
Elemeno MLOps Python Client

Release 0.0.20

engineering@elemeno.ai

Dec 12, 2022

TABLE OF CONTENTS

| | | |
|----------|--|-----------|
| 1 | Modules | 1 |
| 1.1 | MLOps Client API | 1 |
| 1.2 | Datasource API | 1 |
| 1.2.1 | Datasource Type | 1 |
| 1.2.1.1 | Attributes | 1 |
| 1.2.2 | GCP Authentication Type | 1 |
| 1.2.3 | Redshift Authentication Type | 1 |
| 1.3 | Feature Store API | 1 |
| 1.3.1 | Feature Key Type | 1 |
| 1.3.2 | Feature Value Type | 2 |
| 1.3.3 | Feature Key Type | 2 |
| 1.4 | Inference Server API | 2 |
| 1.4.1 | Feature Source Type | 3 |
| 1.4.2 | Feature Source | 3 |
| 2 | Indices and tables | 5 |
| 3 | Indices and tables | 7 |
| | Python Module Index | 9 |
| | Index | 11 |

MODULES

1.1 MLOps Client API

1.2 Datasource API

1.2.1 Datasource Type

class `mlops_client.datasource.datasource_type.DatasourceType`(*value*)

This is a class that defines the type of datasource that is being used.

The DatasourceType class is an enumeration class that defines the type of datasource that is being used.

1.2.1.1 Attributes

REDSHIFT

[DatasourceType] The REDSHIFT datasource type

BIGQUERY

[DatasourceType] The BIGQUERY datasource type

CSV

[DatasourceType] The CSV datasource type

exception `mlops_client.datasource.datasource_type.InvalidTypeError`(*dstype: str*)

1.2.2 GCP Authentication Type

1.2.3 Redshift Authentication Type

1.3 Feature Store API

1.3.1 Feature Key Type

class `mlops_client.feature_store.feature_key.FeatureKey`

This class is used to build a feature key.

Object building functions:

- `with_key_name` (str): The name of the feature key.

- `with_key_value_type` (FeatureValueType): The type of the feature key.
- `build`: Returns a complete instance of the object.

Returns:

FeatureKey: A feature key.

1.3.2 Feature Value Type

class `mlops_client.feature_store.feature_value_type.FeatureValueType`(*value*)

FeatureValueType is an enumeration of the possible types of values that a feature can have.

STRING: A string value. FLOAT: A floating point value. INTEGER: An integer value. ARRAY: An array of values.

exception `mlops_client.feature_store.feature_value_type.InvalidFeatureValueTypeError`(*value_type: str*)

1.3.3 Feature Key Type

1.4 Inference Server API

class `mlops_client.inference_server.inferenceserver_client.InferenceServer`(*headers: Dict[str, str], host: str, client: Optional[ClientSession] = None*)

async `create_rest`(*model_path: str, num_instances: int, sources: List[FeatureSource]*) → Any

Creates a REST inference server for a given model.

Args:

- `model_path`: Path to the model file.
- `num_instances`: Number of instances to create.
- `sources`: A list of FeatureSource objects.

Returns:

- A list of InferenceServer objects.

async `list`(*offset: Optional[str] = None, limit: Optional[int] = None*) → Any

List all inference servers.

Parameters:

- `offset`: An optional string that represents the starting item, should be the value of 'next' field from the previous response.
- `limit`: An optional integer to limit the number of returned items.

Returns:

- A list of InferenceServer objects.

1.4.1 Feature Source Type

class `mlops_client.inference_server.feature_source_type.FeatureSourceType(value)`

FeatureSourceType is an enumeration of the possible sources of features for a feature set.

FEATURE_TABLE: The feature set is based on a feature table. REQUEST_BODY: The feature set is based on a request body. REQUEST_BODY_KEY: The feature set is based on a request body key.

1.4.2 Feature Source

class `mlops_client.inference_server.feature_source.FeatureSource`

Object building functions:

- `with_source_type`: The type of the feature source.
- `with_feature_table_id`: The id of the feature table.
- `with_feature_name`: The name of the feature.
- `with_body_json_path`: The path to the feature in the request body.
- `build`: Returns a complete instance of the object.

Raises:

- `MissingFieldError`: If a required field is not provided.
- `InvalidFeatureValueTypeError`: If the provided source type is invalid.

Returns:

- A feature source.

INDICES AND TABLES

- `genindex`

INDICES AND TABLES

- `genindex`
- `search`

PYTHON MODULE INDEX

m

- `mlops_client.datasource.datasource_type`, 1
- `mlops_client.feature_store.feature_key`, 1
- `mlops_client.feature_store.feature_value_type`,
2
- `mlops_client.inference_server.feature_source`,
3
- `mlops_client.inference_server.feature_source_type`,
3
- `mlops_client.inference_server.inferenceserver_client`,
2

INDEX

C
`create_rest()` (*mlops_client.inference_server.inferenceserver_client.InferenceServer*
method), 2

D
`DatasourceType` (class in *mlops_client.datasource.datasource_type*), 1
`mlops_client.inference_server.feature_source_type`
module, 3
`mlops_client.inference_server.inferenceserver_client`
module, 2

F
`FeatureKey` (class in *mlops_client.feature_store.feature_key*), 1
`FeatureSource` (class in *mlops_client.inference_server.feature_source*), 3
`FeatureSourceType` (class in *mlops_client.inference_server.feature_source_type*), 3
`FeatureValueType` (class in *mlops_client.feature_store.feature_value_type*), 2

I
`InferenceServer` (class in *mlops_client.inference_server.inferenceserver_client*), 2
`InvalidFeatureValueTypeError`, 2
`InvalidTypeError`, 1

L
`list()` (*mlops_client.inference_server.inferenceserver_client.InferenceServer*
method), 2

M
`mlops_client.datasource.datasource_type`
module, 1
`mlops_client.feature_store.feature_key`
module, 1
`mlops_client.feature_store.feature_value_type`
module, 2
`mlops_client.inference_server.feature_source`
module, 3