# **Netlifyでサブディレクトリブログを作る完全ガイド**

## **目次**

1. [全体像の理解](https://claude.ai/chat/968af860-1aa4-420b-9fe3-043b34ef2ed5#%E5%85%A8%E4%BD%93%E5%83%8F%E3%81%AE%E7%90%86%E8%A7%A3)
2. [準備するもの](https://claude.ai/chat/968af860-1aa4-420b-9fe3-043b34ef2ed5#%E6%BA%96%E5%82%99%E3%81%99%E3%82%8B%E3%82%82%E3%81%AE)
3. [ステップ1: プロジェクト構造の決定](https://claude.ai/chat/968af860-1aa4-420b-9fe3-043b34ef2ed5#%E3%82%B9%E3%83%86%E3%83%83%E3%83%971-%E3%83%97%E3%83%AD%E3%82%B8%E3%82%A7%E3%82%AF%E3%83%88%E6%A7%8B%E9%80%A0%E3%81%AE%E6%B1%BA%E5%AE%9A)
4. [ステップ2: ブログサイトの作成](https://claude.ai/chat/968af860-1aa4-420b-9fe3-043b34ef2ed5#%E3%82%B9%E3%83%86%E3%83%83%E3%83%972-%E3%83%96%E3%83%AD%E3%82%B0%E3%82%B5%E3%82%A4%E3%83%88%E3%81%AE%E4%BD%9C%E6%88%90)
5. [ステップ3: Netlifyへのデプロイ](https://claude.ai/chat/968af860-1aa4-420b-9fe3-043b34ef2ed5#%E3%82%B9%E3%83%86%E3%83%83%E3%83%973-netlify%E3%81%B8%E3%81%AE%E3%83%87%E3%83%97%E3%83%AD%E3%82%A4)
6. [ステップ4: サブディレクトリの設定](https://claude.ai/chat/968af860-1aa4-420b-9fe3-043b34ef2ed5#%E3%82%B9%E3%83%86%E3%83%83%E3%83%974-%E3%82%B5%E3%83%96%E3%83%87%E3%82%A3%E3%83%AC%E3%82%AF%E3%83%88%E3%83%AA%E3%81%AE%E8%A8%AD%E5%AE%9A)
7. [ステップ5: SEO設定](https://claude.ai/chat/968af860-1aa4-420b-9fe3-043b34ef2ed5#%E3%82%B9%E3%83%86%E3%83%83%E3%83%975-seo%E8%A8%AD%E5%AE%9A)
8. [ステップ6: コンテンツ管理](https://claude.ai/chat/968af860-1aa4-420b-9fe3-043b34ef2ed5#%E3%82%B9%E3%83%86%E3%83%83%E3%83%976-%E3%82%B3%E3%83%B3%E3%83%86%E3%83%B3%E3%83%84%E7%AE%A1%E7%90%86)

## **全体像の理解**

### **完成イメージ**

taskmateai.net/ → メインサイト（既存）

taskmateai.net/blog/ → ブログ（新規作成）

### **実現方法（2つの選択肢）**

**方法A: 別々のNetlifyプロジェクトとして管理（おすすめ）**

* メインサイトとブログを独立して管理
* リダイレクトで連携
* 管理がシンプル

**方法B: 1つのプロジェクトで管理**

* すべてのコードを1つのリポジトリに
* やや複雑だが一元管理

**→ このガイドでは初心者向けに「方法A」で進めます**

## **準備するもの**

### **必要なアカウント・ツール**

* ✅ Netlifyアカウント（既に持っている）
* ✅ GitHubアカウント
* ✅ Node.js（バージョン18以上）
* ✅ テキストエディタ（VS Codeがおすすめ）

### **確認方法**

ターミナル（コマンドプロンプト）で確認:

node --version

# v18.0.0 以上ならOK

## **ステップ1: プロジェクト構造の決定**

### **今回の構成**

既存: taskmateai.net（メインサイト）

新規: blog用の新しいNetlifyプロジェクト

## **ステップ2: ブログサイトの作成**

### **2-1. Next.jsプロジェクトの作成**

ターミナルを開いて、以下を実行:

# 作業フォルダに移動（例）

cd Desktop

# Next.jsプロジェクトを作成

npx create-next-app@latest taskmateai-blog

質問が表示されるので、以下のように答える:

