at": "2021-03-03T18:58:10Z",
  "updated_at": "2021-03-05T18:58:10Z"
}

```

## graviti.openapi.policy

Interfaces about the dataset policy.

## Module Contents

### Functions

<code>get_object_policy</code> ( <i>access_key</i> , <i>url</i> , <i>owner</i> , <i>dataset</i> , *, <i>actions</i> , <i>is_internal</i> = None, <i>expired</i> = None)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}/policy/object</code> .
--	--

`graviti.openapi.policy.get_object_policy`(*access\_key*, *url*, *owner*, *dataset*, \*, *actions*, *is\_internal*=None, *expired*=None)

Execute the OpenAPI *GET* `/v2/datasets/{owner}/{dataset}/policy/object`.

#### Parameters



- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **actions** (*str*) – The specific actions including “GET” and “PUT”. Supports multiple actions, which need to be separated by |, like “GET|PUT”.
- **is\_internal** (*Optional[bool]*) – Whether to return the intranet upload address, the default value in the OpenAPI is False.
- **expired** (*Optional[int]*) – Token expiry time in seconds. It cannot be negative.

**Returns** The response of OpenAPI.

**Return type** Dict[str, Any]

## Examples

Request permission to get dataset data from OSS:

```
>>> get_policy(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     actions="GET",
... )
{
  "backend_type": "OSS",
  "expire_at": "2022-07-12T06:07:52Z",
  "policy": {
    "AccessKeyId": "LTAI4FjgXD3yFJUasdasd",
    "AccessKeySecret": "LTAI4FjgXD3yFJJKasdad",
    "SecurityToken": "CAISrgJ1q6Ft5B2yfSjIr5bkKILDaseqw",
    "bucket": "content-store-dev",
    "endpoint": "content-store-dev.oss-cn-qingdao.aliyuncs.com"
  }
}
```

Request permission to put dataset data to OSS:

```
>>> get_policy(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     actions="PUT",
... )
{
  "backend_type": "OSS",
  "expire_at": "2022-07-12T06:07:52Z",
  "policy": {
    "AccessKeyId": "LTAI4FjgXD3yFJUasdasd",
```

(continues on next page)

(continued from previous page)

```

    "AccessKeySecret": "LTAI4FjgXD3yFJJKasdad",
    "SecurityToken": "CAISrgJ1q6Ft5B2yfSjIr5bkKILdaseqw",
    "bucket": "content-store-dev",
    "endpoint": "content-store-dev.oss-cn-qingdao.aliyuncs.com",
    "prefix": "051dd0676cc74f548a7e9b7ace45c26b/"
  }
}

```

Request permission to get dataset data from AZURE:

```

>>> get_policy(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     actions="GET",
... )
{
  "backend_type": "AZURE",
  "expire_at": "2022-07-12T06:07:52Z",
  "policy": {
    "container_name": "graviti210304",
    "account_name": "gra220303",
    "sas_param": "se=2022-07-21T10%3A07Z&sig=*****",
    "endpoint_prefix": "https://gra220303.blob.core.window.net/graviti210304/"
  }
}

```

Request permission to put dataset data to AZURE:

```

>>> get_policy(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     actions="PUT",
... )
{
  "backend_type": "AZURE",
  "expire_at": "2022-07-12T06:07:52Z",
  "policy": {
    "container_name": "graviti210304",
    "account_name": "gra220303",
    "prefix": "examplePrefix/",
    "sas_param": "se=2022-07-21T10%3A07Z&sig=*****",
    "endpoint_prefix": "https://gra220303.blob.core.window.net/graviti210304/"
  }
}

```

Request permission to get dataset data from S3:

```

>>> get_policy(
...     "ACCESSKEY-*****",

```

(continues on next page)

(continued from previous page)

```

...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     actions="GET",
... )
{
    "backend_type": "S3",
    "expire_at": "2022-07-12T06:07:52Z",
    "policy": {
        "AccessKeyId": "ASIAQHT*****",
        "AccessKeySecret": "Y6x2a2cHI1Jdx*****",
        "SecurityToken": "FwoGZXIvYXdzEH0aDGYBu*****",
        "bucket": "fat-dataplatform",
        "endpoint": "s3.amazonaws.com",
        "region": "us-west-1"
    }
}

```

Request permission to put dataset data to S3:

```

>>> get_policy(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     actions="PUT",
... )
{
    "backend_type": "S3",
    "expire_at": "2022-07-12T06:07:52Z",
    "policy": {
        "AccessKeyId": "ASIAQHT*****",
        "AccessKeySecret": "Y6x2a2cHI1Jdx*****",
        "SecurityToken": "FwoGZXIvYXdzEH0aDGYBu*****",
        "bucket": "fat-dataplatform",
        "endpoint": "s3.amazonaws.com",
        "prefix": "051dd0676cc74f548a7e9b7ace45c26b/",
        "region": "us-west-1"
    }
}

```

### graviti.openapi.requests

The basic concepts and methods of the Graviti OpenAPI.

## Module Contents

### Functions

---

<code>do(method, url, **kwargs)</code>	Send a request.
<code>open_api_do(method, access_key, url, **kwargs)</code>	Send a request to the Graviti OpenAPI.

---

### Attributes

---

<code>RESPONSE_ERROR_DISTRIBUTOR</code>
---

---

`graviti.openapi.requests.RESPONSE_ERROR_DISTRIBUTOR`

`graviti.openapi.requests.do(method, url, **kwargs)`  
Send a request.

#### Parameters

- **method** (*str*) – The method of the request.
- **url** (*str*) – The URL of the request.
- **\*\*kwargs** – Extra keyword arguments to send in the GET request.
- **kwargs** (*Any*) –

**Returns** Response of the request.

**Return type** requests.models.Response

`graviti.openapi.requests.open_api_do(method, access_key, url, **kwargs)`  
Send a request to the Graviti OpenAPI.

#### Parameters

- **method** (*str*) – The method of the request.
- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **\*\*kwargs** – Extra keyword arguments to send in the POST request.
- **kwargs** (*Any*) –

**Raises** [ResponseError](#) – When the status code OpenAPI returns is unexpected.

**Returns** Response of the request.

**Return type** requests.models.Response

## graviti.openapi.schema

Interfaces about the schema.

## Module Contents

### Functions

---

<code>update_schema</code>	<code>(access_key, url, owner, dataset, *, draft_number, sheet, patch = None, _schema, _avro_schema, _arrow_schema = None)</code>	Execute the OpenAPI <i>PATCH</i> <code>/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}/schema</code> .
----------------------------	---	--

---

`graviti.openapi.schema.update_schema`(*access\_key*, *url*, *owner*, *dataset*, \*, *draft\_number*, *sheet*, *patch=None*, *\_schema*, *\_avro\_schema=None*, *\_arrow\_schema=None*)

Execute the OpenAPI *PATCH* `/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}/schema`.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The sheet name.
- **patch** (*Optional[List[Dict[str, Any]]]*) – The list of patch operations which describe the changes to the json portex schema.
- **\_schema** (*str*) –
- **\_avro\_schema** (*str*) –
- **\_arrow\_schema** (*Optional[str]*) –

**Return type** None

### Examples

```
>>> update_schema(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     draft_number = 1,
...     sheet = "train",
...     _schema = '{"imports": [{"repo": "https://github.com/Project-OpenBytes/
↪portex-standard@main", "types": [{"name": "file.Image"}]}, {"type": "record",
↪"fields": [{"name": "filename", "type": "string"}, {"name": "image", "type":
↪"file.Image"}]}]',
...     _avro_schema = '{"type": "record", "name": "root", "namespace": "cn.graviti.
↪portex", "aliases": [], "fields": [{"name": "filename", "type": "string"}, {"name
↪": "image", "type": {"type": "record", "name": "image", "namespace": "cn.graviti.
↪portex.root", "aliases": [], "fields": [{"name": "checksum", "type": [null,
↪"string"]}]}]}',
```

(continued from previous page)

... )

## graviti.openapi.search

Interfaces about the search.

## Module Contents

### Functions

---

<code>create_search</code>	( <code>access_key</code> , <code>url</code> , <code>owner</code> , <code>dataset</code> , *, <code>commit_id</code> , <code>sheet</code> , <code>criteria</code> , <code>search_id</code> = None, <code>offset</code> = None, <code>limit</code> = None)	Execute the OpenAPI <i>POST</i> <code>/v2/datasets/{owner}/{dataset}/commits/{commit_id}/sheets/{sheet}/search</code> .
----------------------------	---	---

---

`graviti.openapi.search.create_search`(`access_key`, `url`, `owner`, `dataset`, \*, `commit_id`, `sheet`, `criteria`, `search_id`=None, `offset`=None, `limit`=None)  
 Execute the OpenAPI *POST* `/v2/datasets/{owner}/{dataset}/commits/{commit_id}/sheets/{sheet}/search`.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **commit\_id** (*str*) – The commit id.
- **sheet** (*str*) – The sheet name.
- **criteria** (*Dict*[*str*, *Any*]) – The criteria of the search.
- **search\_id** (*Optional*[*str*]) – The search id of this search.
- **offset** (*Optional*[*int*]) – The offset of the page. The default value of this param in OpenAPIv2 is 0.
- **limit** (*Optional*[*int*]) – The limit of the page. The default value of this param in OpenAPIv2 is to get all search results.

**Returns** The response of OpenAPI.

**Return type** *Dict*[*str*, *Any*]

## Examples

```

>>> create_search(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "BDD100K",
...     commit_id = "fde63f357daf46088639e9f57fd81cad",
...     sheet = "train",
...     criteria = {
...         "where": {
...             "$or": [
...                 {
...                     "$eq": ["$.filename", "0000f77c-6257be58.jpg"],
...                 },
...                 {
...                     "$and": [
...                         {
...                             "$eq": ["$.attribute.weather", "clear"]
...                         },
...                         {
...                             "$eq": ["$.attribute.timeofday", "daytime"]
...                         }
...                     ]
...                 }
...             ]
...         },
...         "offset": 0,
...         "limit": 128
...     }
... )
{
  "data": [
    {
      "__record_key": "123750493121329585",
      "filename": "0000f77c-6257be58.jpg",
      "image": {
        "url": "https://content-store-prod-vers",
        "checksum": "dcc197970e607f7576d978972f6fb312911ce005"
      },
      "attribute": {
        "weather": "clear",
        "scene": "city street",
        "timeofday": "daytime"
      }
    },
    ...(total 128 items)
  ],
  "search_id": "5c7de503c88446e8b37a258f71783d7d"
}

```

Use search\_id to avoid creating new search.

```

>>> create_search(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "BDD100K",
...     commit_id = "fde63f357daf46088639e9f57fd81cad",
...     sheet = "train",
...     criteria = {"where": {"$eq": [{"$.filename", "0000f77c-6257be58.jpg"]}}}
...     search_id = "5c7de503c88446e8b37a258f71783d7d"
... )
{
  "data": [
    {
      "__record_key": "123750493121329585",
      "filename": "0000f77c-6257be58.jpg",
      "image": {
        "url": "https://content-store-prod-vers",
        "checksum": "dcc197970e607f7576d978972f6fb312911ce005"
      },
      "attribute": {
        "weather": "clear",
        "scene": "city street",
        "timeofday": "daytime"
      }
    },
    ...(total 128 items)
  ],
  "search_id": "5c7de503c88446e8b37a258f71783d7d"
}

```

## graviti.openapi.sheet

Interfaces about the sheet.

## Module Contents

### Functions

<a href="#"><code>create_sheet</code></a> (access_key, url, owner, dataset, *, draft_number, name, schema, _avro_schema, _arrow_schema = None, record_key_strategy = None)	Execute the OpenAPI <i>POST</i> <code>/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets</code> .
<a href="#"><code>list_draft_sheets</code></a> (access_key, url, owner, dataset, *, draft_number, with_record_count = None, offset = None, limit = None)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets</code> .
<a href="#"><code>list_commit_sheets</code></a> (access_key, url, owner, dataset, *, commit_id, with_record_count = None, offset = None, limit = None)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}/commits/{commit_id}/sheets</code> .

continues on next page



Table 1.44 – continued from previous page

<code>get_draft_sheet</code> (access_key, url, owner, dataset, *, draft_number, sheet, with_record_count = None, schema_format = None)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets</code> .
<code>get_commit_sheet</code> (access_key, url, owner, dataset, *, commit_id, sheet, with_record_count = None, schema_format = None)	Execute the OpenAPI <i>GET</i> <code>/v2/datasets/{owner}/{dataset}/commits/{commit_id}/sheets/{sheet}</code> .
<code>delete_sheet</code> (access_key, url, owner, dataset, *, draft_number, sheet)	Execute the OpenAPI <i>DELETE</i> <code>/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}</code> .

`graviti.openapi.sheet.create_sheet`(access\_key, url, owner, dataset, \*, draft\_number, name, schema, \_avro\_schema, \_arrow\_schema=None, record\_key\_strategy=None)  
Execute the OpenAPI *POST* `/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets`.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **name** (*str*) – The sheet name.
- **schema** (*str*) – The portex schema of the sheet.
- **record\_key\_strategy** (*Optional[str]*) – The `__record_key` generation strategy. If None, it is batch auto-increment sorting record key.
- **\_avro\_schema** (*str*) –
- **\_arrow\_schema** (*Optional[str]*) –

**Return type** None

#### Examples

```
>>> create_sheet(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     draft_number = 1,
...     name = "val",
...     schema = '{"imports": [{"repo": "https://github.com/Project-OpenBytes/
↪portex-standard@main", "types": [{"name": "file.Image"}]}], "type": "record",
↪"fields": [{"name": "filename", "type": "string"}, {"name": "image", "type":
↪"file.Image"}]}',
...     _avro_schema = '{"type": "record", "name": "root", "namespace": "cn.graviti.
↪portex", "aliases": [], "fields": [{"name": "filename", "type": "string"}, {"name
↪": "image", "type": {"type": "record", "name": "image", "namespace": "cn.graviti.
↪portex.root", "aliases": [], "fields": [{"name": "checksum", "type": [null,
↪"string"]}]}]}'
```

(continues on next page)

(continued from previous page)

```
... )
```

```
graviti.openapi.sheet.list_draft_sheets(access_key, url, owner, dataset, *, draft_number,
                                         with_record_count=None, offset=None, limit=None)
```

Execute the OpenAPI *GET* `/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets`.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **with\_record\_count** (*Optional[bool]*) – Whether return the record count of each sheet. The default value of this param in OpenAPI is False.
- **offset** (*Optional[int]*) – The offset of the page. The default value of this param in OpenAPIv2 is 0.
- **limit** (*Optional[int]*) – The limit of the page. The default value of this param in OpenAPIv2 is 128.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

#### Examples

```
>>> list_draft_sheets(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     draft_number = 1,
... )
{
  "sheets": [
    {
      "name": "test",
      "created_at": "2021-03-03T18:58:10Z",
      "updated_at": "2021-03-04T18:58:10Z",
    },
    {
      "name": "trainval",
      "created_at": "2021-03-05T18:58:10Z",
      "updated_at": "2021-03-06T18:58:10Z",
    }
  ],
  "offset": 0,
  "record_size": 2,
  "total_count": 2
}
```

`graviti.openapi.sheet.list_commit_sheets(access_key, url, owner, dataset, *, commit_id, with_record_count=None, offset=None, limit=None)`

Execute the OpenAPI *GET* `/v2/datasets/{owner}/{dataset}/commits/{commit_id}/sheets`.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **commit\_id** (*str*) – The commit id.
- **with\_record\_count** (*Optional[bool]*) – Whether return the record count of each sheet. The default value of this param in OpenAPI is False.
- **offset** (*Optional[int]*) – The offset of the page. The default value of this param in OpenAPIv2 is 0.
- **limit** (*Optional[int]*) – The limit of the page. The default value of this param in OpenAPIv2 is 128.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

#### Examples

```
>>> list_commit_sheets(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     commit_id = "fde63f357daf46088639e9f57fd81cad",
... )
{
  "sheets": [
    {
      "name": "test",
      "created_at": "2021-03-03T18:58:10Z",
      "updated_at": "2021-03-04T18:58:10Z",
    },
    {
      "name": "trainval",
      "created_at": "2021-03-05T18:58:10Z",
      "updated_at": "2021-03-06T18:58:10Z",
    }
  ],
  "offset": 0,
  "record_size": 2,
  "total_count": 2
}
```

`graviti.openapi.sheet.get_draft_sheet(access_key, url, owner, dataset, *, draft_number, sheet, with_record_count=None, schema_format=None)`

Execute the OpenAPI *GET* `/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets`.

**Parameters**

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The sheet name.
- **with\_record\_count** (*Optional[bool]*) – Whether return the record count of each sheet. The default value of this param in OpenAPI is False.
- **schema\_format** (*Optional[str]*) – Fill “JSON”/“YAML” to determine whether the schema\_format of the returned schema is json or yaml. None means “JSON” format.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

**Examples**

```
>>> get_draft_sheet(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     draft_number = 1,
...     sheet = "sheet-2",
...     with_record_count=True,
... )
{
  "name": "trainval",
  "created_at": "2021-03-05T18:58:10Z",
  "updated_at": "2021-03-06T18:58:10Z",
  "record_count": 10000,
  "schema": '{"imports": [{"repo": "https://github.com/Project-OpenBytes/...'}
}
```

```
graviti.openapi.sheet.get_commit_sheet(access_key, url, owner, dataset, *, commit_id, sheet,
                                       with_record_count=None, schema_format=None)
```

Execute the OpenAPI *GET* /v2/datasets/{owner}/{dataset}/commits/{commit\_id}/sheets/{sheet}.

**Parameters**

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **commit\_id** (*str*) – The commit id..
- **sheet** (*str*) – The sheet name.

- **with\_record\_count** (*Optional[bool]*) – Whether return the record count of each sheet. The default value of this param in OpenAPI is False.
- **schema\_format** (*Optional[str]*) – Fill “JSON”/“YAML” to determine whether the schema\_format of the returned schema is json or yaml. None means “JSON” format.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

## Examples

```
>>> get_commit_sheet(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     commit_id = "fde63f357daf46088639e9f57fd81cad",
...     sheet = "sheet-2",
...     with_record_count=True,
... )
{
  "name": "trainval",
  "created_at": "2021-03-05T18:58:10Z",
  "updated_at": "2021-03-06T18:58:10Z",
  "record_count": 10000,
  "schema": '{"imports": [{"repo": "https://github.com/Project-OpenBytes/..."}]}'
}
```

`graviti.openapi.sheet.delete_sheet(access_key, url, owner, dataset, *, draft_number, sheet)`  
Execute the OpenAPI *DELETE* `/v2/datasets/{owner}/{dataset}/drafts/{draft_number}/sheets/{sheet}`.

### Parameters

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The name of the sheet to be deleted.

**Return type** None

## Examples

```
>>> delete_sheet(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     draft_number=1,
...     sheet="sheet-2"
... )
```

## graviti.openapi.tag

Interfaces about the tag.

## Module Contents

### Functions

<code>create_tag</code> (access_key, url, owner, dataset, *, name, revision)	Execute the OpenAPI	<i>POST</i>
	<i>/v2/datasets/{owner}/{dataset}/tags.</i>	
<code>list_tags</code> (access_key, url, owner, dataset, *, offset = None, limit = None)	Execute the OpenAPI	<i>GET</i>
	<i>/v2/datasets/{owner}/{dataset}/tags.</i>	
<code>get_tag</code> (access_key, url, owner, dataset, *, tag)	Execute the OpenAPI	<i>GET</i>
	<i>/v2/datasets/{owner}/{dataset}/tags/{tag}.</i>	
<code>delete_tag</code> (access_key, url, owner, dataset, *, tag)	Execute the OpenAPI	<i>DELETE</i>
	<i>/v2/datasets/{owner}/{dataset}/tags/{tag}.</i>	

`graviti.openapi.tag.create_tag`(access\_key, url, owner, dataset, \*, name, revision)  
Execute the OpenAPI *POST* */v2/datasets/{owner}/{dataset}/tags*.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **name** (*str*) – The tag name to be created for the specific commit.
- **revision** (*str*) – The information to locate the specific commit, which can be the commit id, the branch name, or the tag name.

**Returns** The response of OpenAPI.

**Return type** Dict[str, Any]

## Examples

```
>>> create_tag(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     name="tag-2",
...     revision="986d8ea00da842ed85dd5d5cd14b5aef"
... )
{
  "name": "tag-2",
  "commit_id": "986d8ea00da842ed85dd5d5cd14b5aef",
  "parent_commit_id": "a0d4065872f245e4ad1d0d1186e3d397",
  "title": "commit-1",
  "description": "",
  "committer": "graviti-example",
  "committed_at": "2021-03-03T18:58:10Z"
}
```

`graviti.openapi.tag.list_tags(access_key, url, owner, dataset, *, offset=None, limit=None)`  
 Execute the OpenAPI *GET /v2/datasets/{owner}/{dataset}/tags*.

### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **offset** (*Optional[int]*) – The offset of the page. The default value of this param in OpenAPIv2 is 0.
- **limit** (*Optional[int]*) – The limit of the page. The default value of this param in OpenAPIv2 is 128.

**Returns** The response of OpenAPI.

**Return type** Dict[str, Any]

## Examples

```
>>> list_tags(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST"
... )
{
  "tags": [
    {
      "name": "tag-2",
      "commit_id": "986d8ea00da842ed85dd5d5cd14b5aef",
```

(continues on next page)

(continued from previous page)

```

        "parent_commit_id": "a0d4065872f245e4ad1d0d1186e3d397",
        "title": "commit-1",
        "description": "",
        "committer": "graviti-example",
        "committed_at": "2021-03-03T18:58:10Z"
    }
],
"offset": 0,
"record_size": 1,
"total_count": 1
}

```

`graviti.openapi.tag.get_tag(access_key, url, owner, dataset, *, tag)`  
 Execute the OpenAPI *GET* `/v2/datasets/{owner}/{dataset}/tags/{tag}`.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **tag** (*str*) – The name of the tag to be got.

**Returns** The response of OpenAPI.

**Return type** Dict[*str*, *Any*]

#### Examples

```

>>> get_tag(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     tag="tag-2"
... )
{
  "name": "tag-2",
  "commit_id": "986d8ea00da842ed85dd5d5cd14b5aef",
  "parent_commit_id": "a0d4065872f245e4ad1d0d1186e3d397",
  "title": "commit-1",
  "description": "",
  "committer": "graviti-example",
  "committed_at": "2021-03-03T18:58:10Z"
}

```

`graviti.openapi.tag.delete_tag(access_key, url, owner, dataset, *, tag)`  
 Execute the OpenAPI *DELETE* `/v2/datasets/{owner}/{dataset}/tags/{tag}`.

#### Parameters

- **access\_key** (*str*) – User's access key.



- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **tag** (*str*) – The name of the tag to be deleted.

**Return type** None

### Examples

```
>>> delete_tag(
...     "ACCESSKEY-*****",
...     "https://api.graviti.com",
...     "graviti-example",
...     "MNIST",
...     tag="tag-2"
... )
```

## graviti.openapi.user

Interfaces about the user.

### Module Contents

#### Functions

---

<code>get_current_user</code> ( <i>access_key</i> , <i>url</i> )	Execute the OpenAPI <i>GET /v2/current-user</i> .
--	---

---

`graviti.openapi.user.get_current_user`(*access\_key*, *url*)  
Execute the OpenAPI *GET /v2/current-user*.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.

**Returns** The response of OpenAPI.

**Return type** Dict[str, *Any*]

### Examples

```
>>> get_current_user("ACCESSKEY-*****", "https://api.graviti.com")
{
  "id": "41438e9df9a82a194e1e76cc31c1d8d4",
  "nickname": "czt ual",
  "email": "*****@graviti.com",
  "mobile": null,
  "description": "",
```

(continues on next page)

(continued from previous page)

```
"workspace": "graviti-example",
"team": null
}
```

### `graviti.operation`

Operation module.

### Submodules

#### `graviti.operation.common`

Common tools.

### Module Contents

#### Functions

---

<code>get_schema(schema)</code>	Get portex schema, avro schema and arrow schema.
---------------------------------	--

---

`graviti.operation.common.get_schema(schema)`  
Get portex schema, avro schema and arrow schema.

**Parameters** `schema` (`graviti.portex.PortexType`) – The portex schema.

**Returns** The tuple of portex schema, avro schema and arrow schema.

**Return type** Tuple[str, str, str]

#### `graviti.operation.frame`

Definitions of different operations on a DataFrame.

### Module Contents

#### Classes

---

<code>DataFrameOperation</code>	This class defines the basic method of the operation on a DataFrame.
<code>DataOperation</code>	This class defines the basic method of the data operation on a DataFrame.
<code>AddData</code>	This class defines the operation that add data to a DataFrame.
<code>UpdateSchema</code>	This class defines the operation that update the schema of a DataFrame.

---

continues on next page

Table 1.48 – continued from previous page

<i>UpdateData</i>	This class defines the operation that updates the data of a DataFrame.
<i>DeleteData</i>	This class defines the operation that delete the data of a DataFrame.

**class** graviti.operation.frame.DataFrameOperation

This class defines the basic method of the operation on a DataFrame.

**get\_file\_count**(*self*)

Get the file amount to be uploaded.

**Returns** The file amount to be uploaded.

**Return type** int

**get\_data\_count**(*self*)

Get the data amount to be uploaded.

**Returns** The data amount to be uploaded.

**Return type** int

**abstract do**(*self*, *access\_key*, *url*, *owner*, *dataset*, \*, *draft\_number*, *sheet*, *jobs*, *data\_pbar*, *file\_pbar*, *object\_policy\_manager*)

Execute the OpenAPI create sheet.

**Parameters**

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The sheet name.
- **jobs** (*int*) – The number of the max workers in multi-thread operation.
- **data\_pbar** (*tqdm.tqdm*) – The process bar for uploading structured data.
- **file\_pbar** (*tqdm.tqdm*) – The process bar for uploading binary files.
- **object\_policy\_manager** (*graviti.manager.ObjectPolicyManager*) – The object policy manager of the dataset.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** None

**class** graviti.operation.frame.DataOperation(*data*)

Bases: *DataFrameOperation*

This class defines the basic method of the data operation on a DataFrame.

**Parameters** *data* (*graviti.dataframe.DataFrame*) –

**get\_file\_count**(*self*)

Get the file amount to be uploaded.

**Returns** The file amount to be uploaded.

**Return type** int

**get\_data\_count**(*self*)

Get the data amount to be uploaded.

**Returns** The data amount to be uploaded.

**Return type** int

**class** graviti.operation.frame.**AddData**(*data*)

Bases: [DataOperation](#)

This class defines the operation that add data to a DataFrame.

**Parameters** *data* (*graviti.dataframe.DataFrame*) – The data to be added.

**do**(*self*, *access\_key*, *url*, *owner*, *dataset*, \*, *draft\_number*, *sheet*, *jobs*, *data\_pbar*, *file\_pbar*, *object\_policy\_manager*)

Execute the OpenAPI add data.

**Parameters**

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The sheet name.
- **jobs** (*int*) – The number of the max workers in multi-thread operation.
- **data\_pbar** (*tqdm.tqdm*) – The process bar for uploading structured data.
- **file\_pbar** (*tqdm.tqdm*) – The process bar for uploading binary files.
- **object\_policy\_manager** (*graviti.manager.ObjectPolicyManager*) – The object policy manager of the dataset.

**Return type** None

**class** graviti.operation.frame.**UpdateSchema**(*schema*)

Bases: [DataFrameOperation](#)

This class defines the operation that update the schema of a DataFrame.

**Parameters** *schema* (*graviti.portex.PortexType*) – New portex schema after updated.

**do**(*self*, *access\_key*, *url*, *owner*, *dataset*, \*, *draft\_number*, *sheet*, *jobs*, *data\_pbar*, *file\_pbar*, *object\_policy\_manager*)

Execute the OpenAPI update schema.

**Parameters**

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The sheet name.
- **jobs** (*int*) – The number of the max workers in multi-thread operation.

- **data\_pbar** (*tqdm.tqdm*) – The process bar for uploading structured data.
- **file\_pbar** (*tqdm.tqdm*) – The process bar for uploading binary files.
- **object\_policy\_manager** (*graviti.manager.ObjectPolicyManager*) – The object policy manager of the dataset.

**Return type** None

**class** `graviti.operation.frame.UpdateData(data)`

Bases: `DataOperation`

This class defines the operation that updates the data of a DataFrame.

**Parameters** **data** (*graviti.dataframe.DataFrame*) – The data for updating.

**do**(*self, access\_key, url, owner, dataset, \*, draft\_number, sheet, jobs, data\_pbar, file\_pbar, object\_policy\_manager*)  
Execute the OpenAPI add data.

**Parameters**

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The sheet name.
- **jobs** (*int*) – The number of the max workers in multi-thread operation.
- **data\_pbar** (*tqdm.tqdm*) – The process bar for uploading structured data.
- **file\_pbar** (*tqdm.tqdm*) – The process bar for uploading binary files.
- **object\_policy\_manager** (*graviti.manager.ObjectPolicyManager*) – The object policy manager of the dataset.

**Return type** None

**class** `graviti.operation.frame.DeleteData(record_keys)`

Bases: `DataFrameOperation`

This class defines the operation that delete the data of a DataFrame.

**Parameters** **record\_keys** (*List[str]*) – The record keys of the data to be deleted.

**do**(*self, access\_key, url, owner, dataset, \*, draft\_number, sheet, jobs, data\_pbar, file\_pbar, object\_policy\_manager*)  
Execute the OpenAPI delete data.

**Parameters**

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.
- **sheet** (*str*) – The sheet name.

- **jobs** (*int*) – The number of the max workers in multi-thread operation.
- **data\_pbar** (*tqdm.tqdm*) – The process bar for uploading structured data.
- **file\_pbar** (*tqdm.tqdm*) – The process bar for uploading binary files.
- **object\_policy\_manager** (*graviti.manager.ObjectPolicyManager*) – The object policy manager of the dataset.

**Return type** None

## graviti.operation.sheet

Definitions of different operations about the sheet on a draft.

## Module Contents

### Classes

<a href="#"><i>SheetOperation</i></a>	This class defines the basic method of the operation about the sheet on a draft.
<a href="#"><i>CreateSheet</i></a>	This class defines the operation that create a sheet.
<a href="#"><i>DeleteSheet</i></a>	This class defines the operation that delete a sheet.

**class** graviti.operation.sheet.**SheetOperation**(*sheet*)

This class defines the basic method of the operation about the sheet on a draft.

**Parameters** **sheet** (*str*) – The sheet name.

**abstract do**(*self, access\_key, url, owner, dataset, \*, draft\_number*)

Execute the OpenAPI create sheet.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** None

**class** graviti.operation.sheet.**CreateSheet**(*sheet, schema*)

Bases: [\*SheetOperation\*](#)

This class defines the operation that create a sheet.

#### Parameters

- **sheet** (*str*) – The sheet name.
- **schema** (*graviti.portex.PortexRecordBase*) – The schema of the DataFrame.

**do**(*self*, *access\_key*, *url*, *owner*, *dataset*, \*, *draft\_number*)  
Execute the OpenAPI create sheet.

#### Parameters

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.

**Return type** None

**class** graviti.operation.sheet.DeleteSheet(*sheet*)

Bases: [SheetOperation](#)

This class defines the operation that delete a sheet.

**Parameters** **sheet** (*str*) –

**do**(*self*, *access\_key*, *url*, *owner*, *dataset*, \*, *draft\_number*)  
Execute the OpenAPI delete sheet.

#### Parameters

- **access\_key** (*str*) – User’s access key.
- **url** (*str*) – The URL of the graviti website.
- **owner** (*str*) – The owner of the dataset.
- **dataset** (*str*) – Name of the dataset, unique for a user.
- **draft\_number** (*int*) – The draft number.

**Return type** None

## graviti.paging

Paging module.

### Submodules

#### graviti.paging.factory

Paging list related class.

## Module Contents

### Classes

<a href="#"><i>LazyFactoryBase</i></a>	LazyFactoryBase is the base class of the lazy facotry.
<a href="#"><i>LazyFactory</i></a>	LazyFactory is a factory for requesting source data and creating paging lists.
<a href="#"><i>LazySubFactory</i></a>	LazySubFactory is a factory for creating paging lists.
<a href="#"><i>LazyLowerCaseFactory</i></a>	LazyLowerCaseFactory is a factory to handle the case insensitive data from graviti back-end.
<a href="#"><i>LazyLowerCaseSubFactory</i></a>	LazyLowerCaseSubFactory is a sub-factory to handle the case insensitive data.

**class** graviti.paging.factory.**LazyFactoryBase**

LazyFactoryBase is the base class of the lazy facotry.

**abstract** **create\_list**(*self*, *mapper*)

Create a paging list from the factory.

**Parameters** **mapper** (*Callable*[[*Any*], *\_T*]) – A callable object to convert every item in the pyarrow array.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** *graviti.paging.lists.PagingList*[*\_T*]

**abstract** **create\_mapped\_list**(*self*, *mapper*)

Create a paging list from the factory.

**Parameters** **mapper** (*Callable*[[*Any*], *\_T*]) – A callable object to convert every item in the pyarrow array.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** *graviti.paging.lists.MappedPagingList*[*\_T*]

**abstract** **create\_pyarrow\_list**(*self*)

Create a paging list from the factory.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** *graviti.paging.lists.PyArrowPagingList*[*Any*]

**class** graviti.paging.factory.**LazyFactory**(*total\_count*, *limit*, *getter*, *patype*)

Bases: [\*LazyFactoryBase\*](#)

LazyFactory is a factory for requesting source data and creating paging lists.

#### Parameters

- **total\_count** (*int*) – The total count of the elements in the paging lists.
- **limit** (*int*) – The size of each lazy load page.
- **getter** (*Callable*[[*int*, *int*], *Any*]) – A callable object to get the source data.
- **patype** (*pyarrow.DataType*) – The pyarrow *DataType* of the data in the factory.



## Examples

```
>>> import pyarrow as pa
>>> patype = pa.struct(
...     {
...         "remotePath": pa.string(),
...         "label": pa.struct({"CLASSIFICATION": pa.struct({"category": pa.
↪ string()})}),
...     }
... )
>>> TOTAL_COUNT = 1000
>>> def getter(offset: int, limit: int) -> List[Dict[str, Any]]:
...     stop = min(offset + limit, TOTAL_COUNT)
...     return [
...         {
...             "remotePath": f"{i:06}.jpg",
...             "label": {"CLASSIFICATION": {"category": "cat" if i % 2 else "dog"}}
↪ ,
...         }
...         for i in range(offset, stop)
...     ]
...
>>> factory = LazyFactory(TOTAL_COUNT, 128, getter, patype)
>>> paths = factory["remotePath"].create_pyarrow_list()
>>> categories = factory["label"]["CLASSIFICATION"]["category"].create_pyarrow_
↪ list()
>>> len(paths)
1000
>>> list(paths)
[<pyarrow.StringScalar: '000000.jpg'>,
 <pyarrow.StringScalar: '000001.jpg'>,
 <pyarrow.StringScalar: '000002.jpg'>,
 <pyarrow.StringScalar: '000003.jpg'>,
 <pyarrow.StringScalar: '000004.jpg'>,
 <pyarrow.StringScalar: '000005.jpg'>,
 ...
 <pyarrow.StringScalar: '000999.jpg'>]
>>> len(categories)
1000
>>> list(categories)
[<pyarrow.StringScalar: 'dog'>,
 <pyarrow.StringScalar: 'cat'>,
 <pyarrow.StringScalar: 'dog'>,
 <pyarrow.StringScalar: 'cat'>,
 <pyarrow.StringScalar: 'dog'>,
 ...
 <pyarrow.StringScalar: 'cat'>]
```

**get\_array**(*self*, *pos*, *keys*)

Get the array from the factory.

### Parameters

- **pos** (*int*) – The page number.
- **keys** (*Tuple[str, Ellipsis]*) – The keys to access the array from factory.

**Returns** The requested pyarrow array.

**Return type** `pyarrow.Array`

**create\_list**(*self, mapper*)

Create a paging list from the factory.

**Parameters** **mapper** (*Callable[[Any], \_T]*) – A callable object to convert every item in the pyarrow array.

**Returns** A paging list created from the factory.

**Return type** `graviti.paging.lists.PagingList[_T]`

**create\_mapped\_list**(*self, mapper*)

Create a paging list from the factory.

**Parameters** **mapper** (*Callable[[Any], \_T]*) – A callable object to convert every item in the pyarrow array.

**Returns** A paging list created from the factory.

**Return type** `graviti.paging.lists.MappedPagingList[_T]`

**create\_pyarrow\_list**(*self*)

Create a paging list from the factory.

**Returns** A paging list created from the factory.

**Return type** `graviti.paging.lists.PyArrowPagingList[Any]`

**get\_page\_lengths**(*self*)

A Generator which generates the length of the pages in the factory.

**Yields** The page lengths.

**Return type** `Iterator[int]`

**get\_offsets**(*self*)

Get the Offsets instance created by the total\_count and limit of this factory.

**Returns** The Offsets instance created by the total\_count and limit of this factory.

**Return type** `graviti.paging.offset.Offsets`

**class** `graviti.paging.factory.LazySubFactory`(*factory, keys, patype*)

Bases: `LazyFactoryBase`

LazySubFactory is a factory for creating paging lists.

**Parameters**

- **factory** (`LazyFactory`) – The source LazyFactory instance.
- **keys** (*Tuple[str, Ellipsis]*) – The keys to access the array from the source LazyFactory.
- **patype** (`pyarrow.DataType`) – The pyarrow DataType of the data in the sub-factory.

**create\_list**(*self, mapper*)

Create a paging list from the factory.

**Parameters** **mapper** (*Callable[[Any], \_T]*) – A callable object to convert every item in the pyarrow array.

**Returns** A paging list created from the factory.

**Return type** *graviti.paging.lists.PagingList*[\_T]

**create\_mapped\_list**(*self*, *mapper*)

Create a paging list from the factory.

**Parameters** **mapper** (*Callable*[[*Any*], \_T]) – A callable object to convert every item in the pyarrow array.

**Returns** A paging list created from the factory.

**Return type** *graviti.paging.lists.MappedPagingList*[\_T]

**create\_pyarrow\_list**(*self*)

Create a paging list from the factory.

**Returns** A paging list created from the factory.

**Return type** *graviti.paging.lists.PyArrowPagingList*[*Any*]

**class** *graviti.paging.factory.LazyLowerCaseFactory*(*total\_count*, *limit*, *getter*, *patype*)

Bases: *LazyFactory*

*LazyLowerCaseFactory* is a factory to handle the case insensitive data from graviti back-end.

**Parameters**

- **total\_count** (*int*) – The total count of the elements in the paging lists.
- **limit** (*int*) – The size of each lazy load page.
- **getter** (*Callable*[[*int*, *int*], *Any*]) – A callable object to get the source data.
- **patype** (*pyarrow.DataType*) – The pyarrow *DataType* of the data in the factory.

**get\_array**(*self*, *pos*, *keys*)

Get the array from the factory.

**Parameters**

- **pos** (*int*) – The page number.
- **keys** (*Tuple*[*str*, *Ellipsis*]) – The keys to access the array from factory.

**Returns** The requested pyarrow array.

**Return type** *pyarrow.Array*

**class** *graviti.paging.factory.LazyLowerCaseSubFactory*(*factory*, *keys*, *patype*)

Bases: *LazySubFactory*

*LazyLowerCaseSubFactory* is a sub-factory to handle the case insensitive data.

**Parameters**

- **factory** (*LazyFactory*) – The source *LazyFactory* instance.
- **keys** (*Tuple*[*str*, *Ellipsis*]) – The keys to access the array from the source *LazyFactory*.
- **patype** (*pyarrow.DataType*) – The pyarrow *DataType* of the data in the sub-factory.

## graviti.paging.lists

Paging list related class.

## Module Contents

### Classes

<i>PagingListBase</i>	PagingListBase is the base class of the paging list related classes.
<i>PagingList</i>	PagingList is a list composed of multiple lists (pages).
<i>MappedPagingList</i>	MappedPagingList is a list composed of multiple mapped pages.
<i>PyArrowPagingList</i>	PyArrowPagingList is a list composed of multiple pyarrow arrays (pages).

**class** graviti.paging.lists.**PagingListBase**(*iterable*)

Bases: Sequence[\_T], graviti.utility.ReprMixin

PagingListBase is the base class of the paging list related classes.

#### Parameters

- **array** – The input sequence.
- **iterable** (*Iterable[\_T]*) –

**get\_item**(*self*, *index*)

Get the element in PagingList at the given index.

**Parameters** **index** (*int*) – The input index.

**Returns** The element at the given index.

**Return type** \_T

**get\_slice**(*self*, *index*)

Get the sliced PagingList at the given slice.

#### Parameters

- **index** (*slice*) – The input slice.
- **self** (*\_PLB*) –

**Returns** The sliced PagingList at the given slice.

**Return type** \_PLB

**set\_item**(*self*, *index*, *value*)

Update the element value in PagingList at the given index.

#### Parameters

- **index** (*int*) – The element index.
- **value** (*\_T*) – The value needs to be set into the PagingList.

**Return type** None

**set\_slice**(*self*, *index*, *values*)

Update the element values at the given slice with input PagingList.

**Parameters**

- **index** (*slice*) – The element slice.
- **values** (*\_PLB*) – The PagingList which contains the elements to be set.
- **self** (*\_PLB*) –

**Raises** **ValueError** – When the input size mismatches with the slice size (when step != 1).

**Return type** None

**set\_slice\_iterable**(*self*, *index*, *values*)

Update the element values in PagingList at the given slice with iterable object.

**Parameters**

- **index** (*slice*) – The element slice.
- **values** (*Iterable[\_T]*) – The iterable object which contains the elements to be set.

**Raises** **ValueError** – When the assign input size mismatches with the slice size (when step != 1).

**Return type** None

**extend**(*self*, *values*)

Extend PagingList by appending elements from another PagingList.

**Parameters**

- **values** (*\_PLB*) – The PagingList which contains the elements to be extended.
- **self** (*\_PLB*) –

**Return type** None

**extend\_iterable**(*self*, *values*)

Extend PagingList by appending elements from the iterable.

**Parameters** **values** (*Iterable[\_T]*) – Elements to be extended into the PagingList.

**Return type** None

**extend\_nulls**(*self*, *size*)

Extend PagingList by appending nulls.

**Parameters** **size** (*int*) – The size of the nulls to be extended.

**Return type** None

**copy**(*self*)

Return a copy of the paging list.

**Returns** A copy of the paging list.

**Parameters** **self** (*\_PLB*) –

**Return type** *\_PLB*

**class** graviti.paging.lists.**PagingList**(*iterable*)

Bases: *PagingListBase[\_T]*

PagingList is a list composed of multiple lists (pages).

**Parameters** **iterable** (*Iterable[\_T]*) –

**classmethod** `from_factory(cls, factory, keys, mapper)`

Create PagingList from LazyFactory.

**Parameters**

- **factory** (`graviti.paging.factory.LazyFactory`) – The parent LazyFactory instance.
- **keys** (`Tuple[str, Ellipsis]`) – The keys to access the array from factory.
- **mapper** (`Callable[[Any], _T]`) – A callable object to convert every item in the pyarrow array.
- **cls** (`Type[_PL]`) –

**Returns** The PagingList instance created from given factory.

**Return type** `_PL`

**class** `graviti.paging.lists.MappedPagingList(iterable)`

Bases: `PagingListBase[_T]`

MappedPagingList is a list composed of multiple mapped pages.

**Parameters** `iterable (Iterable[_T])` –

**classmethod** `from_array(cls, array, mapper)`

Create MappedPagingList from the source array.

**Parameters**

- **array** (`Sequence[_T]`) – The source array of the paging list.
- **mapper** (`Callable[[Any], _T]`) – A callable object to convert every item in the pyarrow array.
- **cls** (`Type[_MPL]`) –

**Returns** The PagingList instance created from the given array.

**Return type** `_MPL`

**classmethod** `from_factory(cls, factory, keys, mapper)`

Create MappedPagingList from LazyFactory.

**Parameters**

- **factory** (`graviti.paging.factory.LazyFactory`) – The parent LazyFactory instance.
- **keys** (`Tuple[str, Ellipsis]`) – The keys to access the array from factory.
- **mapper** (`Callable[[Any], _T]`) – A callable object to convert every item in the pyarrow array.
- **cls** (`Type[_MPL]`) –

**Returns** The PagingList instance created from given factory.

**Return type** `_MPL`

**copy**(`self, copier, mapper`)

Return a copy of the paging list.

**Parameters**

- **copier** (`Callable[[_T], _T]`) – A callable object to convert loaded items in the source page to the copied page.

- **mapper** (*Callable*[[*Any*], *\_T*]) – The mapper of the new mapped page.
- **self** (*\_MPL*) –

**Returns** A copy of the paging list.

**Return type** *\_MPL*

**class** `graviti.paging.lists.PyArrowPagingList`(*iterable*)

Bases: *PagingListBase*[*\_T*]

PyArrowPagingList is a list composed of multiple pyarrow arrays (pages).

#### Parameters

- **array** – The input pyarrow array.
- **iterable** (*Iterable*[*\_T*]) –

**classmethod** `from_pyarrow`(*cls*, *array*)

Create PyArrowPagingList from pyarrow array.

#### Parameters

- **array** (*pyarrow.Array*) – The input pyarrow array.
- **cls** (*Type*[*\_PPL*]) –

**Returns** The PyArrowPagingList instance created from given pyarrow array.

**Return type** *\_PPL*

**classmethod** `from_factory`(*cls*, *factory*, *keys*, *patype*)

Create PyArrowPagingList from LazyFactory.

#### Parameters

- **factory** (`graviti.paging.factory.LazyFactory`) – The parent LazyFactory instance.
- **keys** (*Tuple*[*str*, *Ellipsis*]) – The keys to access the array from factory.
- **patype** (*pyarrow.DataType*) – The pyarrow DataType of the elements in the list.
- **cls** (*Type*[*\_PPL*]) –

**Returns** The PyArrowPagingList instance created from given factory.

**Return type** *\_PPL*

**get\_slice**(*self*, *index*)

Get the sliced PyArrowPagingList at the given slice.

#### Parameters

- **index** (*slice*) – The input slice.
- **self** (*\_PPL*) –

**Returns** The sliced PyArrowPagingList at the given slice.

**Return type** *\_PPL*

**set\_slice**(*self*, *index*, *values*)

Update the element values at the given slice with input PyArrowPagingList.

#### Parameters

- **index** (*slice*) – The element slice.

- **values** (*\_PPL*) – The PyArrowPagingList which contains the elements to be set.
- **self** (*\_PPL*) –

**Raises** **ArrowTypeError** – When two pyarrow types mismatch.

**Return type** None

**extend**(*self, values*)

Extend PyArrowPagingList by appending elements from another PyArrowPagingList.

**Parameters**

- **values** (*\_PPL*) – The PyArrowPagingList which contains the elements to be extended.
- **self** (*\_PPL*) –

**Raises** **ArrowTypeError** – When two pyarrow types mismatch.

**Return type** None

**extend\_nulls**(*self, size*)

Extend PyArrowPagingList by appending nulls.

**Parameters** **size** (*int*) – The size of the nulls to be extended.

**Return type** None

**copy**(*self*)

Return a copy of the paging list.

**Returns** A copy of the paging list.

**Parameters** **self** (*\_PPL*) –

**Return type** *\_PPL*

**to\_pyarrow**(*self*)

Convert the paging list to pyarrow ChunkedArray.

**Returns** The pyarrow ChunkedArray.

**Return type** pyarrow.ChunkedArray

## graviti.paging.offset

Paging list offset related class.

## Module Contents

### Classes

---

*Offsets*

The offsets manager of the paging list.

---

**class** graviti.paging.offset.**Offsets**(*total\_count, limit*)

The offsets manager of the paging list.

**Parameters**

- **total\_count** (*int*) – The total count of the elements in the paging list.



- **limit** (*int*) – The size of each page.

**update**(*self*, *start*, *stop*, *lengths*)

Update the offsets when setting or deleting paging list items.

**Parameters**

- **start** (*int*) – The start index.
- **stop** (*int*) – The stop index.
- **lengths** (*Iterable[int]*) – The length of the set values.

**Return type** None

**get\_coordinate**(*self*, *index*)

Get the page coordinate of the elements.

**Parameters** **index** (*int*) – The index of the element in paging list.

**Returns** The page number and the index of the page.

**Return type** Tuple[int, int]

**extend**(*self*, *lengths*)

Update the offsets when extending the paging list.

**Parameters** **lengths** (*Iterable[int]*) – The lengths of the extended pages.

**Return type** None

**copy**(*self*)

Return a copy of the Offsets.

**Returns** A copy of the Offsets.

**Parameters** **self** (*\_O*) –

**Return type** *\_O*

## graviti.paging.page

Page related class.

## Module Contents

### Classes

<i>PageBase</i>	PageBase is the base class of array wrapper and represents a page in paging list.
<i>Page</i>	Page is an array wrapper and represents a page in paging list.
<i>SlicedPage</i>	SlicedPage is an array wrapper and represents a sliced page in paging list.
<i>LazyPage</i>	LazyPage is a placeholder when the paging list page is not loaded yet.
<i>LazySlicedPage</i>	LazySlicedPage is a placeholder when the sliced paging list page is not loaded yet.

continues on next page

Table 1.53 – continued from previous page

<i>MappedPageBase</i>	MappedPageBase is the base class of the page with mapper, is used for nested DataFrame.
<i>MappedPage</i>	MappedPage is an array wrapper and represents a page in paging list.
<i>MappedSlicedPage</i>	MappedSlicedPage is an array wrapper and represents a sliced page in paging list.
<i>MappedLazyPage</i>	LazyPage with a mapper for converting every item in the source array.
<i>MappedLazySlicedPage</i>	LazySlicedPage with a mapper for converting every item in the source array.

**class** graviti.paging.page.**PageBase**

Bases: Sequence[\_T]

PageBase is the base class of array wrapper and represents a page in paging list.

**get\_item**(self, index)

Return the item at the given index.

**Parameters** **index** (int) – Position of the mutable sequence.

**Returns** The item at the given index.

**Return type** \_T

**abstract** **get\_slice**(self, start=None, stop=None, step=None)

Return a sliced page according to the given start and stop index.

**Parameters**

- **start** (Optional[int]) – The start index.
- **stop** (Optional[int]) – The stop index.
- **step** (Optional[int]) – The slice step.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** PageBase[\_T]

**abstract** **get\_array**(self)

Get the array inside the page.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** Sequence[\_T]

**class** graviti.paging.page.**Page**(array)

Bases: PageBase[\_T]

Page is an array wrapper and represents a page in paging list.

**Parameters** **array** (Sequence[\_T]) – The internal sequence of page.

**get\_slice**(self, start=None, stop=None, step=None)

Return a sliced page according to the given start and stop index.

**Parameters**

- **start** (Optional[int]) – The start index.
- **stop** (Optional[int]) – The stop index.
- **step** (Optional[int]) – The slice step.

**Returns** A sliced page according to the given start and stop index.

**Return type** *SlicedPage*[\_T]

**get\_array**(*self*)

Get the array inside the page.

**Returns** The array inside the page.

**Return type** *Sequence*[\_T]

**class** *graviti.paging.page.SlicedPage*(*ranging*, *source\_array*)

Bases: *PageBase*[\_T]

*SlicedPage* is an array wrapper and represents a sliced page in paging list.

#### Parameters

- **ranging** (*range*) – The range instance of this page.
- **array** – The internal sequence of page.
- **source\_array** (*Sequence*[\_T]) –

**get\_slice**(*self*, *start=None*, *stop=None*, *step=None*)

Return a sliced page according to the given start and stop index.

#### Parameters

- **start** (*Optional*[*int*]) – The start index.
- **stop** (*Optional*[*int*]) – The stop index.
- **step** (*Optional*[*int*]) – The slice step.

**Returns** A sliced page according to the given start and stop index.

**Return type** *SlicedPage*[\_T]

**get\_array**(*self*)

Get the array inside the page.

**Returns** The array inside the page.

**Return type** *Sequence*[\_T]

**class** *graviti.paging.page.LazyPage*(*length*, *array\_getter*)

Bases: *PageBase*[\_T]

*LazyPage* is a placeholder when the paging list page is not loaded yet.

#### Parameters

- **length** (*int*) – The length of this page.
- **array\_getter** (*Callable*[[*int*], *Sequence*[\_T]]) – A callable object to get the source array.

**get\_slice**(*self*, *start=None*, *stop=None*, *step=None*)

Return a lazy sliced page according to the given start and stop index.

#### Parameters

- **start** (*Optional*[*int*]) – The start index.
- **stop** (*Optional*[*int*]) – The stop index.
- **step** (*Optional*[*int*]) – The slice step.

**Returns** A sliced page according to the given start and stop index.

**Return type** *LazySlicedPage*[\_T]

**get\_array**(*self*)

Get the array inside the page.

**Returns** The array inside the page.

**Return type** *Sequence*[\_T]

**class** graviti.paging.page.**LazySlicedPage**(*ranging*, *array\_getter*)

Bases: *PageBase*[\_T]

LazySlicedPage is a placeholder when the sliced paging list page is not loaded yet.

#### Parameters

- **ranging** (*range*) – The range instance of this page.
- **array\_getter** (*Callable*[[*Any*], *Sequence*[\_T]]) – A callable object to get the source array.

**get\_slice**(*self*, *start=None*, *stop=None*, *step=None*)

Return a lazy sliced page according to the given start and stop index.

#### Parameters

- **start** (*Optional*[*int*]) – The start index.
- **stop** (*Optional*[*int*]) – The stop index.
- **step** (*Optional*[*int*]) – The slice step.

**Returns** A sliced page according to the given start and stop index.

**Return type** *LazySlicedPage*[\_T]

**get\_array**(*self*)

Get the array inside the page.

**Returns** The array inside the page.

**Return type** *Sequence*[\_T]

**class** graviti.paging.page.**MappedPageBase**

Bases: *PageBase*[\_T]

MappedPageBase is the base class of the page with mapper, is used for nested DataFrame.

**abstract copy**(*self*, *copier*, *mapper*)

Return a copy of the mapped page.

#### Parameters

- **copier** (*Callable*[[*Any*], *Any*]) – A callable object to convert loaded items in the source page to the copied page.
- **mapper** (*Callable*[[*Any*], *Any*]) – The mapper of the new mapped page.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** *MappedPageBase*[\_T]

**class** graviti.paging.page.**MappedPage**(*array*)

Bases: *MappedPageBase*[\_T], *Page*[\_T]

MappedPage is an array wrapper and represents a page in paging list.

**Parameters** `array (Sequence[_T])` –

**get\_slice**(*self*, *start=None*, *stop=None*, *step=None*)

Return a sliced page according to the given start and stop index.

**Parameters**

- **start** (*Optional[int]*) – The start index.
- **stop** (*Optional[int]*) – The stop index.
- **step** (*Optional[int]*) – The slice step.

**Returns** A sliced page according to the given start and stop index.

**Return type** `MappedSlicedPage[_T]`

**copy**(*self*, *copier*, *mapper*)

Return a copy of the mapped page.

**Parameters**

- **copier** (*Callable[[\_T], \_T]*) – A callable object to convert loaded items in the source page to the copied page.
- **mapper** (*Callable[[Any], Any]*) – The mapper of the new mapped page.

**Returns** A copy of the mapped page.

**Return type** `MappedPage[_T]`

**class** `graviti.paging.page.MappedSlicedPage(ranging, source_array)`

Bases: `MappedPageBase[_T]`, `SlicedPage[_T]`

MappedSlicedPage is an array wrapper and represents a sliced page in paging list.

**Parameters**

- **ranging** (*range*) –
- **source\_array** (*Sequence[\_T]*) –

**get\_slice**(*self*, *start=None*, *stop=None*, *step=None*)

Return a sliced page according to the given start and stop index.

**Parameters**

- **start** (*Optional[int]*) – The start index.
- **stop** (*Optional[int]*) – The stop index.
- **step** (*Optional[int]*) – The slice step.

**Returns** A sliced page according to the given start and stop index.

**Return type** `MappedSlicedPage[_T]`

**copy**(*self*, *copier*, *mapper*)

Return a copy of the mapped page.

**Parameters**

- **copier** (*Callable[[Any], Any]*) – A callable object to convert loaded items in the source page to the copied page.
- **mapper** (*Callable[[Any], Any]*) – The mapper of the new mapped page.

**Returns** A copy of the mapped page.

**Return type** *MappedPage*[\_T]

**class** graviti.paging.page.**MappedLazyPage**(length, array\_getter, mapper)

Bases: *MappedPageBase*[\_T]

LazyPage with a mapper for converting every item in the source array.

**Parameters**

- **length** (*int*) – The length of this page.
- **array\_getter** (*Callable*[[], *Sequence*[\_T]]) – A callable object to get the source array.
- **mapper** (*Callable*[[*Any*], *Any*]) – A callable object to convert every item in the source array.

**get\_slice**(self, start=None, stop=None, step=None)

Return a lazy sliced page according to the given start and stop index.

**Parameters**

- **start** (*Optional*[*int*]) – The start index.
- **stop** (*Optional*[*int*]) – The stop index.
- **step** (*Optional*[*int*]) – The slice step.

**Returns** A sliced page according to the given start and stop index.

**Return type** Union[*MappedLazySlicedPage*[\_T], *MappedSlicedPage*[\_T]]

**get\_array**(self)

Get the array inside the page.

**Returns** The array inside the page.

**Return type** *Sequence*[\_T]

**copy**(self, copier, mapper)

Return a copy of the mapped page.

**Parameters**

- **copier** (*Callable*[[*Any*], *Any*]) – A callable object to convert loaded items in the source page to the copied page.
- **mapper** (*Callable*[[*Any*], *Any*]) – The mapper of the new mapped page.

**Returns** A copy of the mapped page.

**Return type** Union[*MappedPage*[\_T], *MappedLazyPage*[\_T]]

**class** graviti.paging.page.**MappedLazySlicedPage**(ranging, array\_getter, mapper)

Bases: *MappedPageBase*[\_T]

LazySlicedPage with a mapper for converting every item in the source array.

**Parameters**

- **ranging** (*range*) – The range instance of this page.
- **array\_getter** (*Callable*[[], *Sequence*[\_T]]) – A callable object to get the source array.
- **mapper** (*Callable*[[*Any*], *Any*]) – A callable object to convert every item in the source array.

**get\_slice**(*self*, *start=None*, *stop=None*, *step=None*)

Return a lazy sliced page according to the given start and stop index.

**Parameters**

- **start** (*Optional[int]*) – The start index.
- **stop** (*Optional[int]*) – The stop index.
- **step** (*Optional[int]*) – The slice step.

**Returns** A sliced page according to the given start and stop index.

**Return type** Union[*MappedLazySlicedPage*[\_T], *MappedSlicedPage*[\_T]]

**get\_array**(*self*)

Get the array inside the page.

**Returns** The array inside the page.

**Return type** Sequence[\_T]

**copy**(*self*, *copier*, *mapper*)

Return a copy of the mapped page.

**Parameters**

- **copier** (*Callable[[Any], Any]*) – A callable object to convert loaded items in the source page to the copied page.
- **mapper** (*Callable[[Any], Any]*) – The mapper of the new mapped page.

**Returns** A copy of the mapped page.

**Return type** Union[*MappedPage*[\_T], *MappedLazySlicedPage*[\_T]]

## graviti.paging.wrapper

PyArrow array wrapper related class.

## Module Contents

### Classes

<i>WrapperRegister</i>	The class decorator to connect pyarrow type and the pyarrow array wrapper.
<i>ScalarWrapper</i>	The wrapper of pyarrow scalar.
<i>ArrayWrapper</i>	The wrapper of pyarrow array.
<i>StructScalarWrapper</i>	The wrapper of pyarrow StructScalar to make it case insensitive.
<i>StructArrayWrapper</i>	The wrapper of pyarrow StructArray to make it case insensitive.
<i>ListScalarWrapper</i>	The wrapper of pyarrow ListScalar to make it case insensitive.
<i>ListArrayWrapper</i>	The wrapper of pyarrow ListArray to make it case insensitive.

**class** graviti.paging.wrapper.**WrapperRegister**(pyarrow\_type\_id)

The class decorator to connect pyarrow type and the pyarrow array wrapper.

**Parameters** **pyarrow\_type\_id** (*int*) – The PyArrow type id.

**classmethod** **get**(cls, pyarrow\_type\_id)

Get the corresponding registered pyarrow array wrapper.

**Parameters** **pyarrow\_type\_id** (*int*) – The PyArrow type id.

**Returns** The corresponding registered pyarrow array wrapper.

**Return type** Type[[ArrayWrapper](#)]

**class** graviti.paging.wrapper.**ScalarWrapper**(scalar)

The wrapper of pyarrow scalar.

**Parameters** **scalar** (*pyarrow.scalar*) – The PyArrow scalar needs to be wrapped.

**property** **is\_valid**(self)

The wrapper of pyarrow Scalar.is\_valid method.

**Returns** Bool value indicating whether this scalar is None.

**Return type** bool

**as\_py**(self)

The wrapper of pyarrow Scalar.as\_py method.

**Returns** Return this value as a Python builtin object.

**Return type** *Any*

**class** graviti.paging.wrapper.**ArrayWrapper**(array)

The wrapper of pyarrow array.

**Parameters** **array** (*pyarrow.Array*) – The PyArrow array needs to be wrapped.

**class** graviti.paging.wrapper.**StructScalarWrapper**(scalar)

Bases: [ScalarWrapper](#)

The wrapper of pyarrow StructScalar to make it case insensitive.

**Parameters** **scalar** (*pyarrow.StructScalar*) – The PyArrow StructScalar needs to be wrapped.

**classmethod** **from\_wrapper**(cls, scalar, wrappers)

Create StructScalarWrapper instance by inputting scalar and wrappers.

**Parameters**

- **scalar** (*pyarrow.ListScalar*) – The PyArrow StructScalar needs to be wrapped.
- **wrappers** (*Dict[str, Type[ArrayWrapper]]*) – The wrappers of the input scalar.
- **cls** (*Type[\_S]*) –

**Returns** The StructScalarWrapper instance created by the input scalar and wrapper.

**Return type** *\_S*

**class** graviti.paging.wrapper.**StructArrayWrapper**(array)

Bases: [ArrayWrapper](#)

The wrapper of pyarrow StructArray to make it case insensitive.

**Parameters** **array** (*pyarrow.StringArray*) – The PyArrow StructArray instance needs to be wrapped.



**field**(*self*, *key*)

The wrapper of pyarrow StructArray.field method.

**Parameters** **key** (*str*) – The name of the field.

**Returns** The child array belonging to the field.

**Return type** pyarrow.Array

**class** graviti.paging.wrapper.**ListScalarWrapper**(*scalar*)

Bases: [ScalarWrapper](#)

The wrapper of pyarrow ListScalar to make it case insensitive.

**Parameters** **scalar** (*pyarrow.ListScalar*) – The PyArrow ListScalar instance needs to be wrapped.

**classmethod** **from\_wrapper**(*cls*, *scalar*, *wrapper*)

Create ListScalarWrapper instance by inputting scalar and wrapper.

**Parameters**

- **scalar** (*pyarrow.ListScalar*) – The PyArrow ListScalar instance needs to be wrapped.
- **wrapper** (*Type[ArrayWrapper]*) – The wrapper of the input scalar.
- **cls** (*Type[\_LS]*) –

**Returns** The ListScalarWrapper instance created by the input scalar and wrapper.

**Return type** *\_LS*

**property** **values**(*self*)

The wrapper of pyarrow ListScalar.values attr.

**Returns** The internal values of the pyarrow scalar.

**Return type** [ArrayWrapper](#)

**class** graviti.paging.wrapper.**ListArrayWrapper**(*array*)

Bases: [ArrayWrapper](#)

The wrapper of pyarrow ListArray to make it case insensitive.

**Parameters** **array** (*pyarrow.ListArray*) – The PyArrow ListArray instance needs to be wrapped.

**classmethod** **from\_wrapper**(*cls*, *array*, *wrapper*)

Create ListArrayWrapper instance by inputting array and wrapper.

**Parameters**

- **array** (*pyarrow.ListArray*) – The PyArrow ListArray instance needs to be wrapped.
- **wrapper** (*Type[ArrayWrapper]*) – The wrapper of the input array.
- **cls** (*Type[\_LA]*) –

**Returns** The ListArrayWrapper instance created by the input array and wrapper.

**Return type** *\_LA*

### `graviti.portex`

Schema module.

### Submodules

#### `graviti.portex.avro`

Code converting PyArrow schema to Avro Schema.

### Module Contents

#### Classes

---

*AvroSchema*

---

*AvroField*

---

*AvroPrimitiveSchema*

---

*AvroRecordSchema*

---

*AvroArraySchema*

---

*AvroEnumSchema*

---

#### Functions

---

*convert\_portex\_schema\_to\_avro(\_schema)*

---

**class** `graviti.portex.avro.AvroSchema`

`to_json(self)`

**class** `graviti.portex.avro.AvroField(name_registry, typ, name, optional=True, has_default=False, default=None)`

    Parameters `typ` (`AvroSchema`) –

`to_json(self)`

**class** `graviti.portex.avro.AvroPrimitiveSchema(typ, has_default=False, default=None)`

    Bases: `AvroSchema`

`to_json(self)`

**class** `graviti.portex.avro.AvroRecordSchema(name_registry, name, namespace, fields, aliases=None)`

    Bases: `AvroSchema`

```

    Parameters fields ([]) –
    to_json(self)
class graviti.portex.avro.AvroArraySchema(items)
    Bases: AvroSchema
    Parameters items (AvroSchema) –
    to_json(self)
class graviti.portex.avro.AvroEnumSchema(values)
    Bases: AvroSchema
    to_json(self)
graviti.portex.avro.convert_portex_schema_to_avro(_schema)

    Parameters _schema (graviti.portex.builtin.record) –

```

`graviti.portex.base`

The base elements of Portex type.

## Module Contents

### Classes

<code>PortexType</code>	The base class of portex type.
<code>PortexRecordBase</code>	The base class of record like Portex types.

### Functions

<code>read_yaml(path)</code>	Read a yaml file into Portex type.
<code>read_json(path)</code>	Read a json file into Portex type.

### Attributes

<code>PYARROW_TYPE_ID_TO_PORTEX_TYPE</code>
---

`graviti.portex.base.PYARROW_TYPE_ID_TO_PORTEX_TYPE`

**class** `graviti.portex.base.PortexType`

The base class of portex type.

**property** `imports(self)`

Get the PortexType imports.

**Returns** The Imports instance of this PortexType.

**Return type** *graviti.portex.package.Imports*

**classmethod** `from_pyobj(cls, content, _imports=None)`

Create Portex type instance from python dict.

**Parameters**

- **content** (*Dict[str, Any]*) – A python dict representing a Portex type.
- **cls** (*Type[\_T]*) –
- **\_imports** (*Optional[graviti.portex.package.Imports]*) –

**Returns** A Portex type instance created from the input python dict.

**Return type** *\_T*

**classmethod** `from_pyarrow(cls, pyarrow_type)`

Create Portex type instance from PyArrow type.

**Parameters**

- **pyarrow\_type** (*pyarrow.DataType*) – The PyArrow type.
- **cls** (*Type[\_T]*) –

**Raises** **TypeError** – When the PyArrow type is not supported.

**Returns** The created Portex type instance.

**Return type** *\_T*

**classmethod** `from_json(cls, content)`

Create Portex type instance from JSON string.

**Parameters**

- **content** (*str*) – A JSON string representing a Portex type.
- **cls** (*Type[\_T]*) –

**Returns** A Portex type instance created from the input JSON string.

**Return type** *\_T*

**classmethod** `from_yaml(cls, content)`

Create Portex type instance from YAML string.

**Parameters**

- **content** (*str*) – A YAML string representing a Portex type.
- **cls** (*Type[\_T]*) –

**Returns** A Portex type instance created from the input YAML string.

**Return type** *\_T*

**to\_pyobj** (*self, \_with\_imports=True*)

Dump the instance to a python dict.

**Returns** A python dict representation of the Portex type.

**Parameters** **\_with\_imports** (*bool*) –

**Return type** *Dict[str, Any]*

**to\_json** (*self*)

Dump the instance to a JSON string.

**Returns** A JSON representation of the Portex type.

**Return type** str

**to\_yaml**(*self*)

Dump the instance to a YAML string.

**Returns** A YAML representation of the Portex type.

**Return type** str

**abstract to\_pyarrow**(*self*)

Convert the Portex type to the corresponding builtin PyArrow DataType.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Returns** The corresponding builtin PyArrow DataType.

**Return type** pyarrow.DataType

**abstract to\_builtin**(*self*)

Expand the top level type to Portex builtin type.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** *graviti.portex.builtin.PortexBuiltinType*

**copy**(*self*)

Get a copy of the portex type.

**Returns** A copy of the portex type.

**Parameters** **self** (*\_T*) –

**Return type** *\_T*

**class** *graviti.portex.base.PortexRecordBase*

Bases: *PortexType*, *graviti.utility.UserMutableMapping*[str, *PortexType*]

The base class of record like Portex types.

**insert**(*self, index, name, portex\_type*)

Insert the name and portex\_type at the index.

**Parameters**

- **index** (*int*) – The index to insert the field.
- **name** (*str*) – The name of the field to be inserted.
- **portex\_type** (*PortexType*) – The portex\_type of the field to be inserted.

**Return type** None

**astype**(*self, name, portex\_type*)

Convert the type of the field with the given name to the new PortexType.

**Parameters**

- **name** (*str*) – The name of the field to convert.
- **portex\_type** (*PortexType*) – The new PortexType of the field to convert to.

**Return type** None

**rename**(*self, old\_name, new\_name*)

Rename the name of a field.

**Parameters**

- **old\_name** (*str*) – The current name of the field to be renamed.
- **new\_name** (*str*) – The new name of the field to assign.

**Return type** None

**to\_pyarrow**(*self*)

Convert the Portex type to the corresponding builtin PyArrow StructType.

**Returns** The corresponding builtin PyArrow StructType.

**Return type** pyarrow.StructType

`graviti.portex.base.read_yaml(path)`

Read a yaml file into Portex type.

**Parameters** **path** (*graviti.utility.PathLike*) – The path of the yaml file.

**Returns** A Portex type instance created from the input yaml file.

**Return type** *PortexType*

`graviti.portex.base.read_json(path)`

Read a json file into Portex type.

**Parameters** **path** (*graviti.utility.PathLike*) – The path of the json file.

**Returns** A Portex type instance created from the input json file.

**Return type** *PortexType*

`graviti.portex.builder`

Portex type builder related classes.

## Module Contents

### Classes

<i>PackageRepo</i>	The local git repo of the external Portex package.
<i>PackageBuilder</i>	The builder of the external Portex package.
<i>TypeBuilder</i>	The builder of the external Portex template type.
<i>BuilderImports</i>	The imports of the Portex template type.

### Functions

<i>build_package</i> (url, revision)	Build an external package.
--------------------------------------	----------------------------

## Attributes

---

[\*EXTERNAL\\_TYPE\\_TO\\_CONTAINER\*](#)

---

`graviti.portex.builder.EXTERNAL_TYPE_TO_CONTAINER`

**class** `graviti.portex.builder.PackageRepo(url, revision)`

The local git repo of the external Portex package.

### Parameters

- **url** (*str*) – The git repo url of the external package.
- **revision** (*str*) – The git repo revision (tag/commit) of the external package.

**get\_root**(*self*)

Get the root directory path of the package repo.

**Returns** The root directory path of the package repo.

**Raises** **TypeError** – when the “ROOT.yaml” not found or more than one “ROOT.yaml” found.

**Return type** `pathlib.Path`

**class** `graviti.portex.builder.PackageBuilder(url, revision)`

The builder of the external Portex package.

### Parameters

- **url** (*str*) – The git repo url of the external package.
- **revision** (*str*) – The git repo revision (tag/commit) of the external package.

**build**(*self*)

Build the Portex external package.

**Returns** The builded Portex external package.

**Return type** `graviti.portex.package.ExternalPackage`

**class** `graviti.portex.builder.TypeBuilder(name, path, builder)`

The builder of the external Portex template type.

### Parameters

- **name** (*str*) – The name of the Portex template type.
- **path** (`pathlib.Path`) – The source file path of the Portex template type.
- **package** – The package the Portex template type belongs to.
- **builder** (`PackageBuilder`) –

**build**(*self*)

Build the Portex external type.

**Returns** The builded Portex external type.

**Raises** **TypeError** – Raise when circular reference detected.

**Return type** `Type[graviti.portex.external.PortexExternalType]`

**class** `graviti.portex.builder.BuilderImports`

Bases: `graviti.portex.package.Imports`

The imports of the Portex template type.

**Parameters** `package` – The package the portex belongs to.

**classmethod** `from_pyobj(cls, content, builder)`

Create Imports instance from python list.

**Parameters**

- **content** (`List[Dict[str, Any]]`) – A python list representing imported types.
- **builder** (`PackageBuilder`) – The package builder.
- **cls** (`Type[_I]`) –

**Returns** A Imports instance created from the input python list.

**Return type** `_I`

`graviti.portex.builder.build_package(url, revision)`

Build an external package.

**Parameters**

- **url** (`str`) – The git repo url of the external package.
- **revision** (`str`) – The git repo revision (tag/commit) of the external package.

**Returns** The ExternalPackage instance.

**Return type** `graviti.portex.package.ExternalPackage`

`graviti.portex.builtin`

The Portex builtin types.

## Module Contents

### Classes

<code>PortexBuiltinType</code>	The base class of Portex builtin type.
<code>string</code>	Portex primitive type <code>string</code> .
<code>binary</code>	Portex primitive type <code>binary</code> .
<code>boolean</code>	Portex primitive type <code>boolean</code> .
<code>int32</code>	Portex primitive type <code>int32</code> .
<code>int64</code>	Portex primitive type <code>int64</code> .
<code>float32</code>	Portex primitive type <code>float32</code> .
<code>float64</code>	Portex primitive type <code>float64</code> .
<code>record</code>	Portex complex type <code>record</code> .
<code>enum</code>	Portex complex type <code>enum</code> .
<code>array</code>	Portex complex type <code>array</code> .



## Attributes

---

### *builtins*

---

graviti.portex.builtin.**builtins**

**class** graviti.portex.builtin.**PortexBuiltinType**(*nullable=False*)

Bases: [graviti.portex.base.PortexType](#)

The base class of Portex builtin type.

**Parameters** **nullable** (*bool*) –

**to\_builtin**(*self*)

Expand the top level type to Portex builtin type.

**Returns** The expanded Portex builtin type.

**Parameters** **self** (*\_T*) –

**Return type** *\_T*

**class** graviti.portex.builtin.**string**(*nullable=False*)

Bases: [PortexBuiltinType](#)

Portex primitive type string.

**Parameters** **nullable** (*bool*) – Whether it is a nullable type.

## Examples

```
>>> t = string()
>>> t
string()
```

**to\_pyarrow**(*self*)

Convert the Portex type to the corresponding builtin PyArrow DataType.

**Returns** The corresponding builtin PyArrow DataType.

**Return type** pyarrow.DataType

**class** graviti.portex.builtin.**binary**(*nullable=False*)

Bases: [PortexBuiltinType](#)

Portex primitive type binary.

**Parameters** **nullable** (*bool*) – Whether it is a nullable type.

### Examples

```
>>> t = binary()
>>> t
binary()
```

**to\_pyarrow(self)**

Convert the Portex type to the corresponding builtin PyArrow DataType.

**Returns** The corresponding builtin PyArrow DataType.

**Return type** pyarrow.DataType

**class** graviti.portex.builtin.**boolean**(nullable=False)

Bases: [PortexBuiltinType](#)

Portex primitive type boolean.

**Parameters** **nullable** (*bool*) – Whether it is a nullable type.

### Examples

```
>>> t = boolean()
>>> t
boolean()
```

**to\_pyarrow(self)**

Convert the Portex type to the corresponding builtin PyArrow DataType.

**Returns** The corresponding builtin PyArrow DataType.

**Return type** pyarrow.DataType

**class** graviti.portex.builtin.**int32**(nullable=False)

Bases: [PortexBuiltinType](#)

Portex primitive type int32.

**Parameters** **nullable** (*bool*) – Whether it is a nullable type.

### Examples

```
>>> t = int32()
>>> t
int32()
```

**to\_pyarrow(self)**

Convert the Portex type to the corresponding builtin PyArrow DataType.

**Returns** The corresponding builtin PyArrow DataType.

**Return type** pyarrow.DataType

**class** graviti.portex.builtin.**int64**(nullable=False)

Bases: [PortexBuiltinType](#)

Portex primitive type int64.

**Parameters** **nullable** (*bool*) – Whether it is a nullable type.

## Examples

```
>>> t = int64()
>>> t
int64()
```

**to\_pyarrow**(*self*)

Convert the Portex type to the corresponding builtin PyArrow DataType.

**Returns** The corresponding builtin PyArrow DataType.

**Return type** pyarrow.DataType

**class** graviti.portex.builtin.**float32**(*nullable=False*)

Bases: [PortexBuiltinType](#)

Portex primitive type float32.

**Parameters** **nullable** (*bool*) – Whether it is a nullable type.

## Examples

```
>>> t = float32()
>>> t
float32()
```

**to\_pyarrow**(*self*)

Convert the Portex type to the corresponding builtin PyArrow DataType.

**Returns** The corresponding builtin PyArrow DataType.

**Return type** pyarrow.DataType

**class** graviti.portex.builtin.**float64**(*nullable=False*)

Bases: [PortexBuiltinType](#)

Portex primitive type float64.

**Parameters** **nullable** (*bool*) – Whether it is a nullable type.

## Examples

```
>>> t = float64()
>>> t
float64()
```

**to\_pyarrow**(*self*)

Convert the Portex type to the corresponding builtin PyArrow DataType.

**Returns** The corresponding builtin PyArrow DataType.

**Return type** pyarrow.DataType

**class** graviti.portex.builtin.**record**(*fields, nullable=False*)

Bases: [PortexBuiltinType](#), [graviti.portex.base.PortexRecordBase](#)

Portex complex type record.

**Parameters**

- **fields** (*Union[Iterable[Tuple[str, graviti.portex.base.PortexType]], Mapping[str, graviti.portex.base.PortexType]]*) – The fields of the record.
- **nullable** (*bool*) – Whether it is a nullable type.

## Examples

Create a record by dict:

```
>>> t = record({"f0": int32(), "f1": float32()})
>>> t
record(
  fields={
    'f0': int32(),
    'f1': float32(),
  },
)
```

Create a record by tuple list:

```
>>> t = record([("f0", string()), ("f1", enum(["v0", "v1"]))]
>>> t
record(
  fields={
    'f0': string(),
    'f1': enum(
      values=['v0', 'v1'],
    ),
  },
)
```

**to\_pyarrow**(*self*)

Convert the Portex type to the corresponding builtin PyArrow DataType.

**Returns** The corresponding builtin PyArrow struct DataType.

**Return type** pyarrow.DataType

**class** graviti.portex.builtin.**enum**(*values, nullable=False*)

Bases: *PortexBuiltinType*

Portex complex type enum.

**Parameters**

- **values** (*Iterable[\_E]*) – The values of the enum members.
- **nullable** (*bool*) – Whether it is a nullable type.

## Examples

```
>>> t = enum(["v0", "v1"])
>>> t
enum(
  values=['v0', 'v1'],
)
```

**to\_pyarrow**(*self*)

Convert the Portex type to the corresponding builtin PyArrow DataType.

**Returns** The corresponding builtin PyArrow DataType.

**Return type** pyarrow.DataType

**class** graviti.portex.builtin.array(*items, length=None, nullable=False*)

Bases: [PortexBuiltinType](#)

Portex complex type array.

**Parameters**

- **items** ([graviti.portex.base.PortexType](#)) – The item type of the array.
- **length** (*Optional[int]*) – The length of the array.
- **nullable** (*bool*) – Whether it is a nullable type.

## Examples

