oerforge.export_all — Export All Content and Assets

Overview

`oerforge.export_all` provides utilities for exporting all site content, assets, and resources from the OERForge project for deployment, backup, or external use. It is designed to help new users and programmers automate the process of collecting and packaging all generated files, static assets, and documentation in a consistent manner.

- **Exports all generated HTML, assets, and documentation**
- **Supports copying to external directories or archives**
- **Ensures directory structure and file integrity**
- **Integrates with build and deployment workflows**
- **Robust logging and error handling**

Functions

export_all_content(output_dir='export', include_assets=True, include_docs=True)

Export all site content, assets, and documentation to a target directory.

Parameters - `output_dir` (str): Target directory for export (default: 'export'). - `include_assets` (bool): Whether to include static assets (CSS, JS, images). - `include_docs` (bool): Whether to include documentation files.

Returns - `None`

Notes - Creates target directories if they do not exist. - Overwrites files in the export directory each time it is called. - Logs all operations and errors for auditing.

---

copy_directory(src, dst)

Recursively copy a directory and its contents to a target location.

Parameters - `src` (str): Source directory path. - `dst` (str): Destination directory path.

Returns - `None`

Notes - Creates destination directory if missing. - Overwrites files in the destination.

---

export_database(db_path, output_dir)

Export the SQLite database file to the export directory.

Parameters - `db_path` (str): Path to the SQLite database file. - `output_dir` (str): Target directory for export.

Returns - `None`

---

export_reports(reports_dir, output_dir)

Export all accessibility and build reports to the export directory.

Parameters - `reports_dir` (str): Source directory for reports. - `output_dir` (str): Target directory for export.

Returns - `None`

---

main()

CLI entry point. Parses arguments and runs export operations as needed.

Logging

All major operations and errors are logged for debugging and auditing.

Error Handling

Robust error handling is implemented for file I/O and directory operations. All failures are logged with context.

Example Usage

```python
from oerforge import export_all
export_all.export_all_content(output_dir='export', include_assets=True, include_docs=True)
```

License

See LICENSE in the project root.