
Kubeflow Metadata Documentation

Google

Mar 13, 2020

CONTENTS

1 Metadata Package	3
Python Module Index	9
Index	11

Main documentation: <https://www.kubeflow.org/docs/components/metadata/>

Source code: <https://github.com/kubeflow/metadata/tree/master/sdk/python>

Tutorial: <https://github.com/kubeflow/metadata/blob/master/sdk/python/sample/demo.ipynb>

METADATA PACKAGE

class kubeflow.metadata.metadata.**ABC**
Bases: object

Helper class that provides a standard way to create an ABC using inheritance.

class kubeflow.metadata.metadata.**Artifact**
Bases: abc.ABC

abstract property **ARTIFACT_TYPE_NAME**
classmethod(function) -> method

Convert a function to be a class method.

A class method receives the class as implicit first argument, just like an instance method receives the instance. To declare a class method, use this idiom:

```
class C: @classmethod def f(cls, arg1, arg2, ...):  
    ...
```

It can be called either on the class (e.g. C.f()) or on an instance (e.g. C().f()). The instance is ignored except for its class. If a class method is called for a derived class, the derived class object is passed as the implied first argument.

Class methods are different than C++ or Java static methods. If you want those, see the staticmethod builtin.

static is_duplicated (a, b)
Checks if two artifacts are duplicated.

The artifacts may be considered duplication even if not all the fields are the same as in mlpb.Artifact. For example, two models can be considered the same if they have the same uri, name and version.

Returns True or False for duplication.

abstract serialization ()
Return type Artifact

class kubeflow.metadata.metadata.**DataSet** (uri=None, name=None, workspace=None, description=None, owner=None, version=None, query=None, labels=None, **kwargs)
Bases: *kubeflow.metadata.metadata.Artifact*

Dataset captures a data set in a machine learning workflow.

uri
Required uri of the data set.

name
Required name of the data set.

workspace
Optional name of the workspace.

description
Optional description of the data set.

owner
Optional owner of the data set.

version
Optional version tagged by the user.

query
Optional query string on how this data set being fetched from a data source.

labels
Optional string key value pairs for labels.

Example

```
>>> metadata.DataSet(description="an example data",
...                  name="mytable-dump",
...                  owner="owner@my-company.org",
...                  uri="file://path/to/dataset",
...                  version="v1.0.0",
...                  query="SELECT * FROM mytable",
...                  labels={"label1", "val1"}))
```

ARTIFACT_TYPE_NAME = 'kubeflow.org/alpha/data_set'

static is_duplicated (*a*, *b*)
Checks if two artifacts are duplicated.

The artifacts may be considered duplication even if not all the fields are the same as in `mlpb.Artifact`. For example, two models can be considered the same if they have the same uri, name and version.

Returns True or False for duplication.

serialization ()

class kubeflow.metadata.metadata.**Execution** (*name=None, workspace=None, run=None, description=None*)

Bases: object

Captures a run of pipeline or notebooks in a workspace and group executions.

Execution also serves as object for logging artifacts as its input or output.

EXECUTION_TYPE_NAME = 'kubeflow.org/alpha/execution'

log_input (*artifact*)
Log an artifact as an input of this execution.

Parameters **artifact** (*Artifact*) – Model, DataSet, Metrics or customized artifact type.

Return type *Artifact*

Returns The same artifact with artifact.id set.

log_output (*artifact*)

Log an artifact as an input of this execution.

Parameters **artifact** (*Artifact*) – Model, DataSet, Metrics or customized artifact type.

Return type *Artifact*

Returns The same artifact with artifact.id set.

serialized()

class kubeflow.metadata.metadata.**List**

Bases: list, typing.MutableSequence

class kubeflow.metadata.metadata.**Mapping**

Bases: collections.abc.Mapping, typing.Collection

class kubeflow.metadata.metadata.**Metrics** (*uri=None, name=None, workspace=None, description=None, owner=None, data_set_id=None, model_id=None, metrics_type=None, values=None, labels=None, **kwargs*)

Bases: *kubeflow.metadata.metadata.Artifact*

Captures an evaluation metrics of a model on a data set.

uri

Required uri of the metrics.

name

Required name of the metrics.

workspace

Optional name of the workspace.

description

Optional description of the metrics.

owner

Optional owner of the metrics.

data_set_id

Optional id of the data set used for evaluation.

model_id

Optional id of a evaluated model.

metrics_type

Optional type of the evaluation.

values

Optional map from metrics name to its value.

labels

Optional string key value pairs for labels.

