
Тестовое задание в Честный знак Documentation

Release v1.0.0

ДЫМНИКОВ Михаил

Feb 27, 2025

CONTENTS

1	app	1
2	config	2
3	database	3
4	logger	6

1.1 main

`app.main.find_non_processed_document(document_type: str | None = 'transfer_document') → Documents | None`

Retrieve the earliest non-processed document from the database.

Parameters

`document_type` (Optional[str]) – The type of document to filter by. Defaults to 'transfer_document'.

Returns

The first document that matches the criteria, or None if no document is found.

Return type

Optional[Documents]

`app.main.process_document_objects(document: Documents) → bool`

Processes data objects linked to the given document.

Validates the document's data, updates matching Data objects per operation details, marks the document as processed, and returns the processing status.

Parameters

`document` (Documents) – The document to process.

Returns

True if processing is successful, otherwise False.

Return type

bool

`app.main.process_document() → bool`

Main function to process a single document.

This function retrieves a non-processed document and processes its associated objects. If there is no document to process or an error occurs, it returns False.

Returns

True if a document was processed successfully; False otherwise.

Return type

bool

2.1 variables

2.1.1 Constants

`config.variables.DB_HOST`

`config.variables.DB_NAME`

`config.variables.DB_USER`

`config.variables.DB_PASSWORD`

`config.variables.DB_PORT`

`config.variables.DATABASE_INFO`

3.1 base_model

3.1.1 Constants

database.base_model.T

```
class database.base_model.BaseModel(**kwargs: Any)
```

Bases: Base, Generic[T]

Base model for all database objects with common methods

classmethod get_session()

classmethod get(pk: object) → T | None

Retrieve an instance by primary key

classmethod all() → List[T]

classmethod query_set() → QuerySet[T]

classmethod filter(*args, **kwargs) → QuerySet[T]

Returns a query set for filtering, which can then be chained with .order_by(), .all(), or .first()

classmethod order_by(*args) → QuerySet[T]

Returns a query set with order_by applied, allowing further calls to .all() or .first()

classmethod first(*args, **kwargs) → T | None

Convenient method to immediately retrieve the first object matching the filter

classmethod values(*columns) → List[tuple]

Retrieve selected columns from the database

delete() → None

Delete the current instance from the database

classmethod create(**kwargs) → T

Create a new instance of the model

classmethod add_all(instances: List[T]) → List[T]

Add a list of model instances to the database in a single operation

save() → T

Save the current instance to the database

```
classmethod count(*args, **kwargs) → int
```

Return the count of records matching the filter or the total number of records

```
classmethod update_all(filter_condition, update_values: dict) → int
```

```
class database.base_model.QuerySet(query, session)
```

Bases: `Generic[T]`

```
filter(*args, **kwargs) → QuerySet[T]
```

```
order_by(*args) → QuerySet[T]
```

```
all() → List[T]
```

```
first() → T | None
```

3.2 data_filler

```
database.data_filler.make_data() → dict
```

Генерация случайных данных для таблицы data в базе, вернёт list, внутри dict по каждой записи

```
database.data_filler.make_documents(data: dict) → list
```

Генерация случайных данных для таблицы documents в базе, вернёт list, внутри dict по каждой записи

3.3 database_config

```
class database.database_config.DatabaseConfig(database_info: dict, database_name: str = 'default')
```

Bases: `object`

Database configuration

```
get_engine_url() → str
```

```
get_session() → scoped_session
```

3.4 models

```
database.models.fill_tables(model, data_tbl, documents_tbl)
```

```
database.models.create_tables()
```

```
class database.models.Data(**kwargs)
```

Bases: `BaseModel`

object

status

level

parent

owner

```
class database.models.Documents(**kwargs)
    Bases: BaseModel
    doc_id
    recieved_at
    document_type
    document_data
    processed_at
```

4.1 LOGGER INIT

IMPORT LIBRARIES

4.1.1 logging_functions

`logger.logging_functions.print_error(e: Exception) → None`

Log and display detailed information about an error asynchronously.

Parameters

`e (Exception)` – The exception to log.

`class logger.logging_functions.Logger`

Bases: `object`

A class to configure and manage asynchronous logging.

`get_logger() → Logger`

Retrieve the configured logger instance.

Returns

The logger instance.

Return type

`logging.Logger`

`shutdown()`

Shutdown the queue listener gracefully.