

# 職務経歴書

作成日: 2025年11月24日

## 基本情報

項目	内容
氏名	坂田 徳明
生年月日	2000年06月14日 (満25歳)
連絡先	<a href="mailto:led06140@gmail.com">led06140@gmail.com</a>

## 職務要約

Flutterを中心としたモバイルアプリ開発に3年半の実務経験を持つエンジニアです。学生インターン時代からNFTマーケットプレイス、ライブ配信、住宅見学アプリなど複数のプロジェクトに参画し、現在はケーブルTVオンデマンド配信アプリのリード開発を担当しています。

要件定義から設計、コーディング、テスト、運用/保守まで幅広い工程を経験し、特に課題解決能力に強みがあります。主にアプリ開発をやっていますが、Vue.jsを用いたWebフロント開発、Firebaseを活用したバックエンド構築など、他の領域にも積極的に挑戦しています。

技術的負債の削減や開発効率化を重視し、リファクタリングや生成AIを活用したテストコード自動生成など、チーム全体の生産性向上に貢献してきました。

## 職務経歴

### 2023/07 - 現在 : 株式会社セイロップ、ケーブルTVオンデマンド配信アプリのマルチデバイス開発・機能追加・保守

#### プロジェクト概要

ケーブルTVのオンデマンド配信アプリ（4デバイス）のリード開発を担当。動画広告・H.265再生など難易度の高い機能の実装、共通化アーキテクチャの整備、テスト自動化を主導。技術基盤からUI/UXまで広範囲を改善し、安定稼働と開発効率向上に貢献しました。

#### 役割・体制

##### 体制

- アプリ/Webフロント：2名
- インフラ/バックエンド：3名

- デザイナー：1名（初期）
- テスト：2名
- PdM：2名、PM：1名

スクラム（2週間スプリント）

↳ リファインメント / プランニング / デイリースクラム / レビュー / レトロスペクティブ

## 担当工程

- ☐ 企画
- ☒ 要件定義
- ☒ 設計
- ☒ コーディング
- ☒ テスト
- ☒ 運用/保守
- ☐ データ分析
- ☐ その他

## 自身の対応範囲

- アプリ開発（Flutter）
  - モバイル / Android TV / Fire TV / 限定配布版の4デバイスを同時開発
  - 認証（ログイン / 新規登録 / アカウント設定）
  - マイリスト（お気に入り / いいね / 視聴履歴）
  - 動画プレイヤー（IMA SDK広告 / H.265対応 / ネイティブコード拡張）
  - ディープリンク対応
  - 既存画面のUI改善、アンケート/お知らせ機能
- 品質改善 / アーキテクチャ整備
  - 密結合化した既存ソースの分離とリファクタリングを主導
  - provider / repository / apiクライアントを分離し、テスト容易性を改善
  - ユニットテスト基盤構築（カバレッジ90%超を維持）
  - インテグレーションテスト導入（デグレの早期検知）
  - 生成AIによるテストコード自動生成の仕組み化
- Webフロント開発（Vue.js）
  - odooベースの管理画面をVue.jsに刷新（約10画面）
  - 画面設計 / UI実装 / API再連携
  - 課金コンテンツ購入サイトの作成
  - アプリディープリンク用の中間サイト実装
- 環境整備
  - Firebase App Distribution + Fastlane によるテスト配布の自動化
  - GitFlowベースのリリース運用を整備

- 事業者別の限定版向けビルド環境構築（Flutter Flavor）
  - Flutterバージョン更新の定期運用
- 

## 課題・解決

### 課題1

テスターがコマンド実行でAPKを手動インストールしており、テストメンバー入れ替え時の引継ぎコストが高かった。

#### 解決

- Firebase App Distributionを導入 → テスターが専用アプリから即インストール可能に
- Fastlaneでビルド～配布を自動化 → 配布コストを大幅削減

