db-facts

Release 4.0.0

Vince Broz

Jun 26, 2020

CONTENTS:

CHAPTER

ONE

DB_FACTS PACKAGE

1.1 Submodules

1.2 db_facts.base64_jinja_context module

db_facts.base64_jinja_context.**pull_base64_jinja_context**(*db_name*, *dbcli_config*)

Returns a Jinja context that exports the following functions:

- b64decode(s: str) -> str: Converts a base64ed string to its original contents.
- b64encode(s: str) -> str: Converts a string to its base64ed form.

Return type Tuple[Dict[str, Any], Dict[str, Callable[[Any], Any]]]

1.3 db_facts.env_jinja_context module

db_facts.env_jinja_context.**pull_env_jinja_context**(*db_name*, *dbcli_config*)

Returns a Jinja context that exports the following functions:

- getenv(key: str, default: Optional[str]=None) -> Optional[str]: Return the value of the environment variable key if it exists, or default if it doesn't. key, default and the result are str.
- env(key: str) -> str: Looks up an environment variable value, or raises KeyError if not found.

Return type Dict[str, Any]

1.4 Module contents

class db_facts.**DBFacts**(**args*, ***kwargs*)

Bases: dict

This is a dictionary type which describes the output of the db() method - a dict of various facts about the database in question. All keys are optional except 'type', and keys should only be provided if relevant to the database type.

bq_default_dataset_id: str

BigQuery-specific - the dataset to be used if no specific dataset is specified

bq_default_project_id: str

BigQuery-specific - the project to be used if no specific project is specified

bq_service_account_json: str
BigQuery-specific - JSON (serialized to a string) representing the service account credentials to be used.

database: str
Database name - this concepts varys quite a bit from database to database, but is often used to distinguish between completely separate databases that share the same underlying infrastructure (e.g., same port and host, but different database).

host: str
Database hostname

password: str
Database password

port: int
Database port number

protocol: str
Database protocol type (often the same value as 'type', but may vary for databases like Redshift which offer protocol compatibilities with e.g. postgres)

type: str
Database type (canonical examples: postgres, vertica, mysql, redshift, bigquery)

user: str
Database username

exception db_facts.**UserErrorException**
Bases: Exception

Raised upon an error related to the inputs to the db() function.

db_facts.**db**(*db_name*, *dbcli_config=None*)
Get connection info for specified database.

Parameters **db_name** (List[str]) – Alias for the particular database endpoint and account to connect to. ['a','b','c'] corresponds to 'a-b-c' on the db-facts command-line.

Raises ***UserErrorException*** – Raised if db_name cannot be found.

Return type DBFacts

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

d

INDEX