

Documentation Automation Script



Automate your documentation workflow! This script helps you copy files to documentation folders, manage versions, and automatically push changes to GitHub - all from the command prompt.

Features

- 📁 **Folder Selection:** Choose from multiple documentation folders
- 🔍 **Duplicate Detection:** Automatically checks if files already exist
- 📦 **Version Management:** Option to overwrite or save as new version with timestamp
- 🚀 **Auto GitHub Push:** Automatically commit and push changes to GitHub
- 💻 **Command Prompt Access:** Run from anywhere using simple commands

Prerequisites

- Python 3.6 or higher
- Git installed and configured
- GitHub repository set up in your documentation folders

Installation & Setup



Step 1: Verify Python Installation

Open Command Prompt and check if Python is installed:

```
python --version
```

If not installed, download from [python.org](https://www.python.org)

Step 2: Verify Git Installation

```
git --version
```

If not installed, download from git-scm.com

Step 3: Add Script to PATH (Optional but Recommended)

To run the script from anywhere in Command Prompt:

1. Open **System Properties** → **Environment Variables**
2. Under **User variables**, find **Path** and click **Edit**
3. Click **New** and add: `d:\All_React_app\Documentry_automation_script`
4. Click **OK** to save
5. **Restart Command Prompt** for changes to take effect

Usage

Basic Usage

Once added to PATH, you can run from anywhere:

```
automate <file_path>
```

Examples

Copy a file from current directory:

```
automate myfile.txt
```

Copy a file with full path:

```
automate "C:\Users\YourName\Desktop\document.pdf"
```

Copy from relative path:

```
automate ..\reports\report.docx
```

Without PATH Setup

If you didn't add to PATH, navigate to the script directory first:

```
cd d:\All_React_app\Documentry_automation_script  
automate myfile.txt
```

Or use the full command:

```
python "d:\All_React_app\Documentry_automation_script\automate_doc.py" myfile.txt
```

Workflow

When you run the script, here's what happens:

1. **File Validation:** Checks if source file exists
2. **Folder Selection:** Shows list of available documentation folders
3. **Choose Destination:** Select folder by number

4. Duplicate Check:

- If file exists → Choose to overwrite or create new version
- If new → File is copied directly

5. GitHub Push (Optional):

- Choose whether to push to GitHub
- Enter custom commit message or use default
- Automatic git add, commit, and push

Example Session

```
=====  
[file] DOCUMENTATION AUTOMATION SCRIPT  
=====
```

```
Source file: my-report.pdf  
=====
```

```
Available Documentation Folders:  
=====
```

1. Project_A_Docs
 2. Project_B_Docs
 3. Research_Notes
- ```
=====
```

```
Enter folder number (or 'q' to quit): 1
=====
```

```
✓ Selected folder: Project_A_Docs
=====
```

```
Copying file...
=====
```

```
✓ File copied successfully: D:\All_React_app\Documentry\Project_A_Docs\my-report.pdf
=====
```

```
Do you want to push to GitHub? (y/n): y
=====
```

```
Enter commit message (or press Enter for default): Added new report
=====
```

```
Pushing to GitHub...
=====
```

```
[file] Adding file: my-report.pdf
[file] Committing changes: Added new report
[rocket] Pushing to remote repository...
[checkmark] Successfully pushed to GitHub!
=====
```

```
[checkmark] Automation completed successfully!
=====
```

## File Versioning

When a file already exists, you get these options:

- **Overwrite:** Replaces the existing file
- **New Version:** Creates a timestamped version
  - Example: `report.pdf` → `report_v20260212_143022.pdf`
- **Cancel:** Exit without making changes

## Directory Structure

```
D:\All_React_app\
├── Documentry_automation_script\ (This script)
│ ├── automate.bat
│ ├── automate_doc.py
│ ├── requirements.txt
│ └── README.md

└── Documentry\ (Your documentation folders)
 ├── Project_A_Docs\
 ├── Project_B_Docs\
 └── Research_Notes\
```

## Git Setup for Documentation Folders

Each documentation folder should be a git repository. To set up:

```
cd "D:\All_React_app\Documentry\YourFolder"
git init
git remote add origin https://github.com/yourusername/your-repo.git
git add .
git commit -m "Initial commit"
git push -u origin main
```

## Troubleshooting

"Source file not found"

- Check if the file path is correct
- Use quotes for paths with spaces: `automate "my file.txt"`

"No documentation folders found"

- Ensure `D:\All_React_app\Documentry` exists
- Create subfolders in the Documentry directory

"Not a git repository"

- Initialize git in your documentation folder
- See "Git Setup for Documentation Folders" above

"Push failed: no upstream branch"

- Run: `git push --set-upstream origin main` (or your branch name)

Command not recognized

- Make sure you added the script directory to PATH
- Restart Command Prompt after editing PATH
- Or use the full python command

## Tips & Best Practices

1. **Keep Documentation Folders Organized:** Each project should have its own folder
2. **Use Descriptive Commit Messages:** Makes it easier to track changes
3. **Regular Pushes:** Don't accumulate too many uncommitted changes
4. **Version Control:** Use the versioning feature for important files
5. **Backup:** Even with git, keep local backups of critical documents

## Customization

To change the base documentation path, edit [automate\\_doc.py](#):

```
Line near the bottom of main() function
documentry_base = r"D:\All_React_app\Documentry" # Change this path
```

## Support

If you encounter issues:

1. Check file paths are correct
2. Verify git is properly configured
3. Ensure Python 3.6+ is installed
4. Check that documentation folders exist

## License

Free to use and modify for personal and commercial projects.

---

## Happy Automating!