**結果** 配布作業が「15分 → 数十秒」に短縮し、テストサイクルが高速化

### 課題2

既存コードは状態管理とAPI連携が1つのファイルに密結合しており、テスト困難・デグレ頻発・機能追加時の影響範囲が不明瞭という問題があった。

#### 解決

- スパゲッティ化したコードを APIクライアント / Repository / Provider の3層構造へ再構成し、責務を分離
- 依存関係逆転を取り入れ、Mockito によるモックが容易なテストブルな設計へ変更
- ユニットテスト / インテグレーションテストを導入し、回帰テストを自動化
- AI（生成AI）でテストコードを自動生成できるように、参照ドキュメントと .claude.md を整備し、工数を大幅削減
  - └ 人手でテストコードを書く費用対効果が低いため、生成AIで再現性のあるテスト生成フローを確立
- ユニットテストのカバレッジは常時 90%以上を維持

**結果** テストフェーズのデグレ件数を大幅削減し、実装時の手戻りを抑制。開発効率と品質の両立を実現した。

### 課題3

IMA SDK（広告）、動画プレイヤー周辺のネイティブコード、依存パッケージ、iOS／Android OSアップデートに起因する不具合が定期的に発生。しかし更新管理の仕組みがなく、問題発生たびに慌ててバージョンアップ対応する非効率な運用となっていた。

#### 解決

- Flutter本体・主要パッケージのアップデートを3ヶ月ごとに実施する方針を策定し、手順書を整備

- 自由課題時間を活用し、自動で最新バージョンを取得しSlack通知するBotを構築
- Flutter SDK／主要パッケージ／Android & iOS関連の更新を毎週チェックし通知
- 日常的にアップデート情報を共有し、技術的負債を蓄積させない体制へ移行

## 成果

- 大規模なバージョンアップ対応が減り、突発障害対応の工数を大幅削減
- OS・SDKアップデートに伴うリスクを早期検知できるようになり、安定稼働に貢献
- 開発チーム全体が最新のFlutterエコシステムを把握しやすくなり、技術負債の削減につながった

## 課題4

プランニングでスプリントバックログは決まるものの、タスク粒度の不一致や影響範囲の認識違いが発生しており、日次の進捗共有も浅かったため、スプリント中盤に手戻りが生じやすい状態だった。

## 解決

- プランニング後にアプリチームだけの追加ミーティングを実施し、タスク詳細化・サブタスク分割・影響範囲の明確化を行う運用を導入
- デイリースクラム前に15分の「アプリデイリー」を設定し、毎日のアウトプット・懸念点を即共有
- 障害や問題はその日のデイリースクラムで即報告できるようにし、他チームを含めた調整の早期化を実現

## 成果

- タスク認識のズレが減り、スプリント序盤から迷いなく着手できる状態を確立
- 問題の早期発見によりスプリント内で柔軟な巻き返しが可能に
- 手戻りや遅延が減り、スプリント完了率が安定
- チーム内の情報共有が活性化し、コミュニケーション品質が向上

## 課題5

事業会社ごとの限定版アプリをプリインストール端末に配布する必要があり、端末ごとにログインや設定を変える必要があった。

さらに、テスト／ステージング／プロダクションの各サン環境を事業会社ごとに分離する必要があった。

## 解決

- FlutterのFlavorを事業会社ごとに定義し、環境ごとの設定を分離
- Makeファイルを用意し、単一コマンドで特定事業会社かつテスト／ステージング／プロダクション環境のAPKをビルド可能に

## 成果

- チーム全体で共通化・属人化の予防（誰でも最新のソースから生成可能）
  - 限定版アプリの管理・運用が容易になり、チームの作業負担を軽減
-

# 2023/05 - 2023/06 : 株式会社セイロップ、市の行政サービス情報を集約したスマホアプリの開発

## プロジェクト概要

市の行政サービス情報を集約したスマホアプリの開発

## 役割・体制

### 自身のポジションと役割

- アプリ開発（iOSおよびAndroid）担当として、Flutterを活用したコーディングを中心に実施しました。
- ウォーターフォール開発の流れの中で、あらかじめ設計された資料ベースにスケジュール通りに進行させる必要がