

# User Guide: Working with Documents and Publishing to GitHub Pages



Author: **European Environment Agency (EEA)**

Date: **Error! Unknown document property name.**

Version: **Error! Unknown document property name.**

# Index

|   |   |
|---|---|
| 1 Step 1: Create Your Feature Branch.....                     | 1 |
| 1.1 Using GitHub (Easy Option):.....                          | 1 |
| 1.2 Using Command Line (Recommended for frequent users):..... | 1 |
| 2 Working on Documents.....                                   | 2 |
| 2.1 Suggested Editors:.....                                   | 2 |
| 2.2 Create or Edit Documents:.....                            | 2 |
| 2.3 Commit Your Changes (Command Line):.....                  | 2 |
| 3 Publishing for Review (Test Environment).....               | 2 |
| 3.1 Create a Pull Request (PR):.....                          | 2 |
| 3.2 Review and Publish:.....                                  | 2 |
| 4 Final Publishing to Main Environment.....                   | 2 |
| 4.1 Merge to Main:.....                                       | 2 |
| 4.2 Final Review and Approval:.....                           | 2 |
| 5 Versioning.....   | 3 |
| 6 How to Revert to a Previous Version.....                    | 3 |

---

## 1 Step 1: Create Your Feature Branch

You can do this either through GitHub's website or using Git commands in your terminal:

### 1.1 Using GitHub (Easy Option):

1. Open GitHub and navigate to your repository.
2. Select the **test** branch from the dropdown menu at the top left.
3. Click the button labeled **New branch** and name it clearly, for example, feature/new-document.

### 1.2 Using Command Line (Recommended for frequent users):

- Open your terminal or command prompt.
- Navigate to your local repository using:

```
cd path/to/your/repository
```

- Make sure your repository is updated:

```
git checkout test
git pull
```

- Create and switch to your new feature branch:

```
git checkout -b feature/new-document
```

## 2 Working on Documents

### 2.1 Suggested Editors:

- **RStudio:** Easy integration with Quarto and Git.
- **Visual Studio Code:** Great support for Quarto documents.
- **Any plain text editor** you prefer.

### 2.2 Create or Edit Documents:

- Add new documents by creating `.qmd` files.
- Edit existing `.qmd` files directly in your chosen editor.

### 2.3 Commit Your Changes (Command Line):

- Stage your changes:

```
git add .
```

- Commit your changes with a clear message:

```
git commit -m "Added introduction to the Privacy Guide"
```

- Push your changes to GitHub:

```
git push origin feature/new-document
```

## 3 Publishing for Review (Test Environment)

### 3.1 Create a Pull Request (PR):

1. Go to your feature branch in GitHub.
2. Click on **Compare & pull request**.
3. Set the pull request destination to the test branch.
4. Provide a simple description of your changes and click **Create pull request**.

### 3.2 Review and Publish:

- Once your pull request is reviewed and approved by the team, it will be merged into the test branch.
- The merged content will automatically be published to the **test GitHub Pages**.
- View and review your documents by visiting the test pages link provided by your team.

## 4 Final Publishing to Main Environment

After reviewing and ensuring your documents are correct:

### 4.1 Merge to Main:

1. Create another pull request from the test branch to the main branch.
2. Clearly mention that the content is ready for final publishing.

### 4.2 Final Review and Approval:

- Your team reviews and approves the pull request.
- Once merged, the documents are automatically published to the main GitHub Pages for everyone to access.

## 5 Versioning

Every time new changes are published to the main branch, a version number is automatically assigned. You don't need to do anything—this happens behind the scenes to help track document versions clearly.

## 6 How to Revert to a Previous Version

If you need to go back to an earlier version:

1. Go to your repository on GitHub and click on **Tags**.
2. Find the version (tag) you want to revert to.
3. To revert locally using command line:

```
git checkout tags/v1.0
```

Replace v1.0 with your desired version.

4. If you want to make the reverted version the new main version again, create a new feature branch from this tag and follow the usual pull request workflow described above.
-