✔ Would you like to use TypeScript? … Yes

✔ Would you like to use ESLint? … Yes

✔ Would you like to use Tailwind CSS? … Yes

✔ Would you like to use `src/` directory? … Yes

✔ Would you like to use App Router? … Yes

✔ Would you like to customize the default import alias? … No

### **2-2. プロジェクトに移動**

cd taskmateai-blog

### **2-3. 基本的なブログ構造を作成**

#### **src/app/page.tsxを編集**

export default function Home() {

return (

<main className="min-h-screen p-8">

<div className="max-w-4xl mx-auto">

<h1 className="text-4xl font-bold mb-8">TaskMateAI ブログ</h1>

<p className="text-gray-600 mb-8">

タスク管理とAIに関する最新情報をお届けします

</p>

<div className="grid gap-6">

<article className="border rounded-lg p-6 hover:shadow-lg transition">

<h2 className="text-2xl font-semibold mb-2">

サンプル記事1: AIでタスク管理を効率化

</h2>

<p className="text-gray-600 mb-4">

最新のAI技術を使ってタスク管理を10倍効率化する方法

</p>

<span className="text-sm text-gray-500">2025年10月10日</span>

</article>

<article className="border rounded-lg p-6 hover:shadow-lg transition">

<h2 className="text-2xl font-semibold mb-2">

サンプル記事2: 生産性向上のコツ

</h2>

<p className="text-gray-600 mb-4">

毎日の生産性を向上させる5つの習慣

</p>

<span className="text-sm text-gray-500">2025年10月9日</span>

</article>

</div>

</div>

</main>

);

}

#### **src/app/layout.tsxのメタデータを編集**

import type { Metadata } from "next";

import "./globals.css";

export const metadata: Metadata = {

title: "TaskMateAI ブログ - AIとタスク管理の情報メディア",

description: "タスク管理の効率化とAI活用に関する最新情報をお届けするブログです",

};

export default function RootLayout({

children,

}: {

children: React.ReactNode;

}) {

return (

<html lang="ja">

<body>{children}</body>

</html>

);

}

### **2-4. ローカルで確認**

npm run dev

ブラウザで http://localhost:3000 を開いて確認

## **ステップ3: Netlifyへのデプロイ**

### **3-1. GitHubリポジトリの作成**

1. GitHub（https://github.com）にログイン
2. 右上の「+」→「New repository」
3. リポジトリ名: taskmateai-blog
4. Publicを選択
5. 「Create repository」をクリック

### **3-2. コードをGitHubにプッシュ**

ターミナルで実行:

# Gitの初期化

git init

git add .

git commit -m "Initial commit"

# GitHubと連携（URLは自分のものに変更）

git branch -M main

git remote add origin https://github.com/あなたのユーザー名/taskmateai-blog.git

git push -u origin main

### **3-3. Netlifyで新規サイトを作成**

1. Netlifyダッシュボード（https://app.netlify.com）にログイン
2. 「Add new site」→「Import an existing project」
3. 「GitHub」を選択
4. taskmateai-blogリポジトリを選択
5. ビルド設定:
   * **Build command**: npm run build
   * **Publish directory**: .next
6. 「Deploy site」をクリック

### **3-4. サイト名の変更（オプション）**

1. 「Site settings」→「General」→「Site details」
2. 「Change site name」
3. 例: taskmateai-blogに変更
4. これで taskmateai-blog.netlify.app でアクセス可能に

## **ステップ4: サブディレクトリの設定**

### **重要: メインサイトとブログを連携させる**

### **4-1. メインサイト（taskmateai.net）の設定**

メインサイトのプロジェクトで、netlify.tomlファイルを作成または編集:

# netlify.toml（メインサイトのルートに配置）

[[redirects]]

from = "/blog"

to = "https://taskmateai-blog.netlify.app"

status = 200

force = true

[[redirects]]

from = "/blog/\*"

to = "https://taskmateai-blog.netlify.app/:splat"

status = 200

force = true

### **4-2. ブログサイトの設定**

ブログプロジェクト（taskmateai-blog）で、next.config.jsを編集:

/\*\* @type {import('next').NextConfig} \*/

const nextConfig = {

basePath: '/blog',

assetPrefix: '/blog',

trailingSlash: true,

}

module.exports = nextConfig

### **4-3. ブログのリンクを修正**

src/app/page.tsxのリンクを修正:

// 内部リンクを作る場合

<Link href="/blog/article-1">記事を読む</Link>

### **4-4. 変更をデプロイ**

git add .

git commit -m "Add basePath configuration"

git push

## **ステップ5: SEO設定**

### **5-1. サイトマップの自動生成**

src/app/sitemap.tsを作成:

import { MetadataRoute } from 'next'

export default function sitemap(): MetadataRoute.Sitemap {

return [

{

url: 'https://taskmateai.net/blog',

lastModified: new Date(),

changeFrequency: 'daily',

priority: 1,

},

{

url: 'https://taskmateai.net/blog/article-1',

lastModified: new Date(),

changeFrequency: 'weekly',

priority: 0.8,

},

]

}

### **5-2. robots.txtの作成**

src/app/robots.tsを作成:

import { MetadataRoute } from 'next'

export default function robots(): MetadataRoute.Robots {

return {

rules: {

userAgent: '\*',

allow: '/',

},

sitemap: 'https://taskmateai.net/blog/sitemap.xml',

}

}

### **5-3. 構造化データの追加**

src/app/layout.tsxに追加:

export default function RootLayout({

children,

}: {

children: React.ReactNode;

}) {

const jsonLd = {

'@context': 'https://schema.org',

'@type': 'Blog',

name: 'TaskMateAI ブログ',

description: 'タスク管理とAIに関する情報メディア',

url: 'https://taskmateai.net/blog',

};

return (

<html lang="ja">

<head>

<script

type="application/ld+json"

dangerouslySetInnerHTML={{ \_\_html: JSON.stringify(jsonLd) }}

/>

</head>

<body>{children}</body>

</html>

);

}

## **ステップ6: コンテンツ管理**

### **方法1: マークダウンファイルで管理（シンプル）**

#### **6-1. 記事フォルダの作成**

mkdir -p content/posts

#### **6-2. 記事ファイルの作成**

content/posts/first-post.md:

---

title: "AIでタスク管理を効率化する方法"

date: "2025-10-10"

description: "最新のAI技術を使ってタスク管理を10倍効率化"

---

# AIでタスク管理を効率化する方法

この記事では、AIを活用してタスク管理を効率化する具体的な方法を紹介します。

## 1. AIによる優先順位付け

AIが自動的にタスクの優先順位を判断...

#### **6-3. マークダウンを読み込むライブラリをインストール**

npm install gray-matter remark remark-html

### **方法2: HeadlessCMS（本格的）**

**Contentfulの場合:**

1. Contentful（https://www.contentful.com/）にサインアップ
2. 新しいSpaceを作成
3. Content modelで「Blog Post」を作成
4. フィールド: title, slug, content, publishDate
5. API keyを取得

**環境変数の設定:**

# .env.local ファイルを作成

CONTENTFUL\_SPACE\_ID=your\_space\_id

CONTENTFUL\_ACCESS\_TOKEN=your\_access\_token

## **次のステップ**

### **やるべきこと**

1. ✅ Google Search Consoleにサイトを登録
2. ✅ Google Analyticsを設定
3. ✅ 定期的に記事を投稿（週1-2本が理想）
4. ✅ メインサイトからブログへのリンクを追加

### **参考リソース**

* Next.js公式ドキュメント: https://nextjs.org/docs
* Netlify公式ドキュメント: https://docs.netlify.com/
* SEO基礎: https://developers.google.com/search/docs

## **トラブルシューティング**

### **Q: /blogにアクセスしてもブログが表示されない**

**A**: メインサイトのnetlify.tomlの設定を確認。リダイレクトが正しく設定されているか確認してください。

### **Q: ビルドエラーが出る**

**A**: npm run buildをローカルで実行して、エラー内容を確認してください。

### **Q: スタイルが崩れる**

**A**: basePathの設定を確認。画像やCSSのパスが正しいか確認してください。

## **まとめ**

これでtaskmateai.net/blogでブログが公開されました！

SEOで成果を出すには:

* **継続的な更新**が最重要
* **質の高いコンテンツ**を心がける
* **内部リンク**でメインサイトと連携
* **データ分析**で改善を続ける

頑張ってください！🚀