

# T3 Stack + Supabaseで アプリ作ってみる



Press Space for next page →





# 自己紹介

- 📝 飯野陽平 ([wheatandcat](https://www.wheatandcat.me/))
- 🏢 法人設立 ([合同会社 UNICORN](#) 代表社員)
- 💻 Work: シェアフル株式会社 CTO
- 📖 Blog: <https://www.wheatandcat.me/>
- 🛠️ 今までに作ったもの
  - [memoir](#)
  - [ペペロミア](#)
  - [MarkyLinky](#)
  - [Atomic Design Check List](#)

# T3 Stackとは？

- 最近、話題の Web アプリ開発のアーキテクチャ
- 以下の 3 つの思想に集点を当てて設計された技術スタック
  - simplicity(簡潔さ)
  - modularity(モジュール性)
  - full-stack typesafety(フルスタックの型安全)
- [t3-app](#)としてコマンドラインツールも公開されている
- 採用されている 6 つの技術については紹介

# Next.js

- [Next.js - The React Framework](#)
- React をベースとしたフルスタックフロントエンドフレームワーク
- 大規模な Web アプリケーションでは採用されていることが多い

# tRPC

- [tRPC](#)
- スキーマやコード生成なしで型安全な API のレスポンスを提供することができるライブラリ
- 以前、紹介のスライドを作成したので詳しくは、以下を参照
  - [tRPC の紹介 - Speaker Deck](#)

# Tailwind CSS

- [Tailwind CSS](#)
- 最近流行りの CSS フレームワーク
- ユーティリティファーストの思想で設計されており、汎用性が高く、自身で css を作成するコストが大幅に軽減される

# TypeScript

- [TypeScript: JavaScript With Syntax For Types.](#)
- JavaScript を拡張して開発された言語
- 型安全を保ちながら開発を行うことができる

# Prisma

- [Prisma | Next-generation ORM for Node.js & TypeScript](#)
- Node.js 製の ORM
- データベースのスキーマ定義を行うことで、型安全なクエリを実行できる
- 以前、紹介のスライドを作成したので詳しくは、以下を参照
  - [Prisma を試してみた - Speaker Deck](#)



# NextAuth.js

- [NextAuth.js](#)
- Next.js をベースとした認証ライブラリ
- 各サービスのログインに対応
- Next.js でフロントエンド、サーバーサイドの API の認証を一元管理できる

# T3 Stack を使ってサービスを作ってみた

- 使用技術
  - T3 Stack
    - ログインは [Discoed](#) で実装
  - [Supabase](#)
    - 以前、紹介のスライドを作成したので詳しくは、以下を参照
      - [Supabase の紹介 - Speaker Deck](#)
  - [vercel](#)
  - [Figma](#)
- リポジトリ
  - <https://github.com/wheatandcat/OOMAKA>

# 作ったサービス



- サービス名: OOMAKA
- URL: <https://oomaka.vercel.app>
- 概要: 大まかに未来の予定を可視化するサービス

# 実装の紹介

- [デモURL](#)
- ※以下、デモしながら紹介
- DB設計はPrismaで[prisma/schema.prisma](#)の通りに定義
- Next.jsでAPI実装は[API Routes](#)を使用
- tRPC + Prismaは[src/server/api/routers/url.ts](#)のように実装
- フロントエンド側の実装は以下の通り
  - 読み込みは[src/pages/index.tsx](#)のようにして取得
  - 書き込みは[src/pages/index.tsx](#)のようにして取得
- ログインは[src/pages/index.tsx](#)で、サクッと実装できる

# まとめ

- T3 Stack の開発体験はとても良い
- 今後の Web 開発のスタンダードになる可能性が高い
- データ通信の簡単な部分は tRPC と Prisma で作成して、複雑な部分は gRPC でマイクロサービスに繋いで開発すると、かなり開発効率の高いプロダクトにできそう

**ご清聴ありがとうございました 🎉**