# oerforge.db\_utils

Database utilities for OERForge: initialization, asset tracking, page-file relationships, site metadata, and general-purpose queries for SQLite.

## Overview

oerforge.db\_utils provides a set of functions to manage and interact with the SQLite database used in the OERForge project. It supports schema setup, record insertion and fetching, asset linking, conversion tracking, and pretty-printing for debugging. Designed for contributors and users who need to extend or inspect the database layer.

## Functions

### initialize\_database

def initialize\_database()

Initialize the database schema, create all required tables, and insert default conversion capabilities.

### set\_relative\_link

def set\_relative\_link(content\_id, relative\_link, db\_path=None)

Update the relative\_link for a content item.

**Parameters** - content\_id (int): Content item ID. - relative\_link (str): Relative link to set. - db\_path (str, optional): Path to the database file.

### set\_menu\_context

def set\_menu\_context(content\_id, menu\_context, db\_path=None)

Update the menu\_context for a content item.

**Parameters** - content\_id (int): Content item ID. - menu\_context (str): Menu context value. - db\_path (str, optional): Path to the database file.

### get\_menu\_items

def get\_menu\_items(db\_path=None)

Fetch all menu items with their links and context.

**Parameters** - db\_path (str, optional): Path to the database file.

**Returns** - list[dict]: List of menu item dictionaries.

### get\_db\_connection

def get\_db\_connection(db\_path=None)

Get a SQLite3 connection to the database.

**Parameters** - db\_path (str, optional): Path to the database file.

**Returns** - sqlite3.Connection: Connection object.

### get\_records

def get\_records(table\_name, where\_clause=None, params=None, db\_path=None, conn=None, cursor=None)

Fetch records from a table with optional WHERE clause and parameters.

**Parameters** - table\_name (str): Table name. - where\_clause (str, optional): SQL WHERE clause (without ‘WHERE’). - params (tuple/list, optional): Parameters for WHERE clause. - db\_path (str, optional): Path to the database file. - conn, cursor: Optional existing connection/cursor.

**Returns** - list[dict]: List of records as dictionaries.

### insert\_records

def insert\_records(table\_name, records, db\_path=None, conn=None, cursor=None)

Batch insert records into any table. Returns list of inserted row IDs.

**Parameters** - table\_name (str): Table name. - records (list[dict]): List of column-value dicts. - db\_path (str, optional): Path to the database file. - conn, cursor: Optional existing connection/cursor.

**Returns** - list[int]: List of inserted row IDs.

### get\_enabled\_conversions

def get\_enabled\_conversions(source\_format, db\_path=None)

Get enabled target formats for a given source format.

**Parameters** - source\_format (str): Source file extension (e.g., ‘.md’). - db\_path (str, optional): Path to the database file.

**Returns** - list[str]: List of enabled target formats.

### pretty\_print\_table

def pretty\_print\_table(table\_name, db\_path=None, conn=None, cursor=None)

Pretty-print all rows of a table to the log and terminal for inspection/debugging.

**Parameters** - table\_name (str): Table name. - db\_path (str, optional): Path to the database file. - conn, cursor: Optional existing connection/cursor.

### log\_event

def log\_event(message, level="INFO")

Log an event to both stdout and a log file in the project root.

**Parameters** - message (str): Log message. - level (str): Severity level (“INFO”, “ERROR”, etc.).

### link\_files\_to\_pages

def link\_files\_to\_pages(file\_page\_pairs, db\_path=None, conn=None, cursor=None)

Link files to pages in the pages\_files table.

**Parameters** - file\_page\_pairs (list[tuple]): Each tuple is (file\_id, page\_path). - db\_path (str, optional): Path to the database file. - conn, cursor: Optional existing connection/cursor.

### get\_available\_conversions\_for\_page

def get\_available\_conversions\_for\_page(output\_path, db\_path=None)

Return all successful conversions for a page output path.

**Parameters** - output\_path (str): Output path of the page. - db\_path (str, optional): Path to the database file.

**Returns** - list[dict]: List of dicts with target\_format, output\_path, and status.

## Usage Example

from oerforge import db\_utils
conn = db\_utils.get\_db\_connection()
db\_utils.pretty\_print\_table('content', conn=conn)

## Requirements

* Python 3.7+
* SQLite3

## See Also

* [SQLite Documentation](https://sqlite.org/docs.html)

## License

See LICENSE in the project root.