```
>>> t = array(int32(0), 100)
>>> t
array(
  items=int32(
    minimum=0,
  ),
  length=100,
)
```

**to\_pyarrow**(*self*)

Convert the Portex type to the corresponding builtin PyArrow DataType.

**Returns** The corresponding builtin PyArrow DataType.

**Return type** pyarrow.DataType

**graviti.portex.enum**

Portex enum values related classes.

## Module Contents

### Classes

<a href="#"><i>EnumValues</i></a>	The base class of portex enum values.
<a href="#"><i>EnumValueList</i></a>	The portex enum values in list format.
<a href="#"><i>EnumValueDict</i></a>	The portex enum values in dict format.

### Functions

<a href="#"><i>create_enum_values</i></a> (values)	The factory function of EnumValues.
--	-------------------------------------

### Attributes

<a href="#"><i>EnumValueType</i></a>
--------------------------------------

graviti.portex.enum.**EnumValueType**

**class** graviti.portex.enum.**EnumValues**

The base class of portex enum values.

**abstract** **to\_pyobj**(self)

Dump the instance to a python list or dict.

**Raises** **NotImplementedError** – The method of the base class should not be called.

**Return type** Union[List[EnumValueType], Dict[int, EnumValueType]]

**class** graviti.portex.enum.**EnumValueList**(values)

Bases: [\*EnumValues\*](#), graviti.utility.UserSequence[[\*EnumValueType\*](#)]

The portex enum values in list format.

**Parameters** **values** (Iterable[EnumValueType]) – The enum values.

**to\_pyobj**(self)

Dump the instance to a python list.

**Returns** A python list representation of the enum values.

**Return type** List[EnumValueType]

**class** graviti.portex.enum.**EnumValueDict**(values)

Bases: [\*EnumValues\*](#), graviti.utility.UserMapping[int, [\*EnumValueType\*](#)]

The portex enum values in dict format.

**Parameters** **values** (Mapping[int, EnumValueType]) – The enum values.

**to\_pyobj**(self)

Dump the instance to a python dict.

**Returns** A python dict representation of the enum values.

**Return type** Dict[int, EnumValueType]

`graviti.portex.enum.create_enum_values(values)`

The factory function of EnumValues.

**Parameters** `values` (`Union[Sequence[EnumValueType], Mapping[int, EnumValueType]]`) – The enum values.

**Returns** The EnumValues instance created by the input enum values.

**Raises** **TypeError** – When the input enum values is not in list or dict format.

**Return type** `EnumValues`

`graviti.portex.external`

Portex external base class.

## Module Contents

### Classes

<code>PortexExternalType</code>	The base class of Portex external type.
---------------------------------	---

### Attributes

<code>EXTERNAL_TYPE_TO_CONTAINER</code>
---

<code>EXTERNAL_TYPE_TO_ELEMENT</code>
---------------------------------------

`graviti.portex.external.EXTERNAL_TYPE_TO_CONTAINER`

`graviti.portex.external.EXTERNAL_TYPE_TO_ELEMENT`

**class** `graviti.portex.external.PortexExternalType(*args, **kwargs)`

Bases: `graviti.portex.base.PortexType`

The base class of Portex external type.

#### Parameters

- `args` (`Any`) –
- `kwargs` (`Any`) –

**property** `internal_type(self)`

Get the internal type of the PortexExternalType.

**Returns** The internal type of the PortexExternalType.

**Return type** `graviti.portex.base.PortexType`

**to\_pyarrow(self)**

Convert the Portex type to the corresponding builtin PyArrow DataType.

**Returns** The corresponding builtin PyArrow DataType.

**Return type** `pyarrow.DataType`

**to\_builtin**(*self*)

Expand the top level of the Portex external type to Portex builtin type.

**Returns** The expanded Portex builtin type.

**Return type** `graviti.portex.builtin.PortexBuiltinType`

## `graviti.portex.factory`

Template factory related classes.

## Module Contents

### Classes

<i>Factory</i>	The base class of the template factory.
<i>FrozenFieldsFactory</i>	The factory for FrozenFields.
<i>FrozenFieldsFactoryWrapper</i>	The factory for FrozenFields which needs kwargs transformed.
<i>ConnectedFieldsFactory</i>	The factory for ConnectedFields.
<i>TypeFactory</i>	The template factory for portex type.
<i>ConstantFactory</i>	The template factory for a constant.
<i>VariableFactory</i>	The template factory for a variable.
<i>ListFactory</i>	The template factory for a list.
<i>DictFactory</i>	The template factory for a dict.
<i>FieldFactory</i>	The template factory for a tuple of name and PortexType.
<i>FieldsFactory</i>	The template factory for a Fields.

### Functions

<i>mapping_unpack_factory_creator</i> (decl, ptype)	Check the object unpack grammar and returns the corresponding factory.
<i>type_factory_creator</i> (decl, imports)	Check the input and returns the corresponding type factory.
<i>string_factory_creator</i> (decl, ptype = PTYPE.Any)	Check whether the input string is variable and returns the corresponding factory.
<i>factory_creator</i> (decl, imports, ptype = PTYPE.Any)	Check input type and returns the corresponding factory.



## Attributes

---

### *UnionFieldsFactory*

---

**class** graviti.portex.factory.**Factory**

The base class of the template factory.

**class** graviti.portex.factory.**FrozenFieldsFactory**(*decl, imports*)

Bases: *Factory*

The factory for FrozenFields.

#### Parameters

- **decl** (*Iterable[Dict[str, Any]]*) – The decalaration of frozen fields.
- **imports** (*graviti.portex.package.Imports*) – The Imports instance to specify the import scope of the fields.

**class** graviti.portex.factory.**FrozenFieldsFactoryWrapper**(*factory, kwargs\_transformer*)

Bases: *Factory*

The factory for FrozenFields which needs kwargs transformed.

#### Parameters

- **factory** (*Union[FrozenFieldsFactory, FrozenFieldsFactoryWrapper]*) – The factory of frozen fields.
- **kwargs\_transformer** (*Callable[Ellipsis, Dict[str, Any]]*) – The method to transform the kwargs to the kwargs of base type.

graviti.portex.factory.**UnionFieldsFactory**

**class** graviti.portex.factory.**ConnectedFieldsFactory**(*decl, class\_, imports, kwargs\_transformer*)

The factory for ConnectedFields.

#### Parameters

- **decl** (*Dict[str, Any]*) – A dict which indicates a portex type.
- **class** – The base type.
- **imports** (*graviti.portex.package.Imports*) – The Imports instance to specify the import scope of the template.
- **kwargs\_transformer** (*Callable[Ellipsis, Dict[str, Any]]*) – The method to transform the kwargs to the kwargs of base type.
- **class\_** (*Type[graviti.portex.base.PortexRecordBase]*) –

**classmethod** **from\_parameter\_name**(*cls, name*)

Create ConnectedFieldsFactory for Fields with the given parameter name.

#### Parameters

- **name** (*str*) – The parameter name of the input fields.
- **cls** (*Type[\_CFF]*) –

**Returns** The created ConnectedFieldsFactory.

**Return type** *\_CFF*

**class** graviti.portex.factory.**TypeFactory**(*decl, imports*)

Bases: *Factory*

The template factory for portex type.

**Parameters**

- **decl** (*Dict[str, Any]*) – A dict which indicates a portex type.
- **imports** (*graviti.portex.package.Imports*) –

**transform\_kwargs**(*self, kwargs*)

Transform the keyword arguments to what the base type needs.

**Parameters** **kwargs** (*Dict[str, Any]*) – The input arguments.

**Returns** The transformed keyword arguments.

**Return type** *Dict[str, Any]*

**class** graviti.portex.factory.**ConstantFactory**(*decl*)

Bases: *Factory*, *Generic[\_C]*

The template factory for a constant.

**Parameters** **decl** (*\_C*) – The constant to be created by the factory.

**class** graviti.portex.factory.**VariableFactory**(*decl, ptype=PTYPE.Any, is\_unpack=False*)

Bases: *Factory*

The template factory for a variable.

**Parameters**

- **decl** (*str*) – The parameter name of the variable.
- **ptype** (*graviti.portex.ptype.PType*) – The parameter type.
- **is\_unpack** (*bool*) –

**class** graviti.portex.factory.**ListFactory**(*decl, ptype=PTYPE.Any*)

Bases: *Factory*

The template factory for a list.

**Parameters**

- **decl** (*List[Any]*) – A list template.
- **ptype** (*graviti.portex.ptype.PType*) – The parameter type of the list.

**class** graviti.portex.factory.**DictFactory**(*decl, ptype=PTYPE.Any*)

Bases: *Factory*

The template factory for a dict.

**Parameters**

- **decl** (*Dict[str, Any]*) – A dict template.
- **ptype** (*graviti.portex.ptype.PType*) – The parameter type of the dict.

**class** graviti.portex.factory.**FieldFactory**(*decl, imports*)

Bases: *Factory*

The template factory for a tuple of name and PortexType.

**Parameters**

- **decl** (*Dict[str, Any]*) – A dict which indicates a tuple of name and PortexType.
- **imports** (*graviti.portex.package.Imports*) –

**class** *graviti.portex.factory.FieldsFactory*(*decl, imports*)

Bases: *Factory*

The template factory for a Fields.

#### Parameters

- **decl** (*List[Union[Dict[str, Any], str]]*) – A list which indicates a Fields.
- **imports** (*graviti.portex.package.Imports*) –

*graviti.portex.factory.mapping\_unpack\_factory\_creator*(*decl, ptype*)

Check the object unpack grammar and returns the corresponding factory.

#### Parameters

- **decl** (*str*) – The parameter decalaration.
- **ptype** (*graviti.portex.ptype.PType*) – The parameter type of the input.

**Raises** **ValueError** – When the object unpack grammar is incorrect.

**Returns** A VariableFactory instance according to the input.

**Return type** *VariableFactory*

*graviti.portex.factory.type\_factory\_creator*(*decl, imports*)

Check the input and returns the corresponding type factory.

#### Parameters

- **decl** (*Dict[str, Any]*) – A dict which indicates a portex type or has object unpack grammar.
- **imports** (*graviti.portex.package.Imports*) – The Imports instance to specify the import scope of the template.

**Raises** **ValueError** – When setting the type name as a parameter.

**Returns** A TypeFactory or a VariableFactory instance.

**Return type** *Union[TypeFactory, VariableFactory]*

*graviti.portex.factory.string\_factory\_creator*(*decl, ptype=PTYPE.Any*)

Check whether the input string is variable and returns the corresponding factory.

#### Parameters

- **decl** (*str*) – A string which indicates a constant or a variable.
- **ptype** (*graviti.portex.ptype.PType*) – The parameter type of the string.

**Returns** A VariableFactory or a ConstantFactory instance according to the input.

**Return type** *Union[VariableFactory, ConstantFactory[str]]*

*graviti.portex.factory.factory\_creator*(*decl, imports, ptype=PTYPE.Any*)

Check input type and returns the corresponding factory.

#### Parameters

- **decl** (*Any*) – A template which indicates any Portex object.
- **imports** (*Optional[graviti.portex.package.Imports]*) – The Imports instance to specify the import scope of the template.

- **ptype** (*graviti.portex.ptype.PType*) – The parameter type of the input.

**Returns** A Factory instance according the input.

**Return type** *Factory*

## graviti.portex.field

Portex record field related classes.

## Module Contents

### Classes

<i>FrozenFields</i>	Represents a frozen fields dict.
<i>Fields</i>	Represents a Portex record fields dict.
<i>ConnectedFields</i>	Fields composed of FrozenFields and Fields.

### Attributes

<i>UnionFields</i>
--------------------

**class** graviti.portex.field.**FrozenFields**(*items=None*)

Bases: graviti.utility.FrozenNameOrderedDict[*graviti.portex.base.PortexType*]

Represents a frozen fields dict.

**Parameters** **items** (*Union[Iterable[Tuple[str, \_V]], Mapping[str, \_V], None]*) –

**insert**(*self, index, name, portex\_type*)

Insert the name and portex\_type at the index.

#### Parameters

- **index** (*int*) – The index to insert the field.
- **name** (*str*) – The name of the field to be inserted.
- **portex\_type** (*graviti.portex.base.PortexType*) – The portex\_type of the field to be inserted.

**Raises** **TypeError** – When calling this method of FrozenFields.

**Return type** None

**astype**(*self, name, portex\_type*)

Convert the type of the field with the given name to the new PortexType.

#### Parameters

- **name** (*str*) – The name of the field to convert.
- **portex\_type** (*graviti.portex.base.PortexType*) – The new PortexType of the field to convert to.

**Raises `TypeError`** – When calling this method of `FrozenFields`.

**Return type** `None`

**`rename(self, old_name, new_name)`**

Rename the name of a field.

**Parameters**

- **`old_name`** (`str`) – The current name of the field to be renamed.
- **`new_name`** (`str`) – The new name of the field to assign.

**Raises `TypeError`** – When calling this method of `FrozenFields`.

**Return type** `None`

**`class graviti.portex.field.Fields(fields=None)`**

Bases: `graviti.utility.NameOrderedDict[graviti.portex.base.PortexType], FrozenFields`

Represents a Portex record fields dict.

**Parameters** **`fields`** (`Union[Iterable[Tuple[str, graviti.portex.base.PortexType]], Mapping[str, graviti.portex.base.PortexType], None]`)

–

**`property imports(self)`**

Get the Fields imports.

**Returns** The Imports instance of this Fields.

**Return type** `graviti.portex.package.Imports`

**`insert(self, index, name, portex_type)`**

Insert the name and portex\_type at the index.

**Parameters**

- **`index`** (`int`) – The index to insert the field.
- **`name`** (`str`) – The name of the field to be inserted.
- **`portex_type`** (`graviti.portex.base.PortexType`) – The portex\_type of the field to be inserted.

**Raises `KeyError`** – When the name already exists in the Fields.

**Return type** `None`

**`astype(self, name, portex_type)`**

Convert the type of the field with the given name to the new PortexType.

**Parameters**

- **`name`** (`str`) – The name of the field to convert.
- **`portex_type`** (`graviti.portex.base.PortexType`) – The new PortexType of the field to convert to.

**Raises `KeyError`** – When the name does not exist in the Fields.

**Return type** `None`

**`rename(self, old_name, new_name)`**

Rename the name of a field.

**Parameters**

- **old\_name** (*str*) – The current name of the field to be renamed.
- **new\_name** (*str*) – The new name of the field to assign.

**Return type** `None`

**classmethod** `from_pyobj(cls, content, imports=None)`

Create Portex fields dict instance from python list.

**Parameters**

- **content** (*List[Dict[str, Any]]*) – A python list representing a Portex fields dict.
- **imports** (*Optional[graviti.portex.package.Imports]*) – The imports of the Portex fields dict.

**Returns** A Portex fields dict instance created from the input python list.

**Return type** *Fields*

**to\_pyobj**(*self*)

Dump the instance to a python list.

**Returns** A Python List representation of the fields dict.

**Return type** *List[Dict[str, Any]]*

**to\_pyarrow**(*self*)

Convert the fields to a list of PyArrow Field.

**Returns** A list of PyArrow Field representing the fields of Portex record.

**Return type** *List[pyarrow.Field]*

**copy**(*self*)

Get a copy of the fields.

**Returns** A copy of the fields.

**Parameters** **self** (*\_T*) –

**Return type** *\_T*

`graviti.portex.field.UnionFields`

**class** `graviti.portex.field.ConnectedFields(multi_fields)`

Bases: *MutableMapping[str, graviti.portex.base.PortexType]*

Fields composed of FrozenFields and Fields.

**Raises** **ValueError** – When there as repeated field names.

**Parameters** **multi\_fields** (*Iterable[UnionFields]*) – The FrozenFields and Fields.

**insert**(*self, index, name, portex\_type*)

Insert the name and portex\_type at the index.

**Parameters**

- **index** (*int*) – The index to insert the field.
- **name** (*str*) – The name of the field to be inserted.
- **portex\_type** (*graviti.portex.base.PortexType*) – The portex\_type of the field to be inserted.

**Raises**

- **ValueError** – When the name already exists in the fields.

- **TypeError** – When trying to insert a field into FrozenFields.

**Return type** None

**astype**(*self*, *name*, *portex\_type*)

Convert the type of the field with the given name to the new PortexType.

**Parameters**

- **name** (*str*) – The name of the field to convert.
- **portex\_type** ([graviti.portex.base.PortexType](#)) – The new PortexType of the field to convert to.

**Return type** None

**rename**(*self*, *old\_name*, *new\_name*)

Rename the name of a field.

**Parameters**

- **old\_name** (*str*) – The current name of the field to be renamed.
- **new\_name** (*str*) – The new name of the field to assign.

**Return type** None

## **graviti.portex.package**

Package related class.

## **Module Contents**

### **Classes**

<i>Package</i>	The base class of Portex package.
<i>BuiltinPackage</i>	The builtin Portex package used to manage builtin types.
<i>LocalPackage</i>	The local Portex package used to manage local types.
<i>ExternalPackage</i>	The external Portex package used to manage external types.
<i>Subpackage</i>	The subset of Portex package, used in <a href="#">Imports</a> .
<i>Packages</i>	The package manager to manage different Portex packages.
<i>Imports</i>	The imports of the Portex template type.

### **Attributes**

<i>packages</i>
-----------------

**class** `graviti.portex.package.Package`

Bases: `graviti.utility.AttrDict[_T]`

The base class of Portex package.

**class** graviti.portex.package.**BuiltinPackage**

Bases: [Package](#)[Type[[graviti.portex.builtin.PortexBuiltinType](#)]]

The builtin Portex package used to manage builtin types.

**class** graviti.portex.package.**LocalPackage**

Bases: [Package](#)[Type[[graviti.portex.base.PortexType](#)]]

The local Portex package used to manage local types.

**class** graviti.portex.package.**ExternalPackage**(url, revision)

Bases: [Package](#)[Type[[graviti.portex.external.PortexExternalType](#)]]

The external Portex package used to manage external types.

**Parameters**

- **url** (str) – The git repo url of the external package.
- **revision** (str) – The git repo revision (tag/commit) of the external package.

**property** **repo**(self)

The repo string of the package.

**Returns** The “<url>@<rev>” format repo string.

**Return type** str

**class** graviti.portex.package.**Subpackage**(package)

Bases: [graviti.utility.UserMapping](#)[str, Type[[graviti.portex.external.PortexExternalType](#)]]

The subset of Portex package, used in [Imports](#).

**Parameters** **package** ([ExternalPackage](#)) – The source package of this subpackage.

**classmethod** **from\_pyobj**(cls, content)

Create [Subpackage](#) instance from python dict.

**Parameters** **content** (Dict[str, Any]) – A python dict representing a subpackage.

**Returns** A [Subpackage](#) instance created from the input python dict.

**Return type** [Subpackage](#)

**to\_pyobj**(self)

Dump the instance to a python dict.

**Returns** A python dict representation of the [Subpackage](#) instance.

**Return type** Dict[str, Any]

**class** graviti.portex.package.**Packages**

The package manager to manage different Portex packages.

graviti.portex.package.**packages**

**class** graviti.portex.package.**Imports**

Bases: Mapping[str, Type[[graviti.portex.base.PortexType](#)]], [graviti.utility.ReprMixin](#)

The imports of the Portex template type.

**Parameters** **package** – The package the portex belongs to.

**update**(self, other)

Update the imports with another imports.



**Parameters** **other** (*Imports*) – An *Imports* instance whose types need to be updated to this imports.

**Return type** None

**classmethod** **from\_pyobj**(*cls*, *content*)

Create *Imports* instance from python list.

**Parameters**

- **content** (*List[Dict[str, Any]]*) – A python list representing imported types.
- **cls** (*Type[\_I]*) –

**Returns** A *Imports* instance created from the input python list.

**Return type** *\_I*

**to\_pyobj**(*self*)

Dump the instance to a python list.

**Returns** A python list representation of the Portex imported types.

**Return type** *List[Dict[str, Any]]*

**add\_subpackage**(*self*, *subpackage*)

Add subpackage to this *Imports* instance.

**Parameters** **subpackage** (*Subpackage*) – The subpackage which needs to be added.

**Raises** **KeyError** – When there are duplicate names in the imports instance.

**Return type** None

**graviti.portex.param**

Parameter related classes.

## Module Contents

### Classes

<i>Param</i>	Represents a parameter of a portex type.
<i>Params</i>	Represents all parameters of a portex type.

### Functions

<i>param</i> (default = Parameter.empty, options = None, ptype = PTYPE.Any)	The factory function of Param.
---	--------------------------------

**graviti.portex.param.param**(*default=Parameter.empty, options=None, ptype=PTYPE.Any*)  
The factory function of Param.

**Parameters**

- **default** (*Any*) – The default value of the parameter.

- **options** (*Optional[Iterable[Any]]*) – All possible values of the parameter.
- **ptype** (*graviti.portex.ptype.PType*) – The parameter type.

**Returns** A tuple which contains “default”, “options” and “ptype”.

**Return type** *Any*

**class** `graviti.portex.param.Param(name, default=_empty, options=None, ptype=PTYPE.Any)`

Bases: `inspect.Parameter`

Represents a parameter of a portex type.

#### Parameters

- **name** (*str*) – The name of the parameter.
- **default** (*Any*) – The default value of the parameter.
- **options** (*Optional[Iterable[Any]]*) – All possible values of the parameter.
- **ptype** (*graviti.portex.ptype.PType*) – The parameter type.

**classmethod** `from_pyobj(cls, pyobj, ptype=PTYPE.Any)`

Create Param instance from python dict.

#### Parameters

- **pyobj** (*Dict[str, Any]*) – A python dict representing a parameter.
- **ptype** (*graviti.portex.ptype.PType*) – The parameter type.

**Returns** A Param instance created from the input python dict.

**Return type** *Param*

**property** `required(self)`

Whether this parameter is a required parameter.

**Returns** True for required and False for non-required parameter.

**Return type** `bool`

**to\_pyobj** (*self*)

Dump the instance to a python dict.

**Returns** A python dict representation of the Param.

**Return type** `Dict[str, Any]`

**check** (*self, arg*)

Check the validity of the parameter.

**Parameters** **arg** (*Any*) – The argument which needs to be checked.

**Returns** The argument after checking.

**Raises** **ValueError** – Raise when the argument is not in options.

**Return type** *Any*

**load** (*self, content, imports*)

Create an instance of the parameter type from the python content.

#### Parameters

- **content** (*Any*) – A python presentation of the parameter type.

- **imports** (`graviti.portex.package.Imports`) – The imports of the parameter type.

**Returns** An instance of the parameter type.

**Return type** *Any*

**dump**(*self*, *arg*)

Dump the parameter type instance into the python presentation.

**Parameters** **arg** (*Any*) – The parameter type instance.

**Returns** The python presentation of the input instance.

**Return type** *Any*

**class** `graviti.portex.param.Params`(*values=None*)

Bases: `graviti.utility.UserMapping[str, Param]`

Represents all parameters of a portex type.

**Parameters** **values** (*Optional[Mapping[str, Param]]*) – The parameters mapping.

**classmethod** **from\_pyobj**(*cls*, *pyobj*, *keys*)

Create Params instance from python list.

**Parameters**

- **pyobj** (*List[Dict[str, Any]]*) – A python dict representing parameters.
- **keys** (*Dict[str, Any]*) – A python dict containing parameter types.

**Returns** A Params instance created from the input python list.

**Return type** *Params*

**to\_pyobj**(*self*)

Dump the instance to a python list.

**Returns** A python list representation of the Params.

**Return type** *List[Dict[str, Any]]*

**add**(*self*, *value*)

Add a parameter.

**Parameters** **value** (*Param*) – The parameter which needs to be added to this instance.

**Raises** **KeyError** – When the parameter name is duplicated.

**Return type** *None*

**update**(*self*, *values*)

Update the parameters.

**Parameters** **values** (*Mapping[str, Param]*) – The parameters which need to be updated to this instance.

**Return type** *None*

**get\_signature**(*self*)

Get the python inspect Signature from parameters.

**Returns** The Signature instance created by all parameters in this instance.

**Return type** *inspect.Signature*

## graviti.portex.ptype

Parameter type related classes.

## Module Contents

### Classes

<i>ParameterType</i>	The base class of parameter type.
<i>Any</i>	Unconstrained parameter type.
<i>Boolean</i>	Parameter type for JSON Boolean.
<i>Array</i>	Parameter type for JSON Array.
<i>Mapping</i>	Parameter type for JSON object.
<i>Number</i>	Parameter type for JSON number.
<i>Integer</i>	Parameter type for JSON integer.
<i>String</i>	Parameter type for JSON string.
<i>Enum</i>	Parameter type for Portex enum values.
<i>Fields</i>	Parameter type for Portex record fields.
<i>PortexType</i>	Parameter type for Portex type.

### Attributes

<i>PType</i>
--------------

---

#### **class** graviti.portex.ptype.ParameterType

The base class of parameter type.

##### **static** `check(arg)`

Check the parameter type.

**Parameters** `arg` (*Any*) – The argument which needs to be checked.

**Returns** The input argument unchanged.

**Return type** *Any*

##### **static** `load(content, _=None)`

Create an instance of the parameter type from the python content.

##### **Parameters**

- **content** (*Any*) – A python presentation of the parameter type.
- **\_** (*Optional*[`graviti.portex.package.Imports`]) – The imports of the parameter type.

**Returns** An instance of the parameter type.

**Return type** *Any*

##### **static** `dump(arg)`

Dump the parameter type instance into the python presentation.

**Parameters** `arg` (*Any*) – The parameter type instance.

**Returns** The python presentation of the input instance.

**Return type** *Any*

`graviti.portex.ptype.PType`

**class** `graviti.portex.ptype.Any`

Bases: *ParameterType*

Unconstrained parameter type.

**class** `graviti.portex.ptype.Boolean`

Bases: `_JsonType`

Parameter type for JSON Boolean.

**class** `graviti.portex.ptype.Array`

Bases: `_JsonType`

Parameter type for JSON Array.

**class** `graviti.portex.ptype.Mapping`

Bases: `_JsonType`

Parameter type for JSON object.

**class** `graviti.portex.ptype.Number`

Bases: `_JsonType`

Parameter type for JSON number.

**class** `graviti.portex.ptype.Integer`

Bases: `_JsonType`

Parameter type for JSON integer.

**class** `graviti.portex.ptype.String`

Bases: `_JsonType`

Parameter type for JSON string.

**class** `graviti.portex.ptype.Enum`

Bases: *ParameterType*

Parameter type for Portex enum values.

**static** `check(arg)`

Check and transfer the parameter type.

**Parameters** `arg` (*Any*) – The argument which needs to be checked.

**Returns** A list of enum values created by the input argument.

**Return type** *graviti.portex.enum.EnumValues*

**static** `load(content, _=None)`

Create Portex EnumValues instance from python object.

**Parameters**

- **content** (*Union[Dict[int, graviti.portex.enum.EnumValueType], List[graviti.portex.enum.EnumValueType], None]*) – A python list or dict representing a EnumValues.
- **\_** (*Optional[graviti.portex.package.Imports]*) – The imports of the Portex field.

**Returns** A Portex EnumValues instance created from the input python list or dict.

**Raises** **TypeError** – When the input enum values is not in list or dict format.

**Return type** Optional[*graviti.portex.enum.EnumValues*]

**static dump**(*arg*)

Dump the input Portex EnumValues instance to a python list or dict.

**Parameters** **arg** (*graviti.portex.enum.EnumValues*) – A Portex EnumValues instance.

**Returns** A Python list or dict representation of the Portex enum values.

**Return type** Union[Dict[int, *graviti.portex.enum.EnumValueType*],  
List[*graviti.portex.enum.EnumValueType*]]

**class** *graviti.portex.ptype.Fields*

Bases: *ParameterType*

Parameter type for Portex record fields.

**static check**(*arg*)

Check and transfer the parameter type.

**Parameters** **arg** (*Any*) – The argument which needs to be checked.

**Returns** A *Fields* instance created by the input argument.

**Return type** *graviti.portex.field.Fields*

**static load**(*content*, *imports=None*)

Create Portex field list instance from python list.

**Parameters**

- **content** (*Optional[List[*Any*]]*) – A python list representing a Portex field list.
- **imports** (*Optional[*graviti.portex.package.Imports*]*) – The imports of the Portex field.

**Returns** A Portex field list instance created from the input python list.

**Return type** *graviti.portex.field.Fields*

**static dump**(*arg*)

Dump the input Portex field list instance to a python list.

**Parameters** **arg** (*graviti.portex.field.Fields*) – A Portex field list instance.

**Returns** A Python list representation of the Portex field list.

**Return type** List[*Any*]

**class** *graviti.portex.ptype.PortexType*

Bases: *ParameterType*

Parameter type for Portex type.

**static check**(*arg*)

Check the parameter type.

**Parameters** **arg** (*Any*) – The argument which needs to be checked.

**Returns** The input argument unchanged.

**Raises** **TypeError** – When the input argument is not a Portex type.

**Return type** *graviti.portex.base.PortexType*

**static load**(*content*, *imports=None*)  
Create Portex type instance from python dict.

**Parameters**

- **content** (*Optional*[*Dict*[*str*, *Any*]]) – A python dict representing a Portex type.
- **imports** (*Optional*[*graviti.portex.package.Imports*]) – The imports of the Portex type.

**Returns** A Portex type instance created from the input python dict.

**Return type** *Optional*[*graviti.portex.base.PortexType*]

**static dump**(*arg*)  
Dump the instance to a python dict.

**Parameters** **arg** (*graviti.portex.base.PortexType*) – A Portex type instance.

**Returns** A python dict representation of the Portex type.

**Return type** *Dict*[*str*, *Any*]

## graviti.portex.register

The portex type register related classes.

## Module Contents

### Classes

<i>ContainerRegister</i>	The class decorator to connect portex type and the data container.
<i>ExternalContainerRegister</i>	The class decorator to connect portex external type and the data container.
<i>ExternalElementResgister</i>	The class decorator to connect portex external type and the element class.
<i>PyArrowConversionRegister</i>	Register the Portex type to set the conversion from PyArrow to Portex.

### Attributes

<i>STANDARD_URL</i>
---------------------

`graviti.portex.register.STANDARD_URL =`  
`https://github.com/Project-OpenBytes/portex-standard`

**class** `graviti.portex.register.ContainerRegister`(\**portex\_types*)

The class decorator to connect portex type and the data container.

**Parameters** **portex\_types** (*Type*[*graviti.portex.base.PortexType*]) – The portex types needs to be connected.

**class** graviti.portex.register.**ExternalContainerRegister**(url, revision, \*names)

The class decorator to connect portex external type and the data container.

**Parameters**

- **url** (str) – The git repo url of the external package.
- **revision** (str) – The git repo revision (tag/commit) of the external package.
- **name** – The portex external type name.
- **names** (str) –

**class** graviti.portex.register.**ExternalElementResgister**(url, revision, \*names)

The class decorator to connect portex external type and the element class.

**Parameters**

- **url** (str) – The git repo url of the external package.
- **revision** (str) – The git repo revision (tag/commit) of the external package.
- **name** – The portex external type name.
- **names** (str) –

**class** graviti.portex.register.**PyArrowConversionRegister**(\*pyarrow\_type\_ids)

Register the Portex type to set the conversion from PyArrow to Portex.

**Parameters** **pyarrow\_type\_ids** (int) – The id of the corresponding PyArrow types.

## graviti.utility

Utility module.

## Submodules

### graviti.utility.attr

Attr related class.

## Module Contents

### Classes

---

*AttrDict*

A dict which allows for attr-style access of values.

---

**class** graviti.utility.attr.**AttrDict**

Bases: `_AttrDict[_T]`, `Mapping[str, _T]`

A dict which allows for attr-style access of values.

**get**(self, key: str) → Optional[\_T]

**get**(self, key: str, default: \_D = ...) → Union[\_D, \_T]

Return the value for the key if it is in the dict, else default.

**Parameters**



- **key** – The key for dict, which can be any immutable type.
- **default** – The value to be returned if key is not in the dict.

**Returns** The value for the key if it is in the dict, else default.

## graviti.utility.collections

Basic concepts of user-defined objects.

## Module Contents

### Classes

<i>UserSequence</i>	UserSequence is a user-defined wrapper around sequence objects.
<i>UserMutableSequence</i>	UserMutableSequence is a user-defined wrapper around mutable sequence objects.
<i>UserMapping</i>	UserMapping is a user-defined wrapper around mapping objects.
<i>UserMutableMapping</i>	UserMutableMapping is a user-defined wrapper around mutable mapping objects.
<i>FrozenNameOrderedDict</i>	This class is an immutable dict of ordered elements, supports searching the element by index.
<i>NameOrderedDict</i>	This class is a dict of ordered elements, supports searching the element by its index.

### class graviti.utility.collections.UserSequence

Bases: Sequence[\_T], [graviti.utility.repr.ReprMixin](#)

UserSequence is a user-defined wrapper around sequence objects.

**index**(self, value, start=0, stop=maxsize)

Return the first index of the value.

#### Parameters

- **value** (\_T) – The value to be found.
- **start** (int) – The start index of the subsequence.
- **stop** (int) – The end index of the subsequence.

**Returns** The First index of value.

**Return type** int

**count**(self, value)

Return the number of occurrences of value.

**Parameters** **value** (\_T) – The value to be counted the number of occurrences.

**Returns** The number of occurrences of value.

**Return type** int

### class graviti.utility.collections.UserMutableSequence

Bases: MutableSequence[\_T], [UserSequence](#)[\_T]

UserMutableSequence is a user-defined wrapper around mutable sequence objects.

**insert**(*self*, *index*, *value*)

Insert object before index.

**Parameters**

- **index** (*int*) – Position of the mutable sequence.
- **value** (*\_T*) – Element to be inserted into the mutable sequence.

**Return type** None

**append**(*self*, *value*)

Append object to the end of the mutable sequence.

**Parameters** **value** (*\_T*) – Element to be appended to the mutable sequence.

**Return type** None

**clear**(*self*)

Remove all items from the mutable sequence.

**Return type** None

**extend**(*self*, *values*)

Extend mutable sequence by appending elements from the iterable.

**Parameters** **values** (*Iterable[\_T]*) – Elements to be Extended into the mutable sequence.

**Return type** None

**reverse**(*self*)

Reverse the items of the mutable sequence in place.

**Return type** None

**pop**(*self*, *index=-1*)

Return the item at index (default last) and remove it from the mutable sequence.

**Parameters** **index** (*int*) – Position of the mutable sequence.

**Returns** Element to be removed from the mutable sequence.

**Return type** *\_T*

**remove**(*self*, *value*)

Remove the first occurrence of value.

**Parameters** **value** (*\_T*) – Element to be removed from the mutable sequence.

**Return type** None

**class** graviti.utility.collections.UserMapping

Bases: Mapping[*\_K*, *\_V*], [graviti.utility.repr.ReprMixin](#)

UserMapping is a user-defined wrapper around mapping objects.

**get**(*self*, *key*: *\_K*) → Optional[*\_V*]

**get**(*self*, *key*: *\_K*, *default*: Union[*\_V*, *\_T*] = ...) → Union[*\_V*, *\_T*]

Return the value for the key if it is in the dict, else default.

**Parameters**

- **key** – The key for dict, which can be any immutable type.
- **default** – The value to be returned if key is not in the dict.

**Returns** The value for the key if it is in the dict, else default.

**items**(*self*)

Return a new view of the (key, value) pairs in dict.

**Returns** The (key, value) pairs in dict.

**Return type** AbstractSet[Tuple[\_K, \_V]]

**keys**(*self*)

Return a new view of the keys in dict.

**Returns** The keys in dict.

**Return type** AbstractSet[\_K]

**values**(*self*)

Return a new view of the values in dict.

**Returns** The values in dict.

**Return type** ValuesView[\_V]

**class** graviti.utility.collections.**UserMutableMapping**

Bases: MutableMapping[\_K, \_V], [UserMapping](#)[\_K, \_V]

UserMutableMapping is a user-defined wrapper around mutable mapping objects.

**clear**(*self*)

Remove all items from the mutable mapping object.

**Return type** None

**pop**(*self*, key: \_K) → \_V

**pop**(*self*, key: \_K, default: Union[\_V, \_T] = ...) → Union[\_V, \_T]

Remove specified item and return the corresponding value.

**Parameters**

- **key** – The key for dict, which can be any immutable type.
- **default** – The value to be returned if the key is not in the dict and it is given.

**Returns** Value to be removed from the mutable mapping object.

**popitem**(*self*)

Remove and return a (key, value) pair as a tuple.

Pairs are returned in LIFO (last-in, first-out) order.

**Returns** A (key, value) pair as a tuple.

**Return type** Tuple[\_K, \_V]

**setdefault**(*self*, key, default=None)

Set the value of the item with the specified key.

If the key is in the dict, return the corresponding value. If not, insert the key with a value of default and return default.

**Parameters**

- **key** (\_K) – The key for dict, which can be any immutable type.
- **default** (\_V) – The value to be set if the key is not in the dict.

**Returns** The value for key if it is in the dict, else default.

**Return type** `_V`

**update**(*self*, *\_\_m*: *Mapping*[*\_K*, *\_V*], *\*\*kwargs*: *\_V*) → None

**update**(*self*, *\_\_m*: *Iterable*[*Tuple*[*\_K*, *\_V*]], *\*\*kwargs*: *\_V*) → None

**update**(*self*, *\*\*kwargs*: *\_V*) → None

Update the dict.

**Parameters**

- ***\_\_m*** – A dict object, a generator object yielding a (key, value) pair or other object which has a *.keys()* method.
- ***\*\*kwargs*** – The value to be added to the mutable mapping.

**class** `graviti.utility.collections.FrozenNameOrderedDict`(*items=None*)

Bases: `Mapping`[`str`, `_V`], `graviti.utility.repr.ReprMixin`

This class is an immutable dict of ordered elements, supports searching the element by index.

**Parameters** ***items*** (*Union*[*Iterable*[*Tuple*[*str*, *\_V*]], *Mapping*[*str*, *\_V*], *None*]) – The items need to be stored into the FrozenNameOrderedDict.

**class** `graviti.utility.collections.NameOrderedDict`(*items=None*)

Bases: `MutableMapping`[`str`, `_V`], `FrozenNameOrderedDict`[`_V`]

This class is a dict of ordered elements, supports searching the element by its index.

**Parameters** ***items*** (*Union*[*Iterable*[*Tuple*[*str*, *\_V*]], *Mapping*[*str*, *\_V*], *None*]) – The items need to be stored into the NameOrderedDict.

**popitem**(*self*)

Remove and return a (key, value) pair as a tuple.

Pairs are returned in LIFO (last-in, first-out) order.

**Raises** **KeyError** – When the dict is empty.

**Returns** A (key, value) pair as a tuple.

**Return type** `Tuple`[`str`, `_V`]

`graviti.utility.common`

Common tools.

## Module Contents

### Classes

---

*ImageMocker*

---

Raise import PIL error for data loader.

---

## Functions

<code>urlnorm(url)</code>	Normalized the input url by removing the trailing slash.
<code>locked(func)</code>	The decorator to add threading lock for methods.
<code>shorten(origin)</code>	Return the first 7 characters of the original string.
<code>convert_iso_to_datetime(iso)</code>	Convert iso 8601 format string to datetime format time with local timezone.
<code>convert_datetime_to_gmt(utctime)</code>	Convert datetime to gmt format string.

## Attributes

<code>locks</code>
--------------------

`graviti.utility.common.locks :DefaultDict[int, threading.Lock]`

`graviti.utility.common.urlnorm(url)`

Normalized the input url by removing the trailing slash.

**Parameters** `url (str)` – the url needs to be normalized.

**Returns** The normalized url.

**Return type** `str`

`graviti.utility.common.locked(func)`

The decorator to add threading lock for methods.

**Parameters** `func (_CallableWithoutReturnValue)` – The method needs to add threading lock.

**Returns** The method with theading locked.

**Return type** `_CallableWithoutReturnValue`

`graviti.utility.common.shorten(origin)`

Return the first 7 characters of the original string.

**Parameters** `origin (str)` – The string needed to be shortened.

**Returns** A string of length 7.

**Return type** `str`

`graviti.utility.common.convert_iso_to_datetime(iso)`

Convert iso 8601 format string to datetime format time with local timezone.

**Parameters** `iso (str)` – The iso 8601 format string.

**Returns** The datetime format time with local timezone.

**Return type** `datetime.datetime`

`graviti.utility.common.convert_datetime_to_gmt(utctime)`

Convert datetime to gmt format string.

**Parameters** `utctime (datetime.datetime)` – The datetime with utc timezone.

**Returns** The gmt format string.

**Return type** `str`

**class** graviti.utility.common.**ImageMocker**(*message*)

Raise import PIL error for data loader.

**Parameters** *message* (*str*) –

**graviti.utility.engine**

Engine control related classes.

## Module Contents

### Classes

<i>Mode</i>	This class defines the engine mode and includes 'LOCAL' and 'ONLINE'.
<i>Online</i>	An engine controller used to start and stop the online mode.
<i>Engine</i>	This is a base class defining the Engine mode.

### Attributes

<i>engine</i>
---------------

**class** graviti.utility.engine.**Mode**

Bases: `enum.Enum`

This class defines the engine mode and includes 'LOCAL' and 'ONLINE'.

**class** graviti.utility.engine.**Online**(*\_engine*)

An engine controller used to start and stop the online mode.

**Parameters** *\_engine* (*Engine*) –

**class** graviti.utility.engine.**Engine**

This is a base class defining the Engine mode.

**online**(*self*)

Init a Online instance.

**Returns** the Online instance.

**Return type** *Online*

graviti.utility.engine.**engine**

## graviti.utility.itertools

The implementation of iteration tools.

### Module Contents

#### Functions

<code>chunked(iterable, n)</code>	Break an iterable instance into tuples of length n.
-----------------------------------	---

graviti.utility.itertools.**chunked**(*iterable*, *n*)

Break an iterable instance into tuples of length n.

#### Parameters

- **iterable** (*Iterable[\_T]*) – The input iterable instance which needs to be broken into tuples of length n.
- **n** (*int*) – The length of each yielded tuples.

**Yields** The tuples of length n.

**Return type** Iterator[Tuple[\_T, Ellipsis]]

#### Examples

