
Feast Documentation

Feast Authors

Jul 12, 2022

CONTENTS

1	Client	1
2	Data Source	3
3	Entity	7
4	Feature Table	9
5	Feature	11
6	Constants	13
	Python Module Index	17
	Index	19

CHAPTER
ONE

CLIENT

DATA SOURCE

```
class feast.data_source.BigQueryOptions(table_ref: str)
    DataSource BigQuery options used to source features from BigQuery query

classmethod from_proto(bigquery_options_proto: BigQueryOptions)
    Creates a BigQueryOptions from a protobuf representation of a BigQuery option

    Parameters
        bigquery_options_proto – A protobuf representation of a DataSource

    Returns
        Returns a BigQueryOptions object based on the bigquery_options protobuf

property table_ref
    Returns the table ref of this BQ table

to_proto() → BigQueryOptions
    Converts an BigQueryOptionsProto object to its protobuf representation.

    Returns
        BigQueryOptionsProto protobuf

class feast.data_source.BigQuerySource(event_timestamp_column: str, table_ref: str,
                                         created_timestamp_column: Optional[str] = "", field_mapping:
                                         Optional[Dict[str, str]] = None, date_partition_column:
                                         Optional[str] = "")

    property bigquery_options
        Returns the bigquery options of this data source

    to_proto() → DataSource
        Converts an DataSourceProto object to its protobuf representation.

class feast.data_source.DataSource(event_timestamp_column: str, created_timestamp_column:
                                     Optional[str] = "", field_mapping: Optional[Dict[str, str]] = None,
                                     date_partition_column: Optional[str] = "")

    DataSource that can be used source features

    property created_timestamp_column
        Returns the created timestamp column of this data source

    property date_partition_column
        Returns the date partition column of this data source
```

property event_timestamp_column

Returns the event timestamp column of this data source

property field_mapping

Returns the field mapping of this data source

static from_proto(data_source)

Convert data source config in FeatureTable spec to a DataSource class object.

to_proto() → DataSource

Converts an DataSourceProto object to its protobuf representation.

class feast.data_source.**FileOptions**(*file_format: FileFormat, file_url: str*)

DataSource File options used to source features from a file

property file_format

Returns the file format of this file

property file_url

Returns the file url of this file

classmethod from_proto(*file_options_proto: FileOptions*)

Creates a FileOptions from a protobuf representation of a file option

Parameters

file_options_proto – a protobuf representation of a datasource

Returns

Returns a FileOptions object based on the file_options protobuf

to_proto() → FileOptions

Converts an FileOptionsProto object to its protobuf representation.

Returns

FileOptionsProto protobuf

class feast.data_source.**FileSource**(*event_timestamp_column: str, file_format: FileFormat, file_url: str, created_timestamp_column: Optional[str] = "", field_mapping: Optional[Dict[str, str]] = None, date_partition_column: Optional[str] = ""*)

property file_options

Returns the file options of this data source

to_proto() → DataSource

Converts an DataSourceProto object to its protobuf representation.

class feast.data_source.**KafkaOptions**(*bootstrap_servers: str, message_format: StreamFormat, topic: str*)

DataSource Kafka options used to source features from Kafka messages

property bootstrap_servers

Returns a comma-separated list of Kafka bootstrap servers

classmethod from_proto(*kafka_options_proto: KafkaOptions*)

Creates a KafkaOptions from a protobuf representation of a kafka option

Parameters

kafka_options_proto – A protobuf representation of a DataSource

Returns

Returns a BigQueryOptions object based on the kafka_options protobuf

property message_format

Returns the data format that is used to encode the feature data in Kafka messages

to_proto() → KafkaOptions

Converts an KafkaOptionsProto object to its protobuf representation.

Returns

KafkaOptionsProto protobuf

property topic

Returns the Kafka topic to collect feature data from

```
class feast.data_source.KafkaSource(event_timestamp_column: str, bootstrap_servers: str,
                                   message_format: StreamFormat, topic: str,
                                   created_timestamp_column: Optional[str] = "", field_mapping:
                                   Optional[Dict[str, str]] = {}, date_partition_column: Optional[str] =
                                   "")
```

property kafka_options

Returns the kafka options of this data source

to_proto() → DataSource

Converts an DataSourceProto object to its protobuf representation.

```
class feast.data_source.KinesisOptions(record_format: StreamFormat, region: str, stream_name: str)
```

DataSource Kinesis options used to source features from Kinesis records

```
classmethod from_proto(kinesis_options_proto: KinesisOptions)
```

Creates a KinesisOptions from a protobuf representation of a kinesis option

Parameters

kinesis_options_proto – A protobuf representation of a DataSource

Returns

Returns a KinesisOptions object based on the kinesis_options protobuf

property record_format

Returns the data format used to encode the feature data in the Kinesis records.

property region

Returns the AWS region of Kinesis stream

property stream_name

Returns the Kinesis stream name to obtain feature data from

to_proto() → KinesisOptions

Converts an KinesisOptionsProto object to its protobuf representation.

Returns

KinesisOptionsProto protobuf

```
class feast.data_source.KinesisSource(event_timestamp_column: str, created_timestamp_column: str,
                                       record_format: StreamFormat, region: str, stream_name: str,
                                       field_mapping: Optional[Dict[str, str]] = {},
                                       date_partition_column: Optional[str] = "")
```

property kinesis_options

Returns the kinesis options of this data source

to_proto() → DataSource

Converts an DataSourceProto object to its protobuf representation.

class feast.data_source.**SourceType**(*value*)

DataSource value type. Used to define source types in DataSource.

ENTITY

```
class feast.entity.Entity(name: str, description: str, value_type: ValueType, labels:
    Optional[MutableMapping[str, str]] = None)
```

Represents a collection of entities and associated metadata.

property created_timestamp

Returns the created_timestamp of this entity

property description

Returns the description of this entity

classmethod from_dict(entity_dict)

Creates an entity from a dict

Parameters

entity_dict – A dict representation of an entity

Returns

Returns a EntityV2 object based on the entity dict

classmethod from_proto(entity_proto: Entity)

Creates an entity from a protobuf representation of an entity

Parameters

entity_proto – A protobuf representation of an entity

Returns

Returns a EntityV2 object based on the entity protobuf

classmethod from_yaml(yml: str)

Creates an entity from a YAML string body or a file path

Parameters

yml – Either a file path containing a yaml file or a YAML string

Returns

Returns a EntityV2 object based on the YAML file

is_valid()

Validates the state of a entity locally. Raises an exception if entity is invalid.

property labels

Returns the labels of this entity. This is the user defined metadata defined as a dictionary.

property last_updated_timestamp

Returns the last_updated_timestamp of this entity

property name

Returns the name of this entity

to_dict() → Dict

Converts entity to dict

Returns

Dictionary object representation of entity

to_proto() → Entity

Converts an entity object to its protobuf representation

Returns

EntityV2Proto protobuf

to_spec_proto() → EntitySpecV2

Converts an EntityV2 object to its protobuf representation. Used when passing EntitySpecV2 object to Feast request.

Returns

EntitySpecV2 protobuf

to_yaml()

Converts a entity to a YAML string.

Returns

Entity string returned in YAML format

property value_type

Returns the type of this entity

FEATURE TABLE

```
class feast.feature_table.FeatureTable(name: str, entities: List[str], features: List[Feature],  
                                         batch_source: Optional[Union[BigQuerySource, FileSource]] =  
                                         None, stream_source: Optional[Union[KafkaSource,  
                                         KinesisSource]] = None, max_age: Optional[Duration] = None,  
                                         labels: Optional[MutableMapping[str, str]] = None,  
                                         online_store: Optional[OnlineStore] = None)
```

Represents a collection of features and associated metadata.

add_feature(feature: *Feature*)

Adds a new feature to the feature table.

property batch_source

Returns the batch source of this feature table

property created_timestamp

Returns the created_timestamp of this feature table

property entities

Returns the entities of this feature table

property features

Returns the features of this feature table

classmethod from_dict(ft_dict)

Creates a feature table from a dict

Parameters

ft_dict – A dict representation of a feature table

Returns

Returns a FeatureTable object based on the feature table dict

classmethod from_proto(feature_table_proto: *FeatureTable*)

Creates a feature table from a protobuf representation of a feature table

Parameters

feature_table_proto – A protobuf representation of a feature table

Returns

Returns a FeatureTableProto object based on the feature table protobuf

classmethod from_yaml(yaml: *str*)

Creates a feature table from a YAML string body or a file path

Parameters

yaml – Either a file path containing a yaml file or a YAML string

Returns

Returns a FeatureTable object based on the YAML file

is_valid()

Validates the state of a feature table locally. Raises an exception if feature table is invalid.

property labels

Returns the labels of this feature table. This is the user defined metadata defined as a dictionary.

property last_updated_timestamp

Returns the last_updated_timestamp of this feature table

property max_age

Returns the maximum age of this feature table. This is the total maximum amount of staleness that will be allowed during feature retrieval for each specific feature that is looked up.

property name

Returns the name of this feature table

property online_store

Returns the online store of this feature table

property stream_source

Returns the stream source of this feature table

to_dict() → Dict

Converts feature table to dict

Returns

Dictionary object representation of feature table

to_proto() → FeatureTable

Converts a feature table object to its protobuf representation

Returns

FeatureTableProto protobuf

to_spec_proto() → FeatureTableSpec

Converts an FeatureTableProto object to its protobuf representation. Used when passing FeatureTableSpecProto object to Feast request.

Returns

FeatureTableSpecProto protobuf

to_yaml()

Converts a feature table to a YAML string.

Returns

Feature table string returned in YAML format

FEATURE

```
class feast.feature.Feature(name: str, dtype: ValueType, labels: Optional[MutableMapping[str, str]] =
                             None)
```

Feature field type

property dtype: *ValueType*

Getter for data type of this field

classmethod from_proto(feature_proto: *FeatureSpecV2*)

Parameters

feature_proto – *FeatureSpecV2* protobuf object

Returns

Feature object

property labels: *MutableMapping[str, str]*

Getter for labels of this field

property name

Getter for name of this field

to_proto() → *FeatureSpecV2*

Converts Feature object to its Protocol Buffer representation

```
class feast.feature.FeatureRef(name: str, feature_table: Optional[str] = None)
```

Feature Reference represents a reference to a specific feature.

classmethod from_proto(proto: *FeatureReferenceV2*)

Construct a feature reference from the given *FeatureReference* proto Arg:

proto: Protobuf *FeatureReference* to construct from

Returns

FeatureRef that refers to the given feature

classmethod from_str(feature_ref_str: *str*)

Parse the given string feature reference into *FeatureRef* model String feature reference should be in the format *feature_table:feature*. Where “*feature_table*” and “*name*” are the *feature_table* name and feature name respectively. :param *feature_ref_str*: String representation of the feature reference

Returns

FeatureRef that refers to the given feature

to_proto() → FeatureReferenceV2

Convert and return this feature table reference to protobuf. :returns: Protobuf representation of this feature table reference.

CONSTANTS

```
feast.constants.CONFIG_FEAST_ENV_VAR_PREFIX: str = 'FEAST_'
    Default prefix to Feast environmental variables

feast.constants.CONFIG_FILE_DEFAULT_DIRECTORY: str = '.feast'
    Default directory to Feast configuration file

feast.constants.CONFIG_FILE_NAME: str = 'config'
    Default Feast configuration file name

feast.constants.CONFIG_FILE_SECTION: str = 'general'
    Default section in Feast configuration file to specify options

class feast.constants.ConfigOptions
    Feast Configuration Options

    AUTH_PROVIDER: str = 'auth_provider'
        Authentication Provider - Google OpenID/OAuth
        Options: "google" / "oauth"

    AUTH_TOKEN: Optional[str] = 'auth_token'
        JWT Auth token for user authentication to Feast

    AZURE_BLOB_ACCOUNT_ACCESS_KEY: Optional[str] = 'azure_blob_account_access_key'
        Account access key for Azure blob storage_client

    AZURE_BLOB_ACCOUNT_NAME: Optional[str] = 'azure_blob_account_name'
        Account name for Azure blob storage_client

    BATCH_FEATURE_REQUEST_WAIT_TIME_SECONDS: str =
    'batch_feature_request_wait_time_seconds'
        Time to wait for historical feature requests before timing out.

    BATCH_INGESTION_PRODUCTION_TIMEOUT: str = 'batch_ingestion_production_timeout'
        Default timeout when running batch ingestion

    CORE_ENABLE_SSL: str = 'core_enable_ssl'
        Enable or disable TLS/SSL to Feast Core

    CORE_SERVER_SSL_CERT: str = 'core_server_ssl_cert'
        Path to certificate(s) to secure connection to Feast Core

    CORE_URL: str = 'core_url'
        Default Feast Core URL
```

DATAPROC_CLUSTER_NAME: `Optional[str] = 'dataproc_cluster_name'`

Dataproc cluster to run Feast Spark Jobs in

DATAPROC_EXECUTOR_CORES = `'dataproc_executor_cores'`

No. of executor cores for Dataproc cluster

DATAPROC_EXECUTOR_INSTANCES = `'dataproc_executor_instances'`

No. of executor instances for Dataproc cluster

DATAPROC_EXECUTOR_MEMORY = `'dataproc_executor_memory'`

No. of executor memory for Dataproc cluster

DATAPROC_PROJECT: `Optional[str] = 'dataproc_project'`

Project of Dataproc cluster

DATAPROC_REGION: `Optional[str] = 'dataproc_region'`

Region of Dataproc cluster

DEADLETTER_PATH: `str = 'deadletter_path'`

Ingestion Job DeadLetter Destination. The choice of storage is connected to the choice of SPARK_LAUNCHER.

Eg. gs://some-bucket/output/, s3://some-bucket/output/, file:///data/subfolder/

EMR_CLUSTER_ID: `Optional[str] = 'emr_cluster_id'`

EMR cluster to run Feast Spark Jobs in

EMR_CLUSTER_TEMPLATE_PATH: `Optional[str] = 'emr_cluster_template_path'`

Template path of EMR cluster

EMR_LOG_LOCATION: `Optional[str] = 'emr_log_location'`

Log path of EMR cluster

EMR_REGION: `Optional[str] = 'emr_region'`

Region of EMR cluster

ENABLE_AUTH: `str = 'enable_auth'`

Enable user authentication to Feast Core

GRPC_CONNECTION_TIMEOUT: `str = 'grpc_connection_timeout'`

Default connection timeout to Feast Serving, Feast Core, and Feast Job Service (in seconds)

GRPC_CONNECTION_TIMEOUT_APPLY: `str = 'grpc_connection_timeout_apply'`

Default gRPC connection timeout when sending an ApplyFeatureTable command to Feast Core (in seconds)

HISTORICAL_FEATURE_OUTPUT_FORMAT: `str = 'historical_feature_output_format'`

File format of historical retrieval features

HISTORICAL_FEATURE_OUTPUT_LOCATION: `Optional[str] = 'historical_feature_output_location'`

File location of historical retrieval features

INGESTION_DROP_INVALID_ROWS = `'ingestion_drop_invalid_rows'`

If set to true rows that do not pass custom validation (see feast.contrib.validation) won't be saved to Online Storage

JOB_SERVICE_ENABLE_CONTROL_LOOP: `str = 'job_service_enable_control_loop'`

Enable or disable control loop for Feast Job Service

JOB_SERVICE_ENABLE_SSL: `str = 'job_service_enable_ssl'`

Enable or disable TLS/SSL to Feast Job Service

JOB_SERVICE_SERVER_SSL_CERT: `str = 'job_service_server_ssl_cert'`

Path to certificate(s) to secure connection to Feast Job Service

JOB_SERVICE_URL: `Optional[str] = 'job_service_url'`

Default Feast Job Service URL

OAUTH_AUDIENCE: `Optional[str] = 'oauth_audience'`

OAuth intended recipients

OAUTH_CLIENT_ID: `Optional[str] = 'oauth_client_id'`

OAuth client ID

OAUTH_CLIENT_SECRET: `Optional[str] = 'oauth_client_secret'`

OAuth client secret

OAUTH_GRANT_TYPE: `Optional[str] = 'oauth_grant_type'`

OAuth grant type

OAUTH_TOKEN_REQUEST_URL: `Optional[str] = 'oauth_token_request_url'`

OAuth token request url

PROJECT: `str = 'project'`

Feast project namespace to use

REDIS_HOST: `str = 'redis_host'`

Default Redis host

REDIS_PORT: `str = 'redis_port'`

Default Redis port

REDIS_SSL: `str = 'redis_ssl'`

Enable or disable TLS/SSL to Redis

S3_ENDPOINT_URL: `Optional[str] = 's3_endpoint_url'`

Endpoint URL for S3 storage_client

SERVING_ENABLE_SSL: `str = 'serving_enable_ssl'`

Enable or disable TLS/SSL to Feast Serving

SERVING_SERVER_SSL_CERT: `str = 'serving_server_ssl_cert'`

Path to certificate(s) to secure connection to Feast Serving

SERVING_URL: `str = 'serving_url'`

Default Feast Serving URL

SPARK_BQ_MATERIALIZATION_DATASET: `Optional[str] = 'spark_bq_materialization_dataset'`

The dataset id where the materialized view of BigQuerySource is going to be created by default, use the same dataset where view is located

SPARK_BQ_MATERIALIZATION_PROJECT: `Optional[str] = 'spark_bq_materialization_project'`

The project id where the materialized view of BigQuerySource is going to be created by default, use the same project where view is located

SPARK_HOME: `Optional[str] = 'spark_home'`

Directory where Spark is installed

SPARK_INGESTION_JAR: `str = 'spark_ingestion_jar'`

Feast Spark Job ingestion jar file. The choice of storage is connected to the choice of SPARK_LAUNCHER.

Eg. “dataproc” (http and gs), “emr” (http and s3), “standalone” (http and file)

SPARK_LAUNCHER: `Optional[str] = 'spark_launcher'`

Spark Job launcher. The choice of storage is connected to the choice of SPARK_LAUNCHER.

Options: “standalone”, “dataproc”, “emr”

SPARK_STAGING_LOCATION: `Optional[str] = 'spark_staging_location'`

Feast Spark Job ingestion jobs staging location. The choice of storage is connected to the choice of SPARK_LAUNCHER.

Eg. gs://some-bucket/output/, s3://some-bucket/output/, file:///data/subfolder/

SPARK_STANDALONE_MASTER: `str = 'spark_standalone_master'`

Spark resource manager master url

STATSD_ENABLED: `str = 'statsd_enabled'`

Enable or disable StatsD

STATSD_HOST: `Optional[str] = 'statsd_host'`

Default StatsD port

STATSD_PORT: `Optional[str] = 'statsd_port'`

Default StatsD port

STENCIL_URL: `str = 'stencil_url'`

ProtoRegistry Address (currently only Stencil Server is supported as registry) <https://github.com/gojekfarm/stencil>

TELEMETRY = 'telemetry'

Telemetry enabled

`feast.constants.DATETIME_COLUMN:` `str = 'datetime'`

Default datetime column name for point-in-time join

`feast.constants.FEAST_CONFIG_FILE_ENV:` `str = 'FEAST_CONFIG'`

Environmental variable to specify Feast configuration file location

PYTHON MODULE INDEX

f

- `feast.constants`, [13](#)
- `feast.data_source`, [3](#)
- `feast.entity`, [7](#)
- `feast.feature`, [11](#)
- `feast.feature_table`, [9](#)

INDEX

A

`add_feature()` (*feast.feature_table.FeatureTable* method), 9
`AUTH_PROVIDER` (*feast.constants.ConfigOptions* attribute), 13
`AUTH_TOKEN` (*feast.constants.ConfigOptions* attribute), 13
`AZURE_BLOB_ACCOUNT_ACCESS_KEY` (*feast.constants.ConfigOptions* attribute), 13
`AZURE_BLOB_ACCOUNT_NAME` (*feast.constants.ConfigOptions* attribute), 13

B

`BATCH_FEATURE_REQUEST_WAIT_TIME_SECONDS` (*feast.constants.ConfigOptions* attribute), 13
`BATCH_INGESTION_PRODUCTION_TIMEOUT` (*feast.constants.ConfigOptions* attribute), 13
`batch_source` (*feast.feature_table.FeatureTable* property), 9
`bigquery_options` (*feast.data_source.BigQuerySource* property), 3
`BigQueryOptions` (class in *feast.data_source*), 3
`BigQuerySource` (class in *feast.data_source*), 3
`bootstrap_servers` (*feast.data_source.KafkaOptions* property), 4

C

`CONFIG_FEAST_ENV_VAR_PREFIX` (in module *feast.constants*), 13
`CONFIG_FILE_DEFAULT_DIRECTORY` (in module *feast.constants*), 13
`CONFIG_FILE_NAME` (in module *feast.constants*), 13
`CONFIG_FILE_SECTION` (in module *feast.constants*), 13
`ConfigOptions` (class in *feast.constants*), 13
`CORE_ENABLE_SSL` (*feast.constants.ConfigOptions* attribute), 13
`CORE_SERVER_SSL_CERT` (*feast.constants.ConfigOptions* attribute), 13
`CORE_URL` (*feast.constants.ConfigOptions* attribute), 13

`created_timestamp` (*feast.entity.Entity* property), 7
`created_timestamp` (*feast.feature_table.FeatureTable* property), 9
`created_timestamp_column` (*feast.data_source.DataSource* property), 3

D

`DATAPROC_CLUSTER_NAME` (*feast.constants.ConfigOptions* attribute), 13
`DATAPROC_EXECUTOR_CORES` (*feast.constants.ConfigOptions* attribute), 14
`DATAPROC_EXECUTOR_INSTANCES` (*feast.constants.ConfigOptions* attribute), 14
`DATAPROC_EXECUTOR_MEMORY` (*feast.constants.ConfigOptions* attribute), 14
`DATAPROC_PROJECT` (*feast.constants.ConfigOptions* attribute), 14
`DATAPROC_REGION` (*feast.constants.ConfigOptions* attribute), 14
`DataSource` (class in *feast.data_source*), 3
`date_partition_column` (*feast.data_source.DataSource* property), 3
`DATETIME_COLUMN` (in module *feast.constants*), 16
`DEADLETTER_PATH` (*feast.constants.ConfigOptions* attribute), 14
`description` (*feast.entity.Entity* property), 7
`dtype` (*feast.feature.Feature* property), 11

E

`EMR_CLUSTER_ID` (*feast.constants.ConfigOptions* attribute), 14
`EMR_CLUSTER_TEMPLATE_PATH` (*feast.constants.ConfigOptions* attribute), 14
`EMR_LOG_LOCATION` (*feast.constants.ConfigOptions* attribute), 14

EMR_REGION (*feast.constants.ConfigOptions* attribute), 14

ENABLE_AUTH (*feast.constants.ConfigOptions* attribute), 14

entities (*feast.feature_table.FeatureTable* property), 9

Entity (class in *feast.entity*), 7

event_timestamp_column (*feast.data_source.DataSource* property), 3

F

feast.constants module, 13

feast.data_source module, 3

feast.entity module, 7

feast.feature module, 11

feast.feature_table module, 9

FEAST_CONFIG_FILE_ENV (in module *feast.constants*), 16

Feature (class in *feast.feature*), 11

FeatureRef (class in *feast.feature*), 11

features (*feast.feature_table.FeatureTable* property), 9

FeatureTable (class in *feast.feature_table*), 9

field_mapping (*feast.data_source.DataSource* property), 4

file_format (*feast.data_source.FileOptions* property), 4

file_options (*feast.data_source.FileSource* property), 4

file_url (*feast.data_source.FileOptions* property), 4

FileOptions (class in *feast.data_source*), 4

FileSource (class in *feast.data_source*), 4

from_dict() (*feast.entity.Entity* class method), 7

from_dict() (*feast.feature_table.FeatureTable* class method), 9

from_proto() (*feast.data_source.BigQueryOptions* class method), 3

from_proto() (*feast.data_source.DataSource* static method), 4

from_proto() (*feast.data_source.FileOptions* class method), 4

from_proto() (*feast.data_source.KafkaOptions* class method), 4

from_proto() (*feast.data_source.KinesisOptions* class method), 5

from_proto() (*feast.entity.Entity* class method), 7

from_proto() (*feast.feature.Feature* class method), 11

from_proto() (*feast.feature.FeatureRef* class method), 11

from_proto() (*feast.feature_table.FeatureTable* class method), 9

from_str() (*feast.feature.FeatureRef* class method), 11

from_yaml() (*feast.entity.Entity* class method), 7

from_yaml() (*feast.feature_table.FeatureTable* class method), 9

G

GRPC_CONNECTION_TIMEOUT (*feast.constants.ConfigOptions* attribute), 14

GRPC_CONNECTION_TIMEOUT_APPLY (*feast.constants.ConfigOptions* attribute), 14

H

HISTORICAL_FEATURE_OUTPUT_FORMAT (*feast.constants.ConfigOptions* attribute), 14

HISTORICAL_FEATURE_OUTPUT_LOCATION (*feast.constants.ConfigOptions* attribute), 14

I

INGESTION_DROP_INVALID_ROWS (*feast.constants.ConfigOptions* attribute), 14

is_valid() (*feast.entity.Entity* method), 7

is_valid() (*feast.feature_table.FeatureTable* method), 10

J

JOB_SERVICE_ENABLE_CONTROL_LOOP (*feast.constants.ConfigOptions* attribute), 14

JOB_SERVICE_ENABLE_SSL (*feast.constants.ConfigOptions* attribute), 14

JOB_SERVICE_SERVER_SSL_CERT (*feast.constants.ConfigOptions* attribute), 15

JOB_SERVICE_URL (*feast.constants.ConfigOptions* attribute), 15

K

kafka_options (*feast.data_source.KafkaSource* property), 5

KafkaOptions (class in *feast.data_source*), 4

KafkaSource (class in *feast.data_source*), 5

kinesis_options (*feast.data_source.KinesisSource* property), 5

KinesisOptions (class in *feast.data_source*), 5

KinesisSource (class in *feast.data_source*), 5

L

labels (*feast.entity.Entity* property), 7
 labels (*feast.feature.Feature* property), 11
 labels (*feast.feature_table.FeatureTable* property), 10
 last_updated_timestamp (*feast.entity.Entity* property), 7
 last_updated_timestamp (*feast.feature_table.FeatureTable* property), 10

M

max_age (*feast.feature_table.FeatureTable* property), 10
 message_format (*feast.data_source.KafkaOptions* property), 5
 module
 feast.constants, 13
 feast.data_source, 3
 feast.entity, 7
 feast.feature, 11
 feast.feature_table, 9

N

name (*feast.entity.Entity* property), 7
 name (*feast.feature.Feature* property), 11
 name (*feast.feature_table.FeatureTable* property), 10

O

OAUTH_AUDIENCE (*feast.constants.ConfigOptions* attribute), 15
 OAUTH_CLIENT_ID (*feast.constants.ConfigOptions* attribute), 15
 OAUTH_CLIENT_SECRET (*feast.constants.ConfigOptions* attribute), 15
 OAUTH_GRANT_TYPE (*feast.constants.ConfigOptions* attribute), 15
 OAUTH_TOKEN_REQUEST_URL (*feast.constants.ConfigOptions* attribute), 15
 online_store (*feast.feature_table.FeatureTable* property), 10

P

PROJECT (*feast.constants.ConfigOptions* attribute), 15

R

record_format (*feast.data_source.KinesisOptions* property), 5
 REDIS_HOST (*feast.constants.ConfigOptions* attribute), 15
 REDIS_PORT (*feast.constants.ConfigOptions* attribute), 15
 REDIS_SSL (*feast.constants.ConfigOptions* attribute), 15
 region (*feast.data_source.KinesisOptions* property), 5

S

S3_ENDPOINT_URL (*feast.constants.ConfigOptions* attribute), 15
 SERVING_ENABLE_SSL (*feast.constants.ConfigOptions* attribute), 15
 SERVING_SERVER_SSL_CERT (*feast.constants.ConfigOptions* attribute), 15
 SERVING_URL (*feast.constants.ConfigOptions* attribute), 15
 SourceType (class in *feast.data_source*), 6
 SPARK_BQ_MATERIALIZATION_DATASET (*feast.constants.ConfigOptions* attribute), 15
 SPARK_BQ_MATERIALIZATION_PROJECT (*feast.constants.ConfigOptions* attribute), 15
 SPARK_HOME (*feast.constants.ConfigOptions* attribute), 15
 SPARK_INGESTION_JAR (*feast.constants.ConfigOptions* attribute), 15
 SPARK_LAUNCHER (*feast.constants.ConfigOptions* attribute), 16
 SPARK_STAGING_LOCATION (*feast.constants.ConfigOptions* attribute), 16
 SPARK_STANDALONE_MASTER (*feast.constants.ConfigOptions* attribute), 16
 STATSD_ENABLED (*feast.constants.ConfigOptions* attribute), 16
 STATSD_HOST (*feast.constants.ConfigOptions* attribute), 16
 STATSD_PORT (*feast.constants.ConfigOptions* attribute), 16
 STENCIL_URL (*feast.constants.ConfigOptions* attribute), 16
 stream_name (*feast.data_source.KinesisOptions* property), 5
 stream_source (*feast.feature_table.FeatureTable* property), 10

T

table_ref (*feast.data_source.BigQueryOptions* property), 3
 TELEMETRY (*feast.constants.ConfigOptions* attribute), 16
 to_dict() (*feast.entity.Entity* method), 8
 to_dict() (*feast.feature_table.FeatureTable* method), 10
 to_proto() (*feast.data_source.BigQueryOptions* method), 3
 to_proto() (*feast.data_source.BigQuerySource* method), 3
 to_proto() (*feast.data_source.DataSource* method), 4

`to_proto()` (*feast.data_source.FileOptions* method), 4
`to_proto()` (*feast.data_source.FileSource* method), 4
`to_proto()` (*feast.data_source.KafkaOptions* method), 5
`to_proto()` (*feast.data_source.KafkaSource* method), 5
`to_proto()` (*feast.data_source.KinesisOptions* method),
5
`to_proto()` (*feast.data_source.KinesisSource* method),
6
`to_proto()` (*feast.entity.Entity* method), 8
`to_proto()` (*feast.feature.Feature* method), 11
`to_proto()` (*feast.feature.FeatureRef* method), 11
`to_proto()` (*feast.feature_table.FeatureTable* method),
10
`to_spec_proto()` (*feast.entity.Entity* method), 8
`to_spec_proto()` (*feast.feature_table.FeatureTable*
method), 10
`to_yaml()` (*feast.entity.Entity* method), 8
`to_yaml()` (*feast.feature_table.FeatureTable* method),
10
`topic` (*feast.data_source.KafkaOptions* property), 5

V

`value_type` (*feast.entity.Entity* property), 8