

# oerforge.copyfile — Project File and Asset Copy Utilities

## Overview

`oerforge.copyfile` provides utilities for copying project content, static assets, and build outputs into deployment directories. It is designed for new users and programmers to automate file preparation for static site hosting (e.g., GitHub Pages).

- Copies all content and assets to `build/` and `docs/`
  - Ensures target directories exist
  - Overwrites files to keep outputs up-to-date
  - Creates `.nojekyll` to prevent Jekyll processing on GitHub Pages
  - Robust logging and error handling
- 

## Functions

### `copy_build_to_docs()`

Non-destructively copy everything from `build/` to `docs/`.

**Parameters** - None

**Returns** - None

**Notes** - Creates `docs/` if missing. - Copies files over themselves, does not remove files from `docs/`. - Preserves directory structure.

---

### `ensure_dir(path)`

Ensure that a directory exists, creating it if necessary.

**Parameters** - `path` (str): Directory path to ensure.

**Returns** - None

**Notes** - Logs directory creation for debugging.

---

### `create_nojekyll(path)`

Create an empty `.nojekyll` file at the given path.

**Parameters** - `path` (str): Path to `.nojekyll` file.

**Returns** - None

**Notes** - Used to prevent GitHub Pages from running Jekyll on the build output.  
- Logs file creation.

---

## Constants

- PROJECT\_ROOT: Absolute path to the project root directory.
  - BUILD\_DIR: Path to the build output directory.
  - CONTENT\_SRC, CONTENT\_DST: Source and destination for content files.
  - CSS\_SRC, CSS\_DST: Source and destination for CSS assets.
  - JS\_SRC, JS\_DST: Source and destination for JS assets.
  - NOJEKYL\_PATH: Path to the .nojekyll file in build.
  - LOG\_PATH: Path to the build log file.
- 

## Logging

All major operations and errors are logged for debugging and auditing. Log files are written to `log/build.log`.

## Error Handling

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

## Example Usage

```
from oerforge.copyfile import copy_build_to_docs, ensure_dir, create_nojekyll
copy_build_to_docs()
ensure_dir('build/files')
create_nojekyll('build/.nojekyll')
```

---

## License

See LICENSE in the project root.