Example

```
>>> metadata.Metrics(  
...     name="MNIST-evaluation",  
...     description=  
...     "validating the MNIST model to recognize handwritten digits",  
...     owner="someone@kubeflow.org",  
...     uri="gcs://my-bucket/mnist-eval.csv",  
...     data_set_id="123",  
...     model_id="12345",  
...     metrics_type=metadata.Metrics.VALIDATION,  
...     values={"accuracy": 0.95},  
...     labels={"mylabel": "11"})
```

```
ARTIFACT_TYPE_NAME = 'kubeflow.org/alpha/metrics'
```

```
PRODUCTION = 'production'
```

```
TESTING = 'testing'
```

```
TRAINING = 'training'
```

```
VALIDATION = 'validation'
```

```
serialization()
```

```
class kubeflow.metadata.metadata.Model(uri=None, name=None, workspace=None, de-  
                                         scription=None, owner=None, version=None,  
                                         model_type=None, training_framework=None,  
                                         hyperparameters=None, labels=None, **kwargs)
```

Bases: *kubeflow.metadata.metadata.Artifact*

Captures a machine learning model.

uri

Required uri of the model artifact, e.g. "gcs://path/to/model.h5".

name

Required name of the model.

workspace

Optional name of the workspace.

description

Optional description of the model.

owner

Optional owner of the model.

model_type

Optional type of the model.

training_framework

Optional framework used to train the model.

hyperparameters

Optional map from hyper param name to its value.

labels

Optional string key value pairs for labels.

kwargs

Optional additional keyword arguments are saved as additional properties of this model.

Example

```
>>> metadata.Model(name="MNIST",
...                 description="model to recognize handwritten digits",
...                 owner="someone@kubeflow.org",
...                 uri="gcs://my-bucket/mnist",
...                 model_type="neural network",
...                 training_framework={
...                     "name": "tensorflow",
...                     "version": "v1.0"
...                 },
...                 hyperparameters={
...                     "learning_rate": 0.5,
...                     "layers": [10, 3, 1],
...                     "early_stop": True
...                 },
...                 version="v0.0.1",
...                 labels={"mylabel": "11"})
```

```
ARTIFACT_TYPE_NAME = 'kubeflow.org/alpha/model'
```

```
static is_duplicated(a, b)
```

Checks if two artifacts are duplicated.

The artifacts may be considered duplication even if not all the fields are the same as in `mlpb.Artifact`. For example, two models can be considered the same if they have the same uri, name and version.

Returns True or False for duplication.

```
serialization()
```

```
class kubeflow.metadata.metadata.Run(workspace=None, name=None, description=None)
```

Bases: object

Run captures a run of pipeline or notebooks in a workspace and group executions.

```
class kubeflow.metadata.metadata.Store(grpc_host='metadata-grpc-service.kubeflow',
                                       grpc_port=8080, root_certificates=None, private_key=None, certificate_chain=None)
```

Bases: object

Metadata Store that connects to the Metadata gRPC service.

```
class kubeflow.metadata.metadata.Workspace(store=None, name=None, description=None, labels=None, reuse_workspace_if_exists=True, backend_url_prefix=None)
```

Bases: object

Groups a set of runs of pipelines, notebooks and their related artifacts and executions.

```
CONTEXT_TYPE_NAME = 'kubeflow.org/alpha/workspace'
```

```
list(artifact_type_name=None)
```

List all artifacts of a given type.

Parameters

- **{str} name of artifact type.** (*artifact_type_name*) –
- **a list of artifacts of the provided typed associated** (*Returns*) –

- **this workspace. Each artifact is represented as a dict.**
(with)–

Return type `List[Artifact]`

Returns A list of Artifact objects.

`kubeflow.metadata.metadata.abstractmethod(funcobj)`

A decorator indicating abstract methods.

Requires that the metaclass is `ABCMeta` or derived from it. A class that has a metaclass derived from `ABCMeta` cannot be instantiated unless all of its abstract methods are overridden. The abstract methods can be called using any of the normal ‘super’ call mechanisms.

Usage:

```
class C(metaclass=ABCMeta): @abstractmethod def my_abstract_method(self, ...):
```

```
...
```

`kubeflow.metadata.metadata.retry(*dargs, **dkw)`

Decorator function that instantiates the Retrying object @param ***dargs**: positional arguments passed to Retrying object @param ****dkw**: keyword arguments passed to the Retrying object

- `genindex`
- `modindex`

PYTHON MODULE INDEX

k

`kubeflow.metadata.metadata`, 3

A

ABC (class in kubeflow.metadata.metadata), 3
 abstractmethod() (in module kube-
 flow.metadata.metadata), 8
 Artifact (class in kubeflow.metadata.metadata), 3
 ARTIFACT_TYPE_NAME (kube-
 flow.metadata.metadata.DataSet attribute),
 4
 ARTIFACT_TYPE_NAME (kube-
 flow.metadata.metadata.Metrics attribute),
 6
 ARTIFACT_TYPE_NAME (kube-
 flow.metadata.metadata.Model attribute),
 7
 ARTIFACT_TYPE_NAME() (kube-
 flow.metadata.metadata.Artifact property),
 3

C

CONTEXT_TYPE_NAME (kube-
 flow.metadata.metadata.Workspace attribute),
 7

D

data_set_id (kubeflow.metadata.metadata.Metrics
 attribute), 5
 DataSet (class in kubeflow.metadata.metadata), 3
 description (kubeflow.metadata.metadata.DataSet
 attribute), 4
 description (kubeflow.metadata.metadata.Metrics
 attribute), 5
 description (kubeflow.metadata.metadata.Model at-
 tribute), 6

E

Execution (class in kubeflow.metadata.metadata), 4
 EXECUTION_TYPE_NAME (kube-
 flow.metadata.metadata.Execution attribute),
 4

H

hyperparameters (kube-
 flow.metadata.metadata.Model attribute),

6

I

is_duplicated() (kube-
 flow.metadata.metadata.Artifact static method),
 3
 is_duplicated() (kube-
 flow.metadata.metadata.DataSet static
 method), 4
 is_duplicated() (kube-
 flow.metadata.metadata.Model static method),
 7

K

kubeflow.metadata.metadata (module), 3
 kwargs (kubeflow.metadata.metadata.Model attribute),
 6

L

labels (kubeflow.metadata.metadata.DataSet at-
 tribute), 4
 labels (kubeflow.metadata.metadata.Metrics at-
 tribute), 5
 labels (kubeflow.metadata.metadata.Model attribute),
 6
 List (class in kubeflow.metadata.metadata), 5
 list() (kubeflow.metadata.metadata.Workspace
 method), 7
 log_input() (kubeflow.metadata.metadata.Execution
 method), 4
 log_output() (kube-
 flow.metadata.metadata.Execution method),
 4

M

Mapping (class in kubeflow.metadata.metadata), 5
 Metrics (class in kubeflow.metadata.metadata), 5
 metrics_type (kubeflow.metadata.metadata.Metrics
 attribute), 5
 Model (class in kubeflow.metadata.metadata), 6
 model_id (kubeflow.metadata.metadata.Metrics
 attribute), 5

`model_type` (*kubeflow.metadata.metadata.Model* attribute), 6

N

`name` (*kubeflow.metadata.metadata.DataSet* attribute), 3

`name` (*kubeflow.metadata.metadata.Metrics* attribute), 5

`name` (*kubeflow.metadata.metadata.Model* attribute), 6

O

`owner` (*kubeflow.metadata.metadata.DataSet* attribute), 4

`owner` (*kubeflow.metadata.metadata.Metrics* attribute), 5

`owner` (*kubeflow.metadata.metadata.Model* attribute), 6

P

`PRODUCTION` (*kubeflow.metadata.metadata.Metrics* attribute), 6

Q

`query` (*kubeflow.metadata.metadata.DataSet* attribute), 4

R

`retry()` (in module *kubeflow.metadata.metadata*), 8

`Run` (class in *kubeflow.metadata.metadata*), 7

S

`serialization()` (*kubeflow.metadata.metadata.Artifact* method), 3

`serialization()` (*kubeflow.metadata.metadata.DataSet* method), 4

`serialization()` (*kubeflow.metadata.metadata.Metrics* method), 6

`serialization()` (*kubeflow.metadata.metadata.Model* method), 7

`serialized()` (*kubeflow.metadata.metadata.Execution* method), 5

`Store` (class in *kubeflow.metadata.metadata*), 7

T

`TESTING` (*kubeflow.metadata.metadata.Metrics* attribute), 6

`TRAINING` (*kubeflow.metadata.metadata.Metrics* attribute), 6

`training_framework` (*kubeflow.metadata.metadata.Model* attribute), 6

U

`uri` (*kubeflow.metadata.metadata.DataSet* attribute), 3

`uri` (*kubeflow.metadata.metadata.Metrics* attribute), 5

`uri` (*kubeflow.metadata.metadata.Model* attribute), 6

V

`VALIDATION` (*kubeflow.metadata.metadata.Metrics* attribute), 6

`values` (*kubeflow.metadata.metadata.Metrics* attribute), 5

`version` (*kubeflow.metadata.metadata.DataSet* attribute), 4

W

`Workspace` (class in *kubeflow.metadata.metadata*), 7

`workspace` (*kubeflow.metadata.metadata.DataSet* attribute), 4

`workspace` (*kubeflow.metadata.metadata.Metrics* attribute), 5

`workspace` (*kubeflow.metadata.metadata.Model* attribute), 6