

Self-Hosted Internal Wiki Guide

Table of Contents

Introduction	2
Why Choose This Setup?	2
Architecture Overview	2
Prerequisites	2
System Requirements	3
Phase 1: Installation	3
Step 1: Create the Project Folder	3
Step 2: Download FileBrowser (The Editor)	3
Step 3: Download Caddy (The Web Server)	3
Phase 2: Configuration	4
Step 1: Create the Caddyfile	4
Advanced Caddyfile Options	4
Step 2: Configure the Firewall	4
Phase 3: Launch	5
Window A: Start the Web Server	5
Window B: Start the Editor	5
Running as Background Services (Optional)	5
Phase 4: Usage	6
Finding Your Server's IP Address	6
Access Points	6
User Management	7
Phase 5: Content Templates	7
Starter Template	7
Additional Templates	8
Meeting Notes Template	8
Project Documentation Template	9
Organizing Your Wiki	10
Recommended Folder Structure	10
Naming Conventions	11
Quick Reference	11
Command Summary	11
Keyboard Shortcuts in FileBrowser Editor	12
Troubleshooting	12
Common Issues	12
Getting Help	12
Security Considerations	12

Recommended Security Practices	13
Enabling HTTPS (Optional)	13
Backup and Recovery	13
Manual Backup.....	13
Automated Backup Script	13
Appendix	14
Glossary	14
Version History.....	14

Introduction

This guide provides complete instructions for setting up an **HTML-Only Internal Wiki** that recreates the "Neocities" experience. You'll edit raw HTML files directly in a browser-based editor, and they are instantly published to your internal network.

Why Choose This Setup?

- **Simplicity** – No database, no complex CMS, just HTML files
- **Speed** – Instant publishing with zero build steps
- **Control** – Full ownership of your content and infrastructure
- **Low Maintenance** – Minimal dependencies and resource usage
- **Learning Friendly** – Great for teams wanting to understand web fundamentals

Architecture Overview

We run two lightweight tools side-by-side on a single computer (the "Server"):

Component	Port	Purpose
FileBrowser (The Kitchen)	8080	Where you log in to create and edit files
Caddy (The Dining Room)	80	Where your team goes to read the website



Think of FileBrowser as your kitchen where you prepare the content, and Caddy as the dining room where guests consume it.

Prerequisites

Before starting, ensure you have:

- A computer that will act as the server (Windows, Mac, or Linux)

- Administrator/sudo access on that computer
- Network connectivity within your organization
- Basic familiarity with the command line/terminal

System Requirements

Requirement	Minimum
RAM	512 MB available
Disk Space	100 MB + content storage
Operating System	Windows 10+, macOS 10.15+, or Linux (Ubuntu 20.04+)
Network	Static IP recommended

Phase 1: Installation

Step 1: Create the Project Folder

Create a folder on your computer named **internal-wiki**. All files will be placed inside this folder.

```
mkdir internal-wiki
cd internal-wiki
```

Step 2: Download FileBrowser (The Editor)

FileBrowser provides a web-based interface for managing and editing files.

Windows (PowerShell)

```
iwr -useb https://raw.githubusercontent.com/filebrowser/get/master/get.ps1 | iex
```

Mac/Linux (Terminal)

```
curl -fsSL https://raw.githubusercontent.com/filebrowser/get/master/get.sh | bash
```



After installation, move the **filebrowser** executable into your **internal-wiki** folder.

Step 3: Download Caddy (The Web Server)

Caddy is a modern web server that's easy to configure and automatically handles many common tasks.

1. Visit [Caddyserver.com](https://caddyserver.com)

2. Select your operating system
3. Download the executable
4. Place it inside your `internal-wiki` folder

Phase 2: Configuration

Step 1: Create the Caddyfile

Inside the `internal-wiki` folder, create a plain text file named `Caddyfile` (with no file extension).

Basic Caddyfile

```
:80 {  
    file_server  
}
```

Advanced Caddyfile Options

For production use, consider these enhancements:

Enhanced Caddyfile with logging and compression

```
:80 {  
    # Enable file serving  
    file_server  
  
    # Enable gzip compression  
    encode gzip  
  
    # Set default index file  
    try_files {path} {path}/ /index.html  
  
    # Access logging  
    log {  
        output file access.log  
        format console  
    }  
}
```

Step 2: Configure the Firewall

You must allow traffic on ports **80** and **8080** for the wiki to be accessible.

▼ Windows

1. Search for "Windows Defender Firewall with Advanced Security"
2. Click **Inbound Rules** → **New Rule**

3. Select **Port** → **Next**
4. Select **TCP** and enter specific local ports: **80, 8080**
5. Select **Allow the connection** → **Next**
6. Check all profiles (Domain, Private, Public) → **Next**
7. Name the rule "Internal Wiki" → **Finish**

▼ macOS

When prompted with "Do you want the application to accept incoming network connections?", click **Allow** for both Caddy and FileBrowser.

▼ Linux (Ubuntu/Debian)

```
sudo ufw allow 80/tcp
sudo ufw allow 8080/tcp
sudo ufw reload
```

Phase 3: Launch

Open two terminal/command prompt windows, both navigated to your **internal-wiki** folder.

Window A: Start the Web Server

```
caddy run
```

You should see output indicating Caddy is running on port 80.

Window B: Start the Editor

```
filebrowser -r . -p 8080 -a 0.0.0.0
```



The **-a 0.0.0.0** flag is **required** to allow other computers on your network to connect to the editor.

Running as Background Services (Optional)

For production use, you may want to run these as background services.

Linux (systemd service example)

```
# /etc/systemd/system/internal-wiki.service
[Unit]
Description=Internal Wiki (Caddy)
```

```
After=network.target
```

```
[Service]
```

```
Type=simple
```

```
WorkingDirectory=/path/to/internal-wiki
```

```
ExecStart=/path/to/internal-wiki/caddy run
```

```
Restart=always
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Phase 4: Usage

Finding Your Server's IP Address

Windows

```
ipconfig
```

Look for the **IPv4 Address** (e.g., **192.168.1.50**).

macOS

```
ifconfig | grep "inet "
```

Linux

```
ip a | grep "inet "  
# or  
hostname -I
```

Access Points

Purpose	URL	Notes
Editor (For content creators)	http://192.168.1.50:8080	Default login: admin / admin
Public Site (For readers)	http://192.168.1.50	Read-only access to published content



Change the default password immediately! Go to Settings → Profile in the FileBrowser interface.

User Management

FileBrowser supports multiple users with different permission levels:

- **Admin** – Full access to all files and settings
- **Editor** – Can create and modify files in designated areas
- **Viewer** – Read-only access (if needed)

To add users:

1. Log in as admin
2. Navigate to **Settings** → **User Management**
3. Click **New User**
4. Set username, password, and permissions

Phase 5: Content Templates

Starter Template

To avoid writing basic HTML boilerplate every time, create a template file.

Create a file named `_template.html` in your wiki root:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Wiki Page</title>
  <style>
    body {
      font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-
serif;
      max-width: 800px;
      margin: 0 auto;
      padding: 20px;
      line-height: 1.6;
      color: #333;
    }
    header {
      border-bottom: 1px solid #ccc;
      padding-bottom: 10px;
      margin-bottom: 20px;
    }
    nav a {
      margin-right: 15px;
```

```

        text-decoration: none;
        color: #007bff;
        font-weight: bold;
    }
    nav a:hover {
        text-decoration: underline;
    }
    h1 { color: #2c3e50; }
    h2 { color: #34495e; border-bottom: 1px solid #eee; padding-bottom: 5px; }
    code { background: #f4f4f4; padding: 2px 6px; border-radius: 3px; }
    pre { background: #f4f4f4; padding: 15px; border-radius: 5px; overflow-x:
auto; }
</style>
</head>
<body>

<header>
    <nav>
        <a href="/"> Home</a>
        <a href="/projects/"> Projects</a>
        <a href="/team/"> Team</a>
        <a href="/docs/"> Docs</a>
    </nav>
</header>

<main>
    <h1>Page Title</h1>
    <p>Start writing your content here...</p>
</main>

<footer style="margin-top: 40px; padding-top: 20px; border-top: 1px solid #eee; color:
#666; font-size: 0.9em;">
    <p>Last updated: <span id="lastmod"></span></p>
    <script>document.getElementById('lastmod').textContent =
document.lastModified;</script>
</footer>

</body>
</html>

```

Additional Templates

Meeting Notes Template

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

<title>Meeting Notes - [Date]</title>
<style>
  body { font-family: sans-serif; max-width: 800px; margin: 0 auto; padding:
20px; line-height: 1.6; }
  .meta { background: #f5f5f5; padding: 15px; border-radius: 5px; margin-bottom:
20px; }
  .action-item { background: #fff3cd; padding: 10px; margin: 10px 0; border-
left: 4px solid #ffc107; }
  .decision { background: #d4edda; padding: 10px; margin: 10px 0; border-left:
4px solid #28a745; }
</style>
</head>
<body>
  <h1>Meeting Notes</h1>

  <div class="meta">
    <p><strong>Date:</strong> [DATE]</p>
    <p><strong>Attendees:</strong> [NAMES]</p>
    <p><strong>Location:</strong> [ROOM/LINK]</p>
  </div>

  <h2>Agenda</h2>
  <ol>
    <li>Topic 1</li>
    <li>Topic 2</li>
  </ol>

  <h2>Discussion</h2>
  <p>Summary of discussion...</p>

  <h2>Decisions</h2>
  <div class="decision">
    <strong>Decision:</strong> Description of decision made
  </div>

  <h2>Action Items</h2>
  <div class="action-item">
    <strong>Task:</strong> Description<br>
    <strong>Owner:</strong> Name<br>
    <strong>Due:</strong> Date
  </div>
</body>
</html>

```

Project Documentation Template

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Project: [Name]</title>
<style>
  body { font-family: sans-serif; max-width: 900px; margin: 0 auto; padding:
20px; line-height: 1.6; }
  .status { display: inline-block; padding: 5px 10px; border-radius: 3px; font-
weight: bold; }
  .status-active { background: #28a745; color: white; }
  .status-planning { background: #007bff; color: white; }
  .status-completed { background: #6c757d; color: white; }
  table { width: 100%; border-collapse: collapse; margin: 20px 0; }
  th, td { border: 1px solid #ddd; padding: 10px; text-align: left; }
  th { background: #f4f4f4; }
</style>
</head>
<body>
  <h1>Project: [Name]</h1>
  <p class="status status-active">Active</p>

  <h2>Overview</h2>
  <p>Brief description of the project...</p>

  <h2>Team</h2>
  <table>
    <tr><th>Role</th><th>Name</th><th>Contact</th></tr>
    <tr><td>Lead</td><td>Name</td><td>email@example.com</td></tr>
  </table>

  <h2>Timeline</h2>
  <table>
    <tr><th>Milestone</th><th>Date</th><th>Status</th></tr>
    <tr><td>Kickoff</td><td>2024-01-01</td><td> Complete</td></tr>
  </table>

  <h2>Resources</h2>
  <ul>
    <li><a href="#">Link to resource</a></li>
  </ul>
</body>
</html>

```

Organizing Your Wiki

Recommended Folder Structure

```

internal-wiki/
├── index.html      # Homepage

```

```

├── _template.html      # Base template (prefix with _ to hide)
├── Caddyfile           # Server config
├── caddy               # Server executable
├── filebrowser         # Editor executable
├── projects/
│   ├── index.html     # Projects overview
│   ├── project-a.html
│   └── project-b.html
├── team/
│   ├── index.html     # Team directory
│   └── onboarding.html
├── docs/
│   ├── index.html     # Documentation home
│   ├── style-guide.html
│   └── processes/
│       └── code-review.html
└── assets/
    ├── css/
    │   └── shared.css
    └── images/
        └── logo.png

```

Naming Conventions

Type	Convention	Example
Pages	lowercase-with-hyphens.html	project-alpha.html
Folders	lowercase-single-word	docs/, team/
Templates	Prefix with underscore	_template.html
Images	descriptive-name.ext	team-photo-2024.jpg

Quick Reference

Command Summary

Goal	Command / Action
Start Editor	filebrowser -r . -p 8080 -a 0.0.0.0
Start Server	caddy run
Edit URL	http://[IP]:8080
View URL	http://[IP]
Reset Password	Delete filebrowser.db file (resets all users)
Find IP (Windows)	ipconfig
Find IP (Mac/Linux)	ifconfig or ip a

Keyboard Shortcuts in FileBrowser Editor

Shortcut	Action
Ctrl + S	Save file
Ctrl + F	Find in file
Ctrl + H	Find and replace
Ctrl + Z	Undo
Ctrl + Shift + Z	Redo

Troubleshooting

Common Issues

1. *Port 80 is already in use*

Another application (like IIS, Apache, or Skype) may be using port 80. Either stop that service or change Caddy to use a different port (e.g., :8000 in your Caddyfile).

2. *Cannot access from other computers*

Check your firewall settings. Ensure ports 80 and 8080 are open. Verify you're using the correct IP address.

3. *FileBrowser shows "connection refused"*

Make sure you included the `-a 0.0.0.0` flag when starting FileBrowser. Without it, only localhost connections are accepted.

4. *Changes not appearing on the website*

Try a hard refresh (Ctrl + Shift + R) in your browser. If using Chrome, you may need to clear cache.

5. *Forgot admin password*

Delete the `filebrowser.db` file in your wiki folder and restart FileBrowser. This resets to the default admin/admin credentials.

Getting Help

- **Caddy Documentation:** <https://caddyserver.com/docs/>
- **FileBrowser Documentation:** <https://filebrowser.org/>

Security Considerations



This setup is designed for **internal network use only**. Do not expose these services to the public internet without additional security measures.

Recommended Security Practices

1. **Change default passwords immediately** after first login
2. **Use HTTPS** if the wiki contains sensitive information (Caddy supports automatic HTTPS)
3. **Restrict network access** to trusted IP ranges if possible
4. **Regular backups** of your wiki content
5. **Keep software updated** – check for updates to Caddy and FileBrowser periodically

Enabling HTTPS (Optional)

For internal HTTPS with a self-signed certificate:

```
https://wiki.internal:443 {  
    tls internal  
    file_server  
}
```

Backup and Recovery

Manual Backup

Simply copy the entire **internal-wiki** folder to your backup location:

```
# Linux/Mac  
cp -r internal-wiki/ /backup/wiki-$(date +%Y%m%d)/  
  
# Windows PowerShell  
Copy-Item -Recurse internal-wiki "C:\Backup\wiki-$(Get-Date -Format 'yyyyMMdd')"
```

Automated Backup Script

backup-wiki.sh (Linux/Mac)

```
#!/bin/bash  
BACKUP_DIR="/backup/wiki"  
SOURCE_DIR="/path/to/internal-wiki"  
DATE=$(date +%Y%m%d_%H%M%S)  
  
mkdir -p "$BACKUP_DIR"  
tar -czf "$BACKUP_DIR/wiki-$DATE.tar.gz" "$SOURCE_DIR"  
  
# Keep only last 7 backups
```

```
ls -t "$BACKUP_DIR"/wiki-*.tar.gz | tail -n +8 | xargs -r rm
```

Appendix

Glossary

FileBrowser

A web-based file manager that provides an interface for creating and editing files.

Caddy

A modern, open-source web server written in Go with automatic HTTPS capabilities.

Caddyfile

The configuration file used by Caddy to define how it serves content.

Static Site

A website consisting of fixed HTML, CSS, and JavaScript files (no server-side processing).

Version History

Version	Date	Changes
1.0	2024	Initial release