

<!DOCTYPE markdown>

端末間でJoplinのデータを同期する

Q.JoplinのデータをWindows・Android・Linux間で共有する方法は？

—

最も安定して Windows・Android・Linux 間で Joplin を共有する方法は、Joplin が公式にサポートする同期サービス（Joplin Cloud / Nextcloud / WebDAV / Dropbox / OneDrive / S3 / ローカルフォルダ）を使う方法です。特に自己ホスト派なら Nextcloud・WebDAV・手軽さなら Dropbox/OneDrive が最適です。

—

・ Joplin を複数 OS 間で同期するベストプラクティス ・Windows・Android・Linux すべて共通）

1. Joplin が公式にサポートする同期方式 Joplin は特定企業に依存しない設計で、以下のサービスを標準サポートしています：

同期方式	特徴	向いている人
-----	---	-----
Joplin Cloud	公式 最も安定 高速	手間なく確実に同期したい
Nextcloud	自己ホスト可能 高機能	自前クラウド派
WebDAV	軽量 多くのサーバで利用可	VPS/レンタルサーバ利用者
Dropbox	手軽 設定が簡単	初心者向け
OneDrive	Windows との相性良い	Microsoft 365 ユーザー
S3	高耐久 大容量	AWS/MinIO ユーザー
ローカルフォルダ	同一 PC 内のみ	USB や NAS での手動同期

—

各デバイスでの設定手順（共通）

■ Windows / Linux（デスクトップ版） 1. Joplin を起動 2. 左下の「設定」→「同期」 3. 同期ターゲットを選択（Dropbox / WebDAV / Nextcloud など） 4. 必要な URL・認証情報を入力 5. 「同期」ボタンを押す

Linux 版は公式スクリプトでインストール可能。

—

■ Android（モバイル版） 1. Joplin Mobile を開く 2. 右上メニュー → 設定 → 同期 3. デスクトップと同じ同期方式を選択 4. 認証情報を入力 5. 同期を実行

Dropbox の場合は OAuth 認証で簡単に接続できます。

—

エンドツーエンド暗号化（E2EE） Joplin は同期サービス側に依存せず、すべてのデータを端末側で暗号化できます。

- 設定 → 暗号化（E2EE）を ON - すべての端末で同じ暗号化キーを入力

あなた向け：最適な同期構成の提案 あなたのワークスタイル(Linux/Windows/Android)自己ホスト志向(Markdown ワークフロー重視)を踏まえると、以下が最も安定します：

Nextcloud または WebDAV(自前サーバ) - 完全に自分で管理できる - Markdown添付ファイルの扱いが安定 - プライバシーと再現性の両立 - Git やバックアップと組み合わせやすい

代替案(手軽さ重視) - Dropbox(設定が最も簡単 - OneDrive(Windows との統合が強い

まとめ - Joplin は Dropbox / OneDrive / Nextcloud / WebDAV / S3 / Joplin Cloudなどで同期可能 - Windows(Linux(Android)すべて同じ同期方式を設定すればよい - 自己ホスト派には Nextcloud / WebDAV が最適 - 手軽さなら Dropbox が最速

必要なら - Nextcloud/WebDAV の構築手順 - Joplin の同期トラブルシューティング - Git 連携による完全再現性ワークフローなども作れます。

どの方式で同期したいですか？

WebDAVサーバを構成する

Q.WebDAVサーバのRaspberryPiでの構成手順を教えてください。

以下に Raspberry Pi で WebDAV サーバを構築する最も安定した手順を、検索結果に基づきつつ、あなたのワークスタイル(再現性 シンプル構成 自己ホスト志向)に合わせてまとめます。

検索結果では主に Apache2 を使った WebDAV 構築手順が詳しく紹介されているため、これをベースにしつつ、構成の意味や注意点も補足します。

Raspberry Pi で WebDAV サーバを構築する手順(Apache2) (参考：ラズパイでの WebDAV 設定手順)

1. 必要パッケージのインストール Raspberry Pi OS(Debian系)なら以下で OK

```
```bash sudo apt update sudo apt install apache2 apache2-utils ```
```

WebDAV モジュールは Apache に同梱されています。

□ 2. WebDAV 用ディレクトリを作成 検索結果では /var/www/html/WebDAV を利用しています。

```
```bash sudo mkdir /var/www/html/WebDAV sudo chown www-data:www-data /var/www/html/WebDAV ```
```

- Apache が書き込めるように所有者を www-data に変更します。

—

□ 3. Basic 認証用のユーザー作成 Apache の htpasswd を使います。

```
```bash cd /etc/apache2 sudo htpasswd -m -c .htpasswd username ```
```

- 初回は -c を付けてファイルを作成 - 2人目以降は -c を付けない

—

□ 4. WebDAV の設定ファイルを作成 検索結果では /etc/apache2/conf-available/basic-auth.conf を作成しています。

例 :

```
```bash sudo nano /etc/apache2/conf-available/webdav.conf ```
```

内容 :

```
``` Alias /webdav /var/www/html/WebDAV
```

```
<Directory /var/www/html/WebDAV>
```

```
 DAV On
 AuthType Basic
 AuthName "WebDAV Restricted"
 AuthUserFile /etc/apache2/.htpasswd
 Require valid-user
```

```
</Directory> ```
```

—

□ 5. Apache モジュールを有効化 WebDAV 関連モジュールを有効化します。

```
```bash sudo a2enmod dav sudo a2enmod dav_fs sudo a2enmod auth_digest sudo a2enconf webdav ```
```

—

□ 6. Apache を再起動 検索結果でも Apache の reload が必要とされています。

```
```bash sudo systemctl restart apache2 ```
```

—

7. 動作確認 ブラウザまたは WebDAV クライアントから :

`http:<RaspberryPiのIPアドレス>/webdav/` 認証が求められれば成功 — Joplin からの接続設定  
Joplin の同期設定で： - 同期方法 WebDAV - URL `http:<RaspberryPiのIP>/webdav/` - ユーザー  
名：htpasswd で作成したユーザー - パスワード：同上

あなた向け：より堅牢な構成の提案 あなたのワークスタイル（再現性 自己ホスト Markdown ワークフロー重視）を踏まえると、以下の改善が有効です：

- 1. HTTPS Let's Encrypt を必ず導入 Basic 認証は平文なので HTTPS が必須。
- 2. データディレクトリを /var/www/html から分離 例 /srv/webdav → バックアップ Git 管理 ZFS/Btrfs などと組み合わせやすい。
- 3. Apache より Caddy の方が簡潔で高速 検索結果では Caddy + WebDAV モジュールの構成が紹介されています。

Caddyfile 例：

```
```:8080 {  
  
    route /webdav/* {  
        basicauth {  
            user JDJhJDEwJG...  
        }  
        webdav {  
            root /srv/webdav  
        }  
    }  
}  
```:
```

Caddy は自動 HTTPS も強力。

参考にした検索結果 - Raspberry Pi の WebDAV 設定手順 - Caddy + WebDAV の構成例

必要なら... - Caddy 版の完全構築手順 - systemd サービス化 - ファイアウォール設定 - 外部公開 Cloudflare Tunnel - Joplin 向け最適化 E2EE 同期衝突対策 )

なども作れます。

どの構成 Apache / Caddy / Nginx で進めたいですか？

## バーチャルドメイン

Q.Apacheでバーチャルドメインを運用している場合は、どのような設定になりますか。

—

ここはしっかり押さえておきたいポイントですね。Apache で VirtualHost (バーチャルドメイン) を運用している場合 WebDAV の設定は “ サイトごとの VirtualHost 内に書く ” のが基本になります。つまり /etc/apache2/sites-available/000-default.conf など に直接書くのではなく、各ドメインの <VirtualHost> ブロックの中に WebDAV の設定を組み込む形です。

あなたのように再現性 構成管理を重視する場合 VirtualHost 単位で設定を完結させるのが最もクリーンです。

—

VirtualHost で WebDAV を動かす構成例 Apache

以下は例として：

- ドメイン notes.example.com - WebDAV パス /webdav - データディレクトリ /srv/webdav - Basic 認証 /etc/apache2/.htpasswd

という構成です。

—

1. VirtualHost 設定ファイル ( 例： /etc/apache2/sites-available/notes.conf )

```
``apache <VirtualHost *:80>
```

```
ServerName notes.example.com
```

```
DocumentRoot は通常のサイト用 (WebDAV とは別)
DocumentRoot /var/www/notes
```

```
WebDAV の公開パス
Alias /webdav /srv/webdav
```

```
<Directory /srv/webdav>
 DAV On
```

```
Options Indexes FollowSymLinks
AllowOverride None
```

```
AuthType Basic
AuthName "WebDAV Restricted"
AuthUserFile /etc/apache2/.htpasswd
Require valid-user
</Directory>
```

```
ErrorLog ${APACHELOGDIR}/notes-error.log
```

CustomLog \${APACHELOGDIR}/notes-access.log combined

</VirtualHost> ```

□ 2. WebDAV ディレクトリの準備

```bash sudo mkdir -p /srv/webdav sudo chown www-data:www-data /srv/webdav ```

□ 3. Basic 認証ユーザーの作成

```bash sudo htpasswd -m -c /etc/apache2/.htpasswd username ```

4. 必要モジュールの有効化

```bash sudo a2enmod dav sudo a2enmod dav\_fs sudo a2enmod auth\_digest ```

□ 5. VirtualHost を有効化

```bash sudo a2ensite notes.conf sudo systemctl reload apache2 ```

（重要□HTTPS を使う場合（推奨）

Let's Encrypt を使うなら：

```bash sudo apt install certbot python3-certbot-apache sudo certbot -apache -d notes.example.com ```

HTTPS 化すると□Basic 認証の平文問題が解消されます。

□ Joplin からの接続設定

- 同期方法□WebDAV - URL□

`https://notes.example.com/webdav/`

- ユーザー名：htpasswd のユーザー - パスワード：同上

あなた向けの補足（構成管理の観点）

あなたのワークスタイルを踏まえると、以下の改善が特に相性が良いです：

- ✓ VirtualHost ごとに WebDAV を分離 → サービス単位で設定が完結し、Git 管理しやすい。
- ✓ /srv/webdav のように DocumentRoot と分離 → バックアップ・ZFS/Btrfs スナップショット・権限管理が明確。
- ✓ HTTPS + Basic 認証 + E2EE Joplin → 通信 保存 同期の三段階で暗号化が成立。

—

必要なら：

- HTTPS + HTTP/2 最適化 - systemd-tmpfiles によるディレクトリ管理 - Fail2ban で WebDAV のブルートフォース対策 - Cloudflare Tunnel で外部公開 - Caddy / Nginx 版の VirtualHost 構成

なども作れます。

次はどこを深掘りしましょうか。

From: <https://kijima.mydns.jp/dokuwiki/> - Kijima's private wiki

Permanent link: <https://kijima.mydns.jp/dokuwiki/doku.php?id=%E7%AB%AF%E6%9C%AB%E9%96%93%E3%81%A7joplin%E3%81%AE%E3%83%87%E3%83%BC%E3%82%BF%E3%82%92%E5%90%8C%E6%9C%9F%E3%81%99%E3%82%BB>

Last update: 2026/07/18 07:26