```
>>> list(chunked(range(9), 3))
[(0, 1, 2), (3, 4, 5), (6, 7, 8)]
```

The last yielded tuple may have fewer than n items if the length of the input iterable instance is not divisible by n:

```
>>> list(chunked(range(10), 3))
[(0, 1, 2), (3, 4, 5), (6, 7, 8), (9,)]
```

## graviti.utility.log

The implementation of logging utilities.

### Module Contents

#### Classes

<code>RequestLogging</code>	This class used to lazy load request to logging.
<code>ResponseLogging</code>	This class used to lazy load response to logging.

## Functions

---

<code>dump_request_and_response(response)</code>	Dumps http request and response.
--	----------------------------------

---

## Attributes

---

`REQUEST_TEMPLATE`

---



---

`RESPONSE_TEMPLATE`

---

`graviti.utility.log.REQUEST_TEMPLATE = Multiline-String`

```

1 =====
2 ##### HTTP Request #####
3 "url": {}
4 "method": {}
5 "headers": {}
6 "body": {}

```

`graviti.utility.log.RESPONSE_TEMPLATE = Multiline-String`

```

1 ##### HTTP Response #####
2 "url": {}
3 "status_code": {}
4 "reason": {}
5 "headers": {}
6 "content": {}
7 "cost_time": {}s
8 =====

```

**class** `graviti.utility.log.RequestLogging(request)`

This class used to lazy load request to logging.

**Parameters** `request` (`requests.models.PreparedRequest`) – The request of the request.

**class** `graviti.utility.log.ResponseLogging(response)`

This class used to lazy load response to logging.

**Parameters** `response` (`requests.models.Response`) – The response of the request.

`graviti.utility.log.dump_request_and_response(response)`

Dumps http request and response.

**Parameters** `response` (`requests.models.Response`) – Http response and response.

### Returns

Http request and response for logging, sample:

```

=====
##### HTTP Request #####
"url": https://gas.graviti.cn/gatewayv2/content-store/putObject
"method": POST

```

(continues on next page)



(continued from previous page)

```

"headers": {
    "User-Agent": "python-requests/2.23.0",
    "Accept-Encoding": "gzip, deflate",
    "Accept": "*/*",
    "Connection": "keep-alive",
    "X-Token": "c3b1808b21024eb38f066809431e5bb9",
    "Content-Type": "multipart/form-data;␣
↪boundary=5adff1fc0524465593d6a9ad68aad7f9",
    "Content-Length": "330001"
}
"body":
--5adff1fc0524465593d6a9ad68aad7f9
b'Content-Disposition: form-data; name="contentSetId"\r\n\r\n'
b'e6110ff1-9e7c-4c98-aaf9-5e35522969b9'

--5adff1fc0524465593d6a9ad68aad7f9
b'Content-Disposition: form-data; name="filePath"\r\n\r\n'
b'4.jpg'

--5adff1fc0524465593d6a9ad68aad7f9
b'Content-Disposition: form-data; name="fileData"; filename="4.jpg"\r\n
↪n\r\n'
[329633 bytes of object data]

--5adff1fc0524465593d6a9ad68aad7f9--

##### HTTP Response #####
"url": https://gas.graviti.cn/gatewayv2/content-stor
"status_code": 200
"reason": OK
"headers": {
    "Date": "Sat, 23 May 2020 13:05:09 GMT",
    "Content-Type": "application/json;charset=utf-8",
    "Content-Length": "69",
    "Connection": "keep-alive",
    "Access-Control-Allow-Origin": "*",
    "X-Kong-Upstream-Latency": "180",
    "X-Kong-Proxy-Latency": "112",
    "Via": "kong/2.0.4"
}
"content": {
    "success": true,
    "code": "DATACENTER-0",
    "message": "success",
    "data": {}
}
"cost_time": 0.0813691616058
=====

```

**Return type** str

### graviti.utility.repr

Repr related methods.

### Module Contents

#### Classes

<i>ReprType</i>	ReprType is an enumeration type.
<i>ReprMixin</i>	ReprMixin provides customized repr config and method.

#### Attributes

<i>MAX_REPR_ROWS</i>
<i>INDENT</i>
<i>repr_config</i>

graviti.utility.repr.**MAX\_REPR\_ROWS** = 10

graviti.utility.repr.**INDENT**

**class** graviti.utility.repr.**ReprType**

Bases: enum.Enum

ReprType is an enumeration type.

It defines the repr strategy type and includes 'INSTANCE', 'SEQUENCE' and 'MAPPING'.

**class** graviti.utility.repr.**ReprMixin**

ReprMixin provides customized repr config and method.

graviti.utility.repr.**repr\_config**

### graviti.utility.requests

The implementation of request related tools.

### Module Contents

#### Classes

<i>Config</i>	This is a base class defining the concept of Request Config.
<i>TimeoutHTTPAdapter</i>	This class defines the http adapter for setting the timeout value.

continues on next page

Table 1.96 – continued from previous page

<i>UserSession</i>	This class defines UserSession.
<i>UserResponse</i>	This class used to read data from Response with stream method.

## Functions

<i>get_session()</i>	Create and return a session per PID so each sub-processes will use their own session.
<i>submit_multithread_tasks</i> (function, arguments, *, jobs)	Multi-thread framework.

## Attributes

<i>logger</i>
<i>config</i>
<i>SESSIONS</i>

graviti.utility.requests.**logger**

**class** graviti.utility.requests.**Config**

This is a base class defining the concept of Request Config.

**max\_retries**

Maximum retry times of the request.

**allowed\_retry\_methods**

The allowed methods for retrying request.

**allowed\_retry\_status**

The allowed status for retrying request. If both methods and status are fitted, the retrying strategy will work.

**timeout**

Timeout value of the request in seconds.

**is\_internal**

Whether the request is from internal.

graviti.utility.requests.**config**

**class** graviti.utility.requests.**TimeoutHTTPAdapter**(\*args, timeout=None, \*\*kwargs)

Bases: requests.adapters.HTTPAdapter

This class defines the http adapter for setting the timeout value.

### Parameters

- **\*args** – Extra arguments to initialize TimeoutHTTPAdapter.
- **timeout** (*Optional[int]*) – Timeout value of the post request in seconds.
- **\*\*kwargs** – Extra keyword arguments to initialize TimeoutHTTPAdapter.

- **args** (*Any*) –
- **kwargs** (*Any*) –

**send**(*self*, *request*, *stream=False*, *timeout=None*, *verify=True*, *cert=None*, *proxies=None*)  
Send the request.

**Parameters**

- **request** (*requests.models.PreparedRequest*) – The PreparedRequest being sent.
- **stream** (*Any*) – Whether to stream the request content.
- **timeout** (*Any*) – Timeout value of the post request in seconds.
- **verify** (*Any*) – A path string to a CA bundle to use or a boolean which controls whether to verify the server's TLS certificate.
- **cert** (*Any*) – User-provided SSL certificate.
- **proxies** (*Any*) – Proxies dict applying to the request.

**Returns** Response object.

**Return type** *Any*

**class** graviti.utility.requests.**UserSession**

Bases: *requests.Session*

This class defines UserSession.

**request**(*self*, *method*, *url*, *\*args*, *\*\*kwargs*)  
Make the request.

**Parameters**

- **method** (*str*) – Method for the request.
- **url** (*str*) – URL for the request.
- **\*args** – Extra arguments to make the request.
- **\*\*kwargs** – Extra keyword arguments to make the request.
- **args** (*Any*) –
- **kwargs** (*Any*) –

**Returns** Response of the request.

**Raises** *ResponseError* – If post response error.

**Return type** *requests.models.Response*

graviti.utility.requests.**SESSIONS** :*DefaultDict*[*int*, *UserSession*]

graviti.utility.requests.**get\_session**()

Create and return a session per PID so each sub-processes will use their own session.

**Returns** The session corresponding to the process.

**Return type** *UserSession*

**class** graviti.utility.requests.**UserResponse**(*response*)

This class used to read data from Response with stream method.

**Parameters** **response** (*requests.models.Response*) – Response of the Session.request().

**close**(*self*)

Close the response.

**Return type** None

**read**(*self*, *amt=None*)

Read data from response.

**Parameters** *amt* (*Optional[int]*) – The needed read amount.

**Returns** Response of the request.

**Return type** bytes

`graviti.utility.requests.submit_multithread_tasks`(*function*, *arguments*, \*, *jobs*)

Multi-thread framework.

**Parameters**

- **function** (*Callable[[\_T], Any]*) – The function to call.
- **arguments** (*Iterable[\_T]*) – The arguments of the function.
- **jobs** (*int*) – The number of the max workers in multi-thread call procession.

**Return type** None

## graviti.utility.typing

Graviti customized types.

## Module Contents

### Classes

---

*NestedDict*

Typehint for nested dict.

---

### Functions

---

*check\_type*(name, value, expected\_type)

Check the type of the argument.

---

### Attributes

---

*PathLike*

---

`graviti.utility.typing.PathLike`

**class** `graviti.utility.typing.NestedDict`

Bases: `typing_extensions.Protocol[_K, _V]`

Typehint for nested dict.

**items**(*self*)

Return (key, value) pairs of the dict.

**Return type** AbstractSet[Tuple[\_K, Union[NestedDict[\_K, \_V], \_V]]]**setdefault**(*self*, *key*, *default*)

Get the value of the key if exists, else set the value as default and return.

**Parameters**

- **key** (\_K) – The key.
- **default** (Union[NestedDict[\_K, \_V], \_V]) – The default value to set if the key does not exist.

**Return type** Union[NestedDict[\_K, \_V], \_V]**graviti.utility.typing.check\_type**(*name*, *value*, *expected\_type*)

Check the type of the argument.

**Parameters**

- **name** (str) – The name of the argument.
- **value** (Any) – The value of the argument.
- **expected\_type** (Type[Any]) – The type of the argument.

**Raises** **TypeError** – When the value is not of the type.**Return type** None

## 1.7.2 Submodules

### graviti.exception

Basic concepts of Graviti custom exceptions.

### Module Contents

#### Classes

---

*ResponseErrorRegister*A class decorator to register the ResponseError into the distributor.

---

**exception** graviti.exception.**GravitiException**(*message=None*)

Bases: Exception

This is the base class for Graviti custom exceptions.

**Parameters** **message** (Optional[str]) – The error message.**exception** graviti.exception.**UtilityError**(*message=None*)Bases: *GravitiException*

This is the base class for custom exceptions in Graviti utility module.

**Parameters** **message** (Optional[str]) –

**exception** `graviti.exception.ImageDecodeError(message=None)`

Bases: [UtilityError](#)

This class defines the exception for the image decode errors.

**Parameters** `message` (*Optional[str]*) –

**exception** `graviti.exception.PortexError(message=None)`

Bases: [GravitiException](#)

This is the base class for custom exceptions in Graviti portex module.

**Parameters** `message` (*Optional[str]*) –

**exception** `graviti.exception.FieldNameConflictError(message=None)`

Bases: [PortexError](#)

This class defines the exception for the portex field name error.

**Parameters** `message` (*Optional[str]*) –

**exception** `graviti.exception.GitNotFoundError(message=_MESSAGE)`

Bases: [PortexError](#)

This class defines the exception for the git command not found error.

**Parameters** `message` (*str*) – The error message.

**exception** `graviti.exception.GitCommandError(message, called_process_error)`

Bases: [PortexError](#)

This class defines the exception for the git command related error.

**Parameters**

- **message** (*str*) – The error message.
- **called\_process\_error** (*subprocess.CalledProcessError*) – The `CalledProcessError` raised from the `subprocess.run()`.

**exception** `graviti.exception.ManagerError(message=None)`

Bases: [GravitiException](#)

This is the base class for custom exceptions in Graviti manager module.

**Parameters** `message` (*Optional[str]*) –

**exception** `graviti.exception.StatusError(message=None)`

Bases: [ManagerError](#)

This class defines the exception for illegal status.

**Parameters** `message` (*Optional[str]*) –

**exception** `graviti.exception.NoCommitsError(message=None)`

Bases: [StatusError](#)

This class defines the exception for illegal operations on dataset with no commit history.

**Parameters** `message` (*Optional[str]*) –

**exception** `graviti.exception.ResourceNameError(resource, name)`

Bases: [ManagerError](#)

This class defines the exception for invalid resource names.

**Parameters**

- **resource** (*str*) –
- **name** (*str*) –

**exception** graviti.exception.**ResponseError**(*message=None, \*, response=None*)

Bases: [ManagerError](#)

This class defines the exception for post response error.

#### Parameters

- **response** (*Optional[requests.models.Response]*) – The response of the request.
- **message** (*Optional[str]*) –

#### response

The response of the request.

**class** graviti.exception.**ResponseErrorRegister**(*status\_code, error\_code=None*)

A class decorator to register the ResponseError into the distributor.

#### Parameters

- **status\_code** (*int*) – The http status code of the specific ResponseError.
- **error\_code** (*Optional[str]*) – The response error code of the specific ResponseError.

**exception** graviti.exception.**AccessDeniedError**(*message=None, \*, response=None*)

Bases: [ResponseError](#)

This class defines the exception for access denied response error.

#### Parameters

- **message** (*Optional[str]*) –
- **response** (*Optional[requests.models.Response]*) –

**exception** graviti.exception.**ForbiddenError**(*message=None, \*, response=None*)

Bases: [ResponseError](#)

This class defines the exception for illegal operations Graviti forbids.

#### Parameters

- **message** (*Optional[str]*) –
- **response** (*Optional[requests.models.Response]*) –

**exception** graviti.exception.**InvalidParamsError**(*message=None, \*, response=None*)

Bases: [ResponseError](#)

This class defines the exception for invalid parameters response error.

#### Parameters

- **message** (*Optional[str]*) –
- **response** (*Optional[requests.models.Response]*) –

**exception** graviti.exception.**NameConflictError**(*message=None, \*, response=None*)

Bases: [ResponseError](#)

This class defines the exception for name conflict response error.

#### Parameters

- **message** (*Optional[str]*) –



- **response** (*Optional[requests.models.Response]*) –

**exception** graviti.exception.**RequestParamsMissingError**(*message=None, \*, response=None*)

Bases: [ResponseError](#)

This class defines the exception for request parameters missing response error.

#### Parameters

- **message** (*Optional[str]*) –
- **response** (*Optional[requests.models.Response]*) –

**exception** graviti.exception.**NotFoundError**(*message=None, \*, response=None*)

Bases: [ResponseError](#)

This class defines the exception for 404 not found response error without error code.

#### Parameters

- **message** (*Optional[str]*) –
- **response** (*Optional[requests.models.Response]*) –

**exception** graviti.exception.**ResourceNotExistError**(*message=None, \*, response=None*)

Bases: [NotFoundError](#)

This class defines the exception for resource not existing response error.

#### Parameters

- **message** (*Optional[str]*) –
- **response** (*Optional[requests.models.Response]*) –

**exception** graviti.exception.**InternalServerError**(*message=None, \*, response=None*)

Bases: [ResponseError](#)

This class defines the exception for internal server error.

#### Parameters

- **message** (*Optional[str]*) –
- **response** (*Optional[requests.models.Response]*) –

**exception** graviti.exception.**UnauthorizedError**(*message=None, \*, response=None*)

Bases: [ResponseError](#)

This class defines the exception for unauthorized response error.

#### Parameters

- **message** (*Optional[str]*) –
- **response** (*Optional[requests.models.Response]*) –

**exception** graviti.exception.**ServiceUnavailableError**(*message=None, \*, response=None*)

Bases: [ResponseError](#)

This class defines the exception for 503 service unavailable error without error code.

#### Parameters

- **message** (*Optional[str]*) –
- **response** (*Optional[requests.models.Response]*) –

### graviti.workspace

The implementation of the Workspace.

## Module Contents

### Classes

---

<i>Workspace</i>	This class defines the initial client to interact between local and server.
------------------	---

---

**class** graviti.workspace.**Workspace**(*access\_key*, *url=""*)

This class defines the initial client to interact between local and server.

#### Parameters

- **access\_key** (*str*) – User's access key.
- **url** (*str*) – The URL of the graviti website.

**property** **access\_key**(*self*)

Return the access key of the user.

**Returns** The access key of the user.

**Return type** str

**property** **url**(*self*)

Return the url of the graviti website.

**Returns** The url of the graviti website.

**Return type** str

**property** **datasets**(*self*)

Get class *DatasetManager* instance.

**Returns** Required *DatasetManager* instance.

**Return type** graviti.manager.DatasetManager

## PYTHON MODULE INDEX

### g

- graviti, 39
- graviti.dataframe, 39
  - graviti.dataframe.column, 39
  - graviti.dataframe.column.indexing, 39
  - graviti.dataframe.column.series, 40
  - graviti.dataframe.container, 51
  - graviti.dataframe.frame, 52
  - graviti.dataframe.indexing, 58
  - graviti.dataframe.row, 43
  - graviti.dataframe.row.indexing, 43
  - graviti.dataframe.row.series, 43
  - graviti.dataframe.sql, 45
    - graviti.dataframe.sql.array, 45
    - graviti.dataframe.sql.container, 48
    - graviti.dataframe.sql.scalar, 49
- graviti.exception, 194
- graviti.file, 59
  - graviti.file.audio, 59
  - graviti.file.base, 59
  - graviti.file.image, 61
  - graviti.file.image\_size, 63
  - graviti.file.point\_cloud, 65
- graviti.manager, 65
  - graviti.manager.branch, 65
  - graviti.manager.commit, 67
  - graviti.manager.common, 69
  - graviti.manager.dataset, 71
  - graviti.manager.draft, 74
  - graviti.manager.lazy, 77
  - graviti.manager.policy, 81
  - graviti.manager.sheets, 84
  - graviti.manager.tag, 84
- graviti.openapi, 86
  - graviti.openapi.branch, 86
  - graviti.openapi.commit, 89
  - graviti.openapi.data, 92
  - graviti.openapi.dataset, 100
  - graviti.openapi.draft, 104
  - graviti.openapi.policy, 108
  - graviti.openapi.requests, 112
  - graviti.openapi.schema, 113
  - graviti.openapi.search, 114
  - graviti.openapi.sheet, 116
  - graviti.openapi.tag, 122
  - graviti.openapi.user, 125
- graviti.operation, 126
  - graviti.operation.common, 126
  - graviti.operation.frame, 126
  - graviti.operation.sheet, 130
- graviti.paging, 131
  - graviti.paging.factory, 131
  - graviti.paging.lists, 136
  - graviti.paging.offset, 140
  - graviti.paging.page, 141
  - graviti.paging.wrapper, 147
- graviti.portex, 150
  - graviti.portex.avro, 150
  - graviti.portex.base, 151
  - graviti.portex.builder, 154
  - graviti.portex.builtin, 156
  - graviti.portex.enum, 161
  - graviti.portex.external, 163
  - graviti.portex.factory, 164
  - graviti.portex.field, 168
  - graviti.portex.package, 171
  - graviti.portex.param, 173
  - graviti.portex.ptype, 176
  - graviti.portex.register, 179
- graviti.utility, 180
  - graviti.utility.attr, 180
  - graviti.utility.collections, 181
  - graviti.utility.common, 184
  - graviti.utility.engine, 186
  - graviti.utility.itertools, 187
  - graviti.utility.log, 187
  - graviti.utility.repr, 190
  - graviti.utility.requests, 190
  - graviti.utility.typing, 193
- graviti.workspace, 198



## A

`access_key()` (*graviti.workspace.Workspace* property), 198

`AccessDeniedError`, 196

`add()` (*graviti.portex.param.Params* method), 175

`add_data()` (in module *graviti.openapi.data*), 97

`add_subpackage()` (*graviti.portex.package.Imports* method), 173

`AddData` (class in *graviti.operation.frame*), 128

`alias` (*graviti.manager.dataset.Dataset* attribute), 72

`all()` (*graviti.dataframe.sql.array.ScalarArray* method), 46

`ALL_BRANCHES` (in module *graviti.manager.common*), 70

`allowed_retry_methods` (*graviti.utility.requests.Config* attribute), 191

`allowed_retry_status` (*graviti.utility.requests.Config* attribute), 191

`Any` (class in *graviti.portex.ptype*), 177

`any()` (*graviti.dataframe.sql.array.ScalarArray* method), 46

`append()` (*graviti.manager.lazy.LazyPagingList* method), 80

`append()` (*graviti.utility.collections.UserMutableSequence* method), 182

`apply()` (*graviti.dataframe.frame.DataFrame* method), 58

`APPLY_KEY` (in module *graviti.dataframe.frame*), 52

`array` (class in *graviti.portex.builtin*), 161

`Array` (class in *graviti.portex.ptype*), 177

`ArrayDistributor` (class in *graviti.dataframe.sql.array*), 47

`ArraySeries` (class in *graviti.dataframe.column.series*), 42

`ArraySeries` (class in *graviti.dataframe.sql.array*), 48

`ArrayWrapper` (class in *graviti.paging.wrapper*), 148

`as_py()` (*graviti.paging.wrapper.ScalarWrapper* method), 148

`astype()` (*graviti.portex.base.PortexRecordBase* method), 153

`astype()` (*graviti.portex.field.ConnectedFields* method), 171

`astype()` (*graviti.portex.field.Fields* method), 169

`astype()` (*graviti.portex.field.FrozenFields* method), 168

`AttrDict` (class in *graviti.utility.attr*), 180

`Audio` (class in *graviti.file.audio*), 59

`AvroArraySchema` (class in *graviti.portex.avro*), 151

`AvroEnumSchema` (class in *graviti.portex.avro*), 151

`AvroField` (class in *graviti.portex.avro*), 150

`AvroPrimitiveSchema` (class in *graviti.portex.avro*), 150

`AvroRecordSchema` (class in *graviti.portex.avro*), 150

`AvroSchema` (class in *graviti.portex.avro*), 150

`AZUREObjectPolicyManager` (class in *graviti.manager.policy*), 82

## B

`backend_type` (*graviti.manager.dataset.Dataset* attribute), 72

`binary` (class in *graviti.portex.builtin*), 157

`BMP` (class in *graviti.file.image\_size*), 64

`boolean` (class in *graviti.portex.builtin*), 158

`Boolean` (class in *graviti.portex.ptype*), 177

`BooleanScalar` (class in *graviti.dataframe.sql.scalar*), 50

`BooleanScalarArray` (class in *graviti.dataframe.sql.array*), 46

`Branch` (class in *graviti.manager.branch*), 65

`branches()` (*graviti.manager.dataset.Dataset* property), 72

`BranchManager` (class in *graviti.manager.branch*), 66

`build()` (*graviti.portex.builder.PackageBuilder* method), 155

`build()` (*graviti.portex.builder.TypeBuilder* method), 155

`build_package()` (in module *graviti.portex.builder*), 156

`BuilderImports` (class in *graviti.portex.builder*), 155

`BuiltinPackage` (class in *graviti.portex.package*), 171

`builtins` (in module *graviti.portex.builtin*), 157

## C

`check()` (*graviti.file.image\_size.ImageFormatBase* class method), 63

`check()` (*graviti.portex.param.Param* method), 174

- `check()` (*graviti.portex.ptype.Enum static method*), 177  
`check()` (*graviti.portex.ptype.Fields static method*), 178  
`check()` (*graviti.portex.ptype.ParameterType static method*), 176  
`check()` (*graviti.portex.ptype.PortexType static method*), 178  
`check_head_status()` (*in module graviti.manager.common*), 70  
`check_type()` (*in module graviti.utility.typing*), 194  
`check_type_for_other()` (*graviti.dataframe.sql.scalar.ComparisonArithmeticOperatorsMixin method*), 50  
`checkout()` (*graviti.manager.dataset.Dataset method*), 73  
`chunked()` (*in module graviti.utility.iterutils*), 187  
`clear()` (*graviti.utility.collections.UserMutableMapping method*), 183  
`clear()` (*graviti.utility.collections.UserMutableSequence method*), 182  
`close()` (*graviti.manager.draft.Draft method*), 75  
`close()` (*graviti.utility.requests.UserResponse method*), 192  
`ColumnSeriesILocIndexer` (*class in graviti.dataframe.column.indexing*), 39  
`ColumnSeriesLocIndexer` (*class in graviti.dataframe.column.indexing*), 39  
`Commit` (*class in graviti.manager.commit*), 67  
`commit()` (*graviti.manager.dataset.Dataset method*), 73  
`commit()` (*graviti.manager.draft.Draft method*), 75  
`commit_draft()` (*in module graviti.openapi.commit*), 89  
`commit_id` (*graviti.manager.dataset.Dataset attribute*), 72  
`CommitManager` (*class in graviti.manager.commit*), 69  
`commits()` (*graviti.manager.dataset.Dataset property*), 73  
`committed_at()` (*graviti.manager.commit.Commit property*), 68  
`committer()` (*graviti.manager.commit.Commit property*), 68  
`ComparisonArithmeticOperatorsMixin` (*class in graviti.dataframe.sql.scalar*), 50  
`Config` (*class in graviti.utility.requests*), 191  
`config` (*graviti.manager.dataset.Dataset attribute*), 72  
`config` (*in module graviti.utility.requests*), 191  
`ConnectedFields` (*class in graviti.portex.field*), 170  
`ConnectedFieldsFactory` (*class in graviti.portex.factory*), 165  
`ConstantFactory` (*class in graviti.portex.factory*), 166  
`Container` (*class in graviti.dataframe.container*), 51  
`ContainerRegister` (*class in graviti.portex.register*), 179  
`convert_datetime_to_gmt()` (*in module graviti.utility.common*), 185  
`convert_iso_to_datetime()` (*in module graviti.utility.common*), 185  
`convert_portex_schema_to_avro()` (*in module graviti.portex.avro*), 151  
`copy()` (*graviti.dataframe.container.Container method*), 51  
`copy()` (*graviti.paging.lists.MappedPagingList method*), 138  
`copy()` (*graviti.paging.lists.PagingListBase method*), 137  
`copy()` (*graviti.paging.lists.PyArrowPagingList method*), 140  
`copy()` (*graviti.paging.offset.Offsets method*), 141  
`copy()` (*graviti.paging.page.MappedLazyPage method*), 146  
`copy()` (*graviti.paging.page.MappedLazySlicedPage method*), 147  
`copy()` (*graviti.paging.page.MappedPage method*), 145  
`copy()` (*graviti.paging.page.MappedPageBase method*), 144  
`copy()` (*graviti.paging.page.MappedSlicedPage method*), 145  
`copy()` (*graviti.portex.base.PortexType method*), 153  
`copy()` (*graviti.portex.field.Fields method*), 170  
`count()` (*graviti.manager.lazy.LazyPagingList method*), 81  
`count()` (*graviti.utility.collections.UserSequence method*), 181  
`create()` (*graviti.manager.branch.BranchManager method*), 66  
`create()` (*graviti.manager.dataset.DatasetManager method*), 74  
`create()` (*graviti.manager.draft.DraftManager method*), 77  
`create()` (*graviti.manager.tag.TagManager method*), 85  
`create_branch()` (*in module graviti.openapi.branch*), 86  
`create_dataset()` (*in module graviti.openapi.dataset*), 100  
`create_draft()` (*in module graviti.openapi.draft*), 104  
`create_enum_values()` (*in module graviti.portex.enum*), 162  
`create_list()` (*graviti.paging.factory.LazyFactory method*), 134  
`create_list()` (*graviti.paging.factory.LazyFactoryBase method*), 132  
`create_list()` (*graviti.paging.factory.LazySubFactory method*), 134  
`create_mapped_list()` (*graviti.paging.factory.LazyFactory method*), 134  
`create_mapped_list()` (*graviti.paging.factory.LazyFactoryBase method*),

- 132  
 create\_mapped\_list() (grav-  
 iti.paging.factory.LazySubFactory method),  
 135  
 create\_pyarrow\_list() (grav-  
 iti.paging.factory.LazyFactory method),  
 134  
 create\_pyarrow\_list() (grav-  
 iti.paging.factory.LazyFactoryBase method),  
 132  
 create\_pyarrow\_list() (grav-  
 iti.paging.factory.LazySubFactory method),  
 135  
 create\_search() (in module graviti.openapi.search),  
 114  
 create\_sheet() (in module graviti.openapi.sheet), 117  
 create\_tag() (in module graviti.openapi.tag), 122  
 created\_at (graviti.manager.dataset.Dataset attribute),  
 72  
 CreateSheet (class in graviti.operation.sheet), 130  
 CURRENT\_BRANCH (in module graviti.manager.common),  
 70  
 CURRENT\_COMMIT (in module graviti.manager.common),  
 70
- ## D
- data (graviti.manager.lazy.LazyItem attribute), 78  
 DataFrame (class in graviti.dataframe.frame), 52  
 DataFrame (class in graviti.dataframe.sql.array), 47  
 DataFrameIlocIndexer (class in grav-  
 iti.dataframe.indexing), 58  
 DataFrameLocIndexer (class in grav-  
 iti.dataframe.indexing), 58  
 DataFrameOperation (class in graviti.operation.frame),  
 127  
 DataOperation (class in graviti.operation.frame), 127  
 Dataset (class in graviti.manager.dataset), 71  
 dataset\_id (graviti.manager.dataset.Dataset attribute),  
 72  
 DatasetManager (class in graviti.manager.dataset), 73  
 datasets() (graviti.workspace.Workspace property),  
 198  
 default\_branch (graviti.manager.dataset.Dataset at-  
 tribute), 72  
 DefaultValue (class in graviti.manager.common), 70  
 delete() (graviti.manager.branch.BranchManager  
 method), 66  
 delete() (graviti.manager.dataset.DatasetManager  
 method), 74  
 delete() (graviti.manager.tag.TagManager method), 85  
 delete\_branch() (in module graviti.openapi.branch),  
 88  
 delete\_data() (in module graviti.openapi.data), 98  
 delete\_dataset() (in module graviti.openapi.dataset),  
 104  
 delete\_sheet() (in module graviti.openapi.sheet), 121  
 delete\_tag() (in module graviti.openapi.tag), 124  
 DeleteData (class in graviti.operation.frame), 129  
 DeleteSheet (class in graviti.operation.sheet), 131  
 description() (graviti.manager.commit.Commit prop-  
 erty), 68  
 DictFactory (class in graviti.portex.factory), 166  
 do() (graviti.operation.frame.AddData method), 128  
 do() (graviti.operation.frame.DataFrameOperation  
 method), 127  
 do() (graviti.operation.frame.DeleteData method), 129  
 do() (graviti.operation.frame.UpdateData method), 129  
 do() (graviti.operation.frame.UpdateSchema method),  
 128  
 do() (graviti.operation.sheet.CreateSheet method), 130  
 do() (graviti.operation.sheet.DeleteSheet method), 131  
 do() (graviti.operation.sheet.SheetOperation method),  
 130  
 do() (in module graviti.openapi.requests), 112  
 Draft (class in graviti.manager.draft), 74  
 DraftManager (class in graviti.manager.draft), 76  
 drafts() (graviti.manager.dataset.Dataset property), 73  
 dump() (graviti.portex.param.Param method), 175  
 dump() (graviti.portex.ptype.Enum static method), 178  
 dump() (graviti.portex.ptype.Fields static method), 178  
 dump() (graviti.portex.ptype.ParameterType static  
 method), 176  
 dump() (graviti.portex.ptype.PortexType static method),  
 179  
 dump\_request\_and\_response() (in module grav-  
 iti.utility.log), 188
- ## E
- edit() (graviti.manager.dataset.Dataset method), 73  
 edit() (graviti.manager.draft.Draft method), 75  
 Engine (class in graviti.utility.engine), 186  
 engine (in module graviti.utility.engine), 186  
 enum (class in graviti.portex.builtin), 160  
 Enum (class in graviti.portex.ptype), 177  
 EnumScalar (class in graviti.dataframe.sql.scalar), 50  
 EnumScalarArray (class in graviti.dataframe.sql.array),  
 46  
 EnumSeries (class in graviti.dataframe.column.series),  
 42  
 EnumValueDict (class in graviti.portex.enum), 162  
 EnumValueList (class in graviti.portex.enum), 162  
 EnumValues (class in graviti.portex.enum), 162  
 EnumValueType (in module graviti.portex.enum), 162  
 extend() (graviti.dataframe.frame.DataFrame method),  
 56  
 extend() (graviti.manager.lazy.LazyPagingList  
 method), 81



- extend() (graviti.paging.lists.PagingListBase method), 137  
 extend() (graviti.paging.lists.PyArrowPagingList method), 140  
 extend() (graviti.paging.offset.Offsets method), 141  
 extend() (graviti.utility.collections.UserMutableSequence method), 182  
 extend\_iterable() (graviti.paging.lists.PagingListBase method), 137  
 extend\_nulls() (graviti.paging.lists.PagingListBase method), 137  
 extend\_nulls() (graviti.paging.lists.PyArrowPagingList method), 140  
 extension() (graviti.file.base.File property), 60  
 extension() (graviti.file.base.FileBase property), 60  
 EXTERNAL\_TYPE\_TO\_CONTAINER (in module graviti.portex.builder), 155  
 EXTERNAL\_TYPE\_TO\_CONTAINER (in module graviti.portex.external), 163  
 EXTERNAL\_TYPE\_TO\_ELEMENT (in module graviti.portex.external), 163  
 ExternalContainerRegister (class in graviti.portex.register), 179  
 ExternalElementResgister (class in graviti.portex.register), 180  
 ExternalPackage (class in graviti.portex.package), 172
- ## F
- Factory (class in graviti.portex.factory), 165  
 factory\_creator() (in module graviti.portex.factory), 167  
 field() (graviti.paging.wrapper.StructArrayWrapper method), 148  
 FieldFactory (class in graviti.portex.factory), 166  
 FileNameConflictError, 195  
 Fields (class in graviti.portex.field), 169  
 Fields (class in graviti.portex.ptype), 178  
 FieldsFactory (class in graviti.portex.factory), 167  
 File (class in graviti.file.base), 60  
 FileBase (class in graviti.file.base), 60  
 FileSeries (class in graviti.dataframe.column.series), 42  
 FLIF (class in graviti.file.image\_size), 64  
 float32 (class in graviti.portex.builtin), 159  
 float64 (class in graviti.portex.builtin), 159  
 ForbiddenError, 196  
 from\_array() (graviti.paging.lists.MappedPagingList class method), 138  
 from\_data() (graviti.manager.lazy.LazyItem class method), 78  
 from\_factory() (graviti.paging.lists.MappedPagingList method), 138  
 from\_factory() (graviti.paging.lists.PagingList class method), 137  
 from\_factory() (graviti.paging.lists.PyArrowPagingList class method), 139  
 from\_items() (graviti.manager.lazy.LazyPage class method), 79  
 from\_json() (graviti.portex.base.PortexType class method), 152  
 from\_page() (graviti.manager.lazy.LazyItem class method), 78  
 from\_parameter\_name() (graviti.portex.factory.ConnectedFieldsFactory class method), 165  
 from\_pyarrow() (graviti.dataframe.column.series.SeriesBase class method), 40  
 from\_pyarrow() (graviti.dataframe.frame.DataFrame class method), 53  
 from\_pyarrow() (graviti.paging.lists.PyArrowPagingList class method), 139  
 from\_pyarrow() (graviti.portex.base.PortexType class method), 152  
 from\_pyobj() (graviti.portex.base.PortexType class method), 152  
 from\_pyobj() (graviti.portex.builder.BuilderImports class method), 156  
 from\_pyobj() (graviti.portex.field.Fields class method), 170  
 from\_pyobj() (graviti.portex.package.Imports class method), 173  
 from\_pyobj() (graviti.portex.package.Subpackage class method), 172  
 from\_pyobj() (graviti.portex.param.Param class method), 174  
 from\_pyobj() (graviti.portex.param.Params class method), 175  
 from\_response() (graviti.manager.commit.Commit class method), 67  
 from\_response() (graviti.manager.commit.NamedCommit class method), 68  
 from\_upper() (graviti.dataframe.sql.array.ArrayDistributor class method), 48  
 from\_upper() (graviti.dataframe.sql.container.SearchContainer class method), 49  
 from\_upper() (graviti.dataframe.sql.container.SearchScalarContainer class method), 48  
 from\_wrapper() (graviti.paging.wrapper.ListArrayWrapper class method), 149  
 from\_wrapper() (graviti.paging.wrapper.StructArrayWrapper class method), 148



*iti.paging.wrapper.ListScalarWrapper* class method), 149  
*from\_wrapper()* (*graviti.paging.wrapper.StructScalarWrapper* class method), 148  
*from\_yaml()* (*graviti.portex.base.PortexType* class method), 152  
*FrozenFields* (class in *graviti.portex.field*), 168  
*FrozenFieldsFactory* (class in *graviti.portex.factory*), 165  
*FrozenFieldsFactoryWrapper* (class in *graviti.portex.factory*), 165  
*FrozenNameOrderedDict* (class in *graviti.utility.collections*), 184

## G

*get()* (*graviti.manager.branch.BranchManager* method), 66  
*get()* (*graviti.manager.commit.CommitManager* method), 69  
*get()* (*graviti.manager.dataset.DatasetManager* method), 74  
*get()* (*graviti.manager.draft.DraftManager* method), 77  
*get()* (*graviti.manager.lazy.LazyItem* method), 78  
*get()* (*graviti.manager.tag.TagManager* method), 85  
*get()* (*graviti.paging.wrapper.WrapperRegister* class method), 148  
*get()* (*graviti.utility.attr.AttrDict* method), 180  
*get()* (*graviti.utility.collections.UserMapping* method), 182  
*get\_array()* (*graviti.paging.factory.LazyFactory* method), 133  
*get\_array()* (*graviti.paging.factory.LazyLowerCaseFactory* method), 135  
*get\_array()* (*graviti.paging.page.LazyPage* method), 144  
*get\_array()* (*graviti.paging.page.LazySlicedPage* method), 144  
*get\_array()* (*graviti.paging.page.MappedLazyPage* method), 146  
*get\_array()* (*graviti.paging.page.MappedLazySlicedPage* method), 147  
*get\_array()* (*graviti.paging.page.Page* method), 143  
*get\_array()* (*graviti.paging.page.PageBase* method), 142  
*get\_array()* (*graviti.paging.page.SlicedPage* method), 143  
*get\_branch()* (in module *graviti.openapi.branch*), 88  
*get\_checksum()* (*graviti.file.base.File* method), 61  
*get\_commit()* (in module *graviti.openapi.commit*), 91  
*get\_commit\_sheet()* (in module *graviti.openapi.sheet*), 120  
*get\_coordinate()* (*graviti.paging.offset.Offsets* method), 141  
*get\_current\_user()* (in module *graviti.openapi.user*), 125  
*get\_data\_count()* (*graviti.operation.frame.DataFrameOperation* method), 127  
*get\_data\_count()* (*graviti.operation.frame.DataOperation* method), 128  
*get\_dataset()* (in module *graviti.openapi.dataset*), 101  
*get\_draft()* (in module *graviti.openapi.draft*), 106  
*get\_draft\_sheet()* (in module *graviti.openapi.sheet*), 119  
*get\_file\_count()* (*graviti.operation.frame.DataFrameOperation* method), 127  
*get\_file\_count()* (*graviti.operation.frame.DataOperation* method), 127  
*get\_image\_size()* (*graviti.file.image\_size.ImageFormatBase* class method), 64  
*get\_image\_size()* (in module *graviti.file.image\_size*), 63  
*get\_item()* (*graviti.paging.lists.PagingListBase* method), 136  
*get\_item()* (*graviti.paging.page.PageBase* method), 142  
*get\_object()* (*graviti.manager.policy.AZUREObjectPolicyManager* method), 83  
*get\_object()* (*graviti.manager.policy.ObjectPolicyManager* method), 81  
*get\_object()* (*graviti.manager.policy.OSSObjectPolicyManager* method), 82  
*get\_object()* (*graviti.manager.policy.S3ObjectPolicyManager* method), 83  
*get\_object\_policy()* (in module *graviti.openapi.policy*), 108  
*get\_offsets()* (*graviti.paging.factory.LazyFactory* method), 134  
*get\_page\_lengths()* (*graviti.paging.factory.LazyFactory* method), 134  
*get\_policy()* (in module *graviti.openapi.data*), 99  
*get\_revision()* (in module *graviti.openapi.commit*), 92  
*get\_root()* (*graviti.portex.builder.PackageRepo* method), 155  
*get\_schema()* (in module *graviti.operation.common*), 126  
*get\_session()* (in module *graviti.utility.requests*), 192  
*get\_signature()* (*graviti.portex.param.Params* method), 175  
*get\_slice()* (*graviti.paging.lists.PagingListBase* method), 136

`get_slice()` (*graviti.paging.lists.PyArrowPagingList* method), 139

`get_slice()` (*graviti.paging.page.LazyPage* method), 143

`get_slice()` (*graviti.paging.page.LazySlicedPage* method), 144

`get_slice()` (*graviti.paging.page.MappedLazyPage* method), 146

`get_slice()` (*graviti.paging.page.MappedLazySlicedPage* method), 146

`get_slice()` (*graviti.paging.page.MappedPage* method), 145

`get_slice()` (*graviti.paging.page.MappedSlicedPage* method), 145

`get_slice()` (*graviti.paging.page.Page* method), 142

`get_slice()` (*graviti.paging.page.PageBase* method), 142

`get_slice()` (*graviti.paging.page.SlicedPage* method), 143

`get_tag()` (in module *graviti.openapi.tag*), 124

`GIF` (class in *graviti.file.image\_size*), 64

`GitCommandError`, 195

`GitNotFoundError`, 195

`graviti`  
module, 39

`graviti.dataframe`  
module, 39

`graviti.dataframe.column`  
module, 39

`graviti.dataframe.column.indexing`  
module, 39

`graviti.dataframe.column.series`  
module, 40

`graviti.dataframe.container`  
module, 51

`graviti.dataframe.frame`  
module, 52

`graviti.dataframe.indexing`  
module, 58

`graviti.dataframe.row`  
module, 43

`graviti.dataframe.row.indexing`  
module, 43

`graviti.dataframe.row.series`  
module, 43

`graviti.dataframe.sql`  
module, 45

`graviti.dataframe.sql.array`  
module, 45

`graviti.dataframe.sql.container`  
module, 48

`graviti.dataframe.sql.scalar`  
module, 49

`graviti.exception`  
module, 194

`graviti.file`  
module, 59

`graviti.file.audio`  
module, 59

`graviti.file.base`  
module, 59

`graviti.file.image`  
module, 61

`graviti.file.image_size`  
module, 63

`graviti.file.point_cloud`  
module, 65

`graviti.manager`  
module, 65

`graviti.manager.branch`  
module, 65

`graviti.manager.commit`  
module, 67

`graviti.manager.common`  
module, 69

`graviti.manager.dataset`  
module, 71

`graviti.manager.draft`  
module, 74

`graviti.manager.lazy`  
module, 77

`graviti.manager.policy`  
module, 81

`graviti.manager.sheets`  
module, 84

`graviti.manager.tag`  
module, 84

`graviti.openapi`  
module, 86

`graviti.openapi.branch`  
module, 86

`graviti.openapi.commit`  
module, 89

`graviti.openapi.data`  
module, 92

`graviti.openapi.dataset`  
module, 100

`graviti.openapi.draft`  
module, 104

`graviti.openapi.policy`  
module, 108

`graviti.openapi.requests`  
module, 112

`graviti.openapi.schema`  
module, 113

`graviti.openapi.search`  
module, 114

`graviti.openapi.sheet`

- module, 116
- graviti.openapi.tag
  - module, 122
- graviti.openapi.user
  - module, 125
- graviti.operation
  - module, 126
- graviti.operation.common
  - module, 126
- graviti.operation.frame
  - module, 126
- graviti.operation.sheet
  - module, 130
- graviti.paging
  - module, 131
- graviti.paging.factory
  - module, 131
- graviti.paging.lists
  - module, 136
- graviti.paging.offset
  - module, 140
- graviti.paging.page
  - module, 141
- graviti.paging.wrapper
  - module, 147
- graviti.portex
  - module, 150
- graviti.portex.avro
  - module, 150
- graviti.portex.base
  - module, 151
- graviti.portex.builder
  - module, 154
- graviti.portex.builtin
  - module, 156
- graviti.portex.enum
  - module, 161
- graviti.portex.external
  - module, 163
- graviti.portex.factory
  - module, 164
- graviti.portex.field
  - module, 168
- graviti.portex.package
  - module, 171
- graviti.portex.param
  - module, 173
- graviti.portex.ptype
  - module, 176
- graviti.portex.register
  - module, 179
- graviti.utility
  - module, 180
- graviti.utility.attr

- module, 180
- graviti.utility.collections
  - module, 181
- graviti.utility.common
  - module, 184
- graviti.utility.engine
  - module, 186
- graviti.utility.itertools
  - module, 187
- graviti.utility.log
  - module, 187
- graviti.utility.repr
  - module, 190
- graviti.utility.requests
  - module, 190
- graviti.utility.typing
  - module, 193
- graviti.workspace
  - module, 198
- GravitiException, 194

## H

- handler (*in module graviti.manager.dataset*), 71
- head() (*graviti.dataframe.frame.DataFrame method*), 54
- HEAD() (*graviti.manager.dataset.Dataset property*), 72
- height() (*graviti.file.image.Image property*), 62
- height() (*graviti.file.image.RemoteImage property*), 62

## I

- ICO (*class in graviti.file.image\_size*), 64
- iloc() (*graviti.dataframe.column.series.SeriesBase property*), 40
- iloc() (*graviti.dataframe.container.Container property*), 51
- iloc() (*graviti.dataframe.frame.DataFrame property*), 53
- iloc() (*graviti.dataframe.row.series.Series property*), 44
- Image (*class in graviti.file.image*), 61
- Image (*in module graviti.file.image\_size*), 63
- ImageDecodeError, 194
- ImageFormatBase (*class in graviti.file.image\_size*), 63
- ImageMocker (*class in graviti.utility.common*), 185
- Imports (*class in graviti.portex.package*), 172
- imports() (*graviti.portex.base.PortexType property*), 151
- imports() (*graviti.portex.field.Fields property*), 169
- INDENT (*in module graviti.utility.repr*), 190
- index() (*graviti.manager.lazy.LazyPagingList method*), 80
- index() (*graviti.utility.collections.UserSequence method*), 181
- InitPage (*class in graviti.manager.lazy*), 79
- insert() (*graviti.manager.lazy.LazyPagingList method*), 80

- `insert()` (*graviti.portex.base.PortexRecordBase* method), 153  
`insert()` (*graviti.portex.field.ConnectedFields* method), 170  
`insert()` (*graviti.portex.field.Fields* method), 169  
`insert()` (*graviti.portex.field.FrozenFields* method), 168  
`insert()` (*graviti.utility.collections.UserMutableSequence* method), 182  
`int32` (class in *graviti.portex.builtin*), 158  
`int64` (class in *graviti.portex.builtin*), 158  
`Integer` (class in *graviti.portex.ptype*), 177  
`internal_type()` (*graviti.portex.external.PortexExternalType* property), 163  
`InternalServerError`, 197  
`InvalidParamsError`, 196  
`is_internal` (*graviti.utility.requests.Config* attribute), 191  
`is_public` (*graviti.manager.dataset.Dataset* attribute), 72  
`is_valid()` (*graviti.paging.wrapper.ScalarWrapper* property), 148  
`items` (*graviti.manager.lazy.InitPage* attribute), 80  
`items` (*graviti.manager.lazy.LazyPage* attribute), 79  
`items()` (*graviti.manager.sheets.Sheets* method), 84  
`items()` (*graviti.utility.collections.UserMapping* method), 183  
`items()` (*graviti.utility.typing.NestedDict* method), 193
- ## J
- `JPEG` (class in *graviti.file.image\_size*), 64  
`JPEG2000` (class in *graviti.file.image\_size*), 64
- ## K
- `key()` (*graviti.file.base.File* property), 60  
`key()` (*graviti.file.base.FileBase* property), 60  
`keys()` (*graviti.manager.sheets.Sheets* method), 84  
`keys()` (*graviti.utility.collections.UserMapping* method), 183
- ## L
- `LazyFactory` (class in *graviti.paging.factory*), 132  
`LazyFactoryBase` (class in *graviti.paging.factory*), 132  
`LazyItem` (class in *graviti.manager.lazy*), 78  
`LazyLowerCaseFactory` (class in *graviti.paging.factory*), 135  
`LazyLowerCaseSubFactory` (class in *graviti.paging.factory*), 135  
`LazyPage` (class in *graviti.manager.lazy*), 79  
`LazyPage` (class in *graviti.paging.page*), 143  
`LazyPagingList` (class in *graviti.manager.lazy*), 80  
`LazySlicedPage` (class in *graviti.paging.page*), 144  
`LazySubFactory` (class in *graviti.paging.factory*), 134  
`LIMIT` (in module *graviti.manager.common*), 70  
`list()` (*graviti.manager.branch.BranchManager* method), 66  
`list()` (*graviti.manager.commit.CommitManager* method), 69  
`list()` (*graviti.manager.dataset.DatasetManager* method), 74  
`list()` (*graviti.manager.draft.DraftManager* method), 77  
`list()` (*graviti.manager.tag.TagManager* method), 85  
`list_branches()` (in module *graviti.openapi.branch*), 87  
`list_commit_data()` (in module *graviti.openapi.data*), 95  
`list_commit_sheets()` (in module *graviti.openapi.sheet*), 118  
`list_commits()` (in module *graviti.openapi.commit*), 90  
`list_datasets()` (in module *graviti.openapi.dataset*), 102  
`list_draft_data()` (in module *graviti.openapi.data*), 93  
`list_draft_sheets()` (in module *graviti.openapi.sheet*), 118  
`list_drafts()` (in module *graviti.openapi.draft*), 105  
`list_tags()` (in module *graviti.openapi.tag*), 123  
`ListArrayWrapper` (class in *graviti.paging.wrapper*), 149  
`ListFactory` (class in *graviti.portex.factory*), 166  
`ListScalarWrapper` (class in *graviti.paging.wrapper*), 149  
`load()` (*graviti.portex.param.Param* method), 174  
`load()` (*graviti.portex.ptype.Enum* static method), 177  
`load()` (*graviti.portex.ptype.Fields* static method), 178  
`load()` (*graviti.portex.ptype.ParameterType* static method), 176  
`load()` (*graviti.portex.ptype.PortexType* static method), 178  
`loc()` (*graviti.dataframe.column.series.SeriesBase* property), 41  
`loc()` (*graviti.dataframe.container.Container* property), 51  
`loc()` (*graviti.dataframe.frame.DataFrame* property), 53  
`loc()` (*graviti.dataframe.row.series.Series* property), 44  
`LocalPackage` (class in *graviti.portex.package*), 172  
`locked()` (in module *graviti.utility.common*), 185  
`locks` (in module *graviti.utility.common*), 185  
`logger` (in module *graviti.manager.dataset*), 71  
`logger` (in module *graviti.utility.requests*), 191  
`LogicalOperatorsMixin` (class in *graviti.dataframe.sql.array*), 45  
`LogicalOperatorsMixin` (class in *graviti.dataframe.sql.scalar*), 50

## M

ManagerError, 195

MappedLazyPage (class in *graviti.paging.page*), 146

MappedLazySlicedPage (class in *graviti.paging.page*), 146

MappedPage (class in *graviti.paging.page*), 144

MappedPageBase (class in *graviti.paging.page*), 144

MappedPagingList (class in *graviti.paging.lists*), 138

MappedSlicedPage (class in *graviti.paging.page*), 145

Mapping (class in *graviti.portex.ptype*), 177

mapping\_unpack\_factory\_creator() (in module *graviti.portex.factory*), 167

max() (*graviti.dataframe.sql.array.NumberScalarArray* method), 47

MAX\_REPR\_ROWS (in module *graviti.utility.repr*), 190

max\_retries (*graviti.utility.requests.Config* attribute), 191

min() (*graviti.dataframe.sql.array.NumberScalarArray* method), 47

Mode (class in *graviti.utility.engine*), 186

module

*graviti*, 39

*graviti.dataframe*, 39

*graviti.dataframe.column*, 39

*graviti.dataframe.column.indexing*, 39

*graviti.dataframe.column.series*, 40

*graviti.dataframe.container*, 51

*graviti.dataframe.frame*, 52

*graviti.dataframe.indexing*, 58

*graviti.dataframe.row*, 43

*graviti.dataframe.row.indexing*, 43

*graviti.dataframe.row.series*, 43

*graviti.dataframe.sql*, 45

*graviti.dataframe.sql.array*, 45

*graviti.dataframe.sql.container*, 48

*graviti.dataframe.sql.scalar*, 49

*graviti.exception*, 194

*graviti.file*, 59

*graviti.file.audio*, 59

*graviti.file.base*, 59

*graviti.file.image*, 61

*graviti.file.image\_size*, 63

*graviti.file.point\_cloud*, 65

*graviti.manager*, 65

*graviti.manager.branch*, 65

*graviti.manager.commit*, 67

*graviti.manager.common*, 69

*graviti.manager.dataset*, 71

*graviti.manager.draft*, 74

*graviti.manager.lazy*, 77

*graviti.manager.policy*, 81

*graviti.manager.sheets*, 84

*graviti.manager.tag*, 84

*graviti.openapi*, 86

*graviti.openapi.branch*, 86

*graviti.openapi.commit*, 89

*graviti.openapi.data*, 92

*graviti.openapi.dataset*, 100

*graviti.openapi.draft*, 104

*graviti.openapi.policy*, 108

*graviti.openapi.requests*, 112

*graviti.openapi.schema*, 113

*graviti.openapi.search*, 114

*graviti.openapi.sheet*, 116

*graviti.openapi.tag*, 122

*graviti.openapi.user*, 125

*graviti.operation*, 126

*graviti.operation.common*, 126

*graviti.operation.frame*, 126

*graviti.operation.sheet*, 130

*graviti.paging*, 131

*graviti.paging.factory*, 131

*graviti.paging.lists*, 136

*graviti.paging.offset*, 140

*graviti.paging.page*, 141

*graviti.paging.wrapper*, 147

*graviti.portex*, 150

*graviti.portex.avro*, 150

*graviti.portex.base*, 151

*graviti.portex.builder*, 154

*graviti.portex.builtin*, 156

*graviti.portex.enum*, 161

*graviti.portex.external*, 163

*graviti.portex.factory*, 164

*graviti.portex.field*, 168

*graviti.portex.package*, 171

*graviti.portex.param*, 173

*graviti.portex.ptype*, 176

*graviti.portex.register*, 179

*graviti.utility*, 180

*graviti.utility.attr*, 180

*graviti.utility.collections*, 181

*graviti.utility.common*, 184

*graviti.utility.engine*, 186

*graviti.utility.itertools*, 187

*graviti.utility.log*, 187

*graviti.utility.repr*, 190

*graviti.utility.requests*, 190

*graviti.utility.typing*, 193

*graviti.workspace*, 198

## N

name (*graviti.manager.dataset.Dataset* attribute), 72

NameConflictError, 196

NamedCommit (class in *graviti.manager.commit*), 68

NameOrderedDict (class in *graviti.utility.collections*), 184

NestedDict (class in *graviti.utility.typing*), 193



- NoCommitsError, 195
- NotFoundError, 197
- Number (class in *graviti.portex.ptype*), 177
- NumberScalar (class in *graviti.dataframe.sql.scalar*), 50
- NumberScalarArray (class in *graviti.dataframe.sql.array*), 46
- NUMERICAL\_PRIORITY (in module *graviti.dataframe.sql.scalar*), 50
- ## O
- ObjectPolicyManager (class in *graviti.manager.policy*), 81
- ObjectPolicyManagerType (class in *graviti.manager.dataset*), 71
- Offsets (class in *graviti.paging.offset*), 140
- OldPNG (class in *graviti.file.image\_size*), 64
- Online (class in *graviti.utility.engine*), 186
- online() (*graviti.utility.engine.Engine* method), 186
- open() (*graviti.file.base.File* method), 61
- open() (*graviti.file.base.FileBase* method), 60
- open() (*graviti.file.base.RemoteFile* method), 61
- open\_api\_do() (in module *graviti.openapi.requests*), 112
- OSSObjectPolicyManager (class in *graviti.manager.policy*), 82
- owner (*graviti.manager.dataset.Dataset* attribute), 72
- ## P
- Package (class in *graviti.portex.package*), 171
- PackageBuilder (class in *graviti.portex.builder*), 155
- PackageRepo (class in *graviti.portex.builder*), 155
- Packages (class in *graviti.portex.package*), 172
- packages (in module *graviti.portex.package*), 172
- Page (class in *graviti.paging.page*), 142
- page (*graviti.manager.lazy.LazyItem* attribute), 78
- PageBase (class in *graviti.paging.page*), 142
- PagingGenerator (in module *graviti.manager.lazy*), 78
- PagingList (class in *graviti.paging.lists*), 137
- PagingListBase (class in *graviti.paging.lists*), 136
- Param (class in *graviti.portex.param*), 174
- param() (in module *graviti.portex.param*), 173
- ParameterType (class in *graviti.portex.ptype*), 176
- Params (class in *graviti.portex.param*), 175
- parent (*graviti.manager.draft.Draft* attribute), 75
- parent() (*graviti.manager.commit.Commit* property), 67
- path() (*graviti.file.base.File* property), 60
- PathLike (in module *graviti.utility.typing*), 193
- PNG (class in *graviti.file.image\_size*), 64
- PointCloud (class in *graviti.file.point\_cloud*), 65
- pop() (*graviti.manager.lazy.LazyPagingList* method), 80
- pop() (*graviti.utility.collections.UserMutableMapping* method), 183
- pop() (*graviti.utility.collections.UserMutableSequence* method), 182
- popitem() (*graviti.utility.collections.NameOrderedDict* method), 184
- popitem() (*graviti.utility.collections.UserMutableMapping* method), 183
- PortexBuiltinType (class in *graviti.portex.builtin*), 157
- PortexError, 195
- PortexExternalType (class in *graviti.portex.external*), 163
- PortexRecordBase (class in *graviti.portex.base*), 153
- PortexType (class in *graviti.portex.base*), 151
- PortexType (class in *graviti.portex.ptype*), 178
- prefix() (*graviti.manager.policy.ObjectPolicyManager* property), 81
- PType (in module *graviti.portex.ptype*), 177
- pull() (*graviti.manager.lazy.LazyPage* method), 79
- put\_object() (*graviti.manager.policy.AZUREObjectPolicyManager* method), 83
- put\_object() (*graviti.manager.policy.ObjectPolicyManager* method), 82
- put\_object() (*graviti.manager.policy.OSSObjectPolicyManager* method), 82
- put\_object() (*graviti.manager.policy.S3ObjectPolicyManager* method), 83
- PYARROW\_TYPE\_ID\_TO\_PORTEX\_TYPE (in module *graviti.portex.base*), 151
- PyArrowConversionRegister (class in *graviti.portex.register*), 180
- PyArrowPagingList (class in *graviti.paging.lists*), 139
- ## Q
- query() (*graviti.dataframe.frame.DataFrame* method), 57
- query() (*graviti.dataframe.sql.array.DataFrame* method), 47
- query() (*graviti.dataframe.sql.array.ScalarArray* method), 46
- ## R
- read() (*graviti.utility.requests.UserResponse* method), 193
- read\_json() (in module *graviti.portex.base*), 154
- read\_yaml() (in module *graviti.portex.base*), 154
- record (class in *graviti.portex.builtin*), 159
- RECORD\_KEY (in module *graviti.dataframe.container*), 51
- RemoteAudio (class in *graviti.file.audio*), 59
- RemoteFile (class in *graviti.file.base*), 61
- RemoteImage (class in *graviti.file.image*), 62
- RemotePointCloud (class in *graviti.file.point\_cloud*), 65
- remove() (*graviti.utility.collections.UserMutableSequence* method), 182

- `rename()` (*graviti.portex.base.PortexRecordBase* method), 153  
`rename()` (*graviti.portex.field.ConnectedFields* method), 171  
`rename()` (*graviti.portex.field.Fields* method), 169  
`rename()` (*graviti.portex.field.FrozenFields* method), 169  
`repo()` (*graviti.portex.package.ExternalPackage* property), 172  
`repr_config` (in module *graviti.utility.repr*), 190  
`ReprMixin` (class in *graviti.utility.repr*), 190  
`ReprType` (class in *graviti.utility.repr*), 190  
`request()` (*graviti.utility.requests.UserSession* method), 192  
`REQUEST_TEMPLATE` (in module *graviti.utility.log*), 188  
`RequestLogging` (class in *graviti.utility.log*), 188  
`RequestParamsMissingError`, 197  
`required()` (*graviti.portex.param.Param* property), 174  
`ResourceNameError`, 195  
`ResourceNotExistError`, 197  
`response` (*graviti.exception.ResponseError* attribute), 196  
`RESPONSE_ERROR_DISTRIBUTOR` (in module *graviti.openapi.requests*), 112  
`RESPONSE_TEMPLATE` (in module *graviti.utility.log*), 188  
`ResponseError`, 196  
`ResponseErrorRegister` (class in *graviti.exception*), 196  
`ResponseLogging` (class in *graviti.utility.log*), 188  
`ReturnGenerator` (class in *graviti.manager.lazy*), 79  
`reverse()` (*graviti.manager.lazy.LazyPagingList* method), 80  
`reverse()` (*graviti.utility.collections.UserMutableSequence* method), 182  
`RevisionType` (class in *graviti.manager.dataset*), 71  
`RowSeries` (class in *graviti.dataframe.sql.scalar*), 51  
`RowSeriesILocIndexer` (class in *graviti.dataframe.row.indexing*), 43  
`RowSeriesLocIndexer` (class in *graviti.dataframe.row.indexing*), 43
- ## S
- `S3ObjectPolicyManager` (class in *graviti.manager.policy*), 83  
`ScalarArray` (class in *graviti.dataframe.sql.array*), 45  
`ScalarWrapper` (class in *graviti.paging.wrapper*), 148  
`search()` (*graviti.manager.commit.Commit* method), 68  
`SearchContainer` (class in *graviti.dataframe.sql.container*), 49  
`SearchContainerRegister` (class in *graviti.dataframe.sql.container*), 49  
`SearchScalarContainer` (class in *graviti.dataframe.sql.container*), 48  
`send()` (*graviti.utility.requests.TimeoutHTTPAdapter* method), 192  
`Series` (class in *graviti.dataframe.column.series*), 41  
`Series` (class in *graviti.dataframe.row.series*), 43  
`SeriesBase` (class in *graviti.dataframe.column.series*), 40  
`ServiceUnavailableError`, 197  
`SESSIONS` (in module *graviti.utility.requests*), 192  
`set_item()` (*graviti.paging.lists.PagingListBase* method), 136  
`set_slice()` (*graviti.paging.lists.PagingListBase* method), 136  
`set_slice()` (*graviti.paging.lists.PyArrowPagingList* method), 139  
`set_slice_iterable()` (*graviti.paging.lists.PagingListBase* method), 137  
`setdefault()` (*graviti.utility.collections.UserMutableMapping* method), 183  
`setdefault()` (*graviti.utility.typing.NestedDict* method), 194  
`shape()` (*graviti.dataframe.frame.DataFrame* property), 54  
`SheetOperation` (class in *graviti.operation.sheet*), 130  
`Sheets` (class in *graviti.manager.sheets*), 84  
`shorten()` (in module *graviti.utility.common*), 185  
`size()` (*graviti.dataframe.frame.DataFrame* property), 54  
`size()` (*graviti.dataframe.sql.array.NumberScalarArray* method), 47  
`size()` (*graviti.file.base.File* property), 61  
`size()` (*graviti.file.base.FileBase* property), 60  
`SlicedPage` (class in *graviti.paging.page*), 143  
`STANDARD_URL` (in module *graviti.portex.register*), 179  
`StatusError`, 195  
`StatusWarning`, 70  
`string` (class in *graviti.portex.builtin*), 157  
`String` (class in *graviti.portex.ptype*), 177  
`string_factory_creator()` (in module *graviti.portex.factory*), 167  
`StringScalar` (class in *graviti.dataframe.sql.scalar*), 50  
`StringScalarArray` (class in *graviti.dataframe.sql.array*), 46  
`StructArrayWrapper` (class in *graviti.paging.wrapper*), 148  
`StructScalarWrapper` (class in *graviti.paging.wrapper*), 148  
`submit_multithread_tasks()` (in module *graviti.utility.requests*), 193  
`Subpackage` (class in *graviti.portex.package*), 172  
`sum()` (*graviti.dataframe.sql.array.NumberScalarArray* method), 47
- ## T
- `Tag` (class in *graviti.manager.tag*), 84  
`TagManager` (class in *graviti.manager.tag*), 85

- tags() (graviti.manager.dataset.Dataset property), 73
- tail() (graviti.dataframe.frame.DataFrame method), 55
- TIFF (class in graviti.file.image\_size), 64
- timeout (graviti.utility.requests.Config attribute), 191
- TimeoutHTTPAdapter (class in graviti.utility.requests), 191
- title() (graviti.manager.commit.Commit property), 67
- to\_builtin() (graviti.portex.base.PortexType method), 153
- to\_builtin() (graviti.portex.builtin.PortexBuiltinType method), 157
- to\_builtin() (graviti.portex.external.PortexExternalType method), 164
- to\_json() (graviti.portex.avro.AvroArraySchema method), 151
- to\_json() (graviti.portex.avro.AvroEnumSchema method), 151
- to\_json() (graviti.portex.avro.AvroField method), 150
- to\_json() (graviti.portex.avro.AvroPrimitiveSchema method), 150
- to\_json() (graviti.portex.avro.AvroRecordSchema method), 151
- to\_json() (graviti.portex.avro.AvroSchema method), 150
- to\_json() (graviti.portex.base.PortexType method), 152
- to\_pyarrow() (graviti.paging.lists.PyArrowPagingList method), 140
- to\_pyarrow() (graviti.portex.base.PortexRecordBase method), 154
- to\_pyarrow() (graviti.portex.base.PortexType method), 153
- to\_pyarrow() (graviti.portex.builtin.array method), 161
- to\_pyarrow() (graviti.portex.builtin.binary method), 158
- to\_pyarrow() (graviti.portex.builtin.boolean method), 158
- to\_pyarrow() (graviti.portex.builtin.enum method), 161
- to\_pyarrow() (graviti.portex.builtin.float32 method), 159
- to\_pyarrow() (graviti.portex.builtin.float64 method), 159
- to\_pyarrow() (graviti.portex.builtin.int32 method), 158
- to\_pyarrow() (graviti.portex.builtin.int64 method), 159
- to\_pyarrow() (graviti.portex.builtin.record method), 160
- to\_pyarrow() (graviti.portex.builtin.string method), 157
- to\_pyarrow() (graviti.portex.external.PortexExternalType method), 163
- to\_pyarrow() (graviti.portex.field.Fields method), 170
- to\_pylist() (graviti.dataframe.column.series.ArraySeries method), 42
- to\_pylist() (graviti.dataframe.column.series.EnumSeries method), 42
- to\_pylist() (graviti.dataframe.column.series.FileSeries method), 42
- to\_pylist() (graviti.dataframe.column.series.Series method), 42
- to\_pylist() (graviti.dataframe.container.Container method), 51
- to\_pylist() (graviti.dataframe.frame.DataFrame method), 57
- to\_pyobj() (graviti.portex.base.PortexType method), 152
- to\_pyobj() (graviti.portex.enum.EnumValueDict method), 162
- to\_pyobj() (graviti.portex.enum.EnumValueList method), 162
- to\_pyobj() (graviti.portex.enum.EnumValues method), 162
- to\_pyobj() (graviti.portex.field.Fields method), 170
- to\_pyobj() (graviti.portex.package.Imports method), 173
- to\_pyobj() (graviti.portex.package.Subpackage method), 172
- to\_pyobj() (graviti.portex.param.Param method), 174
- to\_pyobj() (graviti.portex.param.Params method), 175
- to\_yaml() (graviti.portex.base.PortexType method), 153
- total\_count (graviti.manager.lazy.InitPage attribute), 80
- transform\_kwargs() (graviti.portex.factory.TypeFactory method), 166
- type\_factory\_creator() (in module graviti.portex.factory), 167
- TypeBuilder (class in graviti.portex.builder), 155
- TypeFactory (class in graviti.portex.factory), 165
- ## U
- UnauthorizedError, 197
- UnionFields (in module graviti.portex.field), 170
- UnionFieldsFactory (in module graviti.portex.factory), 165
- update() (graviti.paging.offset.Offsets method), 141
- update() (graviti.portex.package.Imports method), 172
- update() (graviti.portex.param.Params method), 175
- update() (graviti.utility.collections.UserMutableMapping method), 184
- update\_data() (in module graviti.openapi.data), 96
- update\_dataset() (in module graviti.openapi.dataset), 103
- update\_draft() (in module graviti.openapi.draft), 107
- update\_schema() (in module graviti.openapi.schema), 113
- updated\_at (graviti.manager.dataset.Dataset attribute), 72
- UpdateData (class in graviti.operation.frame), 129
- UpdateSchema (class in graviti.operation.frame), 128
- upload() (graviti.manager.draft.Draft method), 76



[url\(\)](#) (*graviti.workspace.Workspace* property), 198  
[urlnorm\(\)](#) (*in module graviti.utility.common*), 185  
[UserMapping](#) (*class in graviti.utility.collections*), 182  
[UserMutableMapping](#) (*class in graviti.utility.collections*), 183  
[UserMutableSequence](#) (*class in graviti.utility.collections*), 181  
[UserResponse](#) (*class in graviti.utility.requests*), 192  
[UserSequence](#) (*class in graviti.utility.collections*), 181  
[UserSession](#) (*class in graviti.utility.requests*), 192  
[UtilityError](#), 194

## V

[value](#) (*graviti.manager.lazy.ReturnGenerator* attribute), 79  
[values\(\)](#) (*graviti.manager.sheets.Sheets* method), 84  
[values\(\)](#) (*graviti.paging.wrapper.ListScalarWrapper* property), 149  
[values\(\)](#) (*graviti.utility.collections.UserMapping* method), 183  
[VariableFactory](#) (*class in graviti.portex.factory*), 166

## W

[WebP](#) (*class in graviti.file.image\_size*), 64  
[width\(\)](#) (*graviti.file.image.Image* property), 62  
[width\(\)](#) (*graviti.file.image.RemoteImage* property), 62  
[Workspace](#) (*class in graviti.workspace*), 198  
[WrapperRegister](#) (*class in graviti.paging.wrapper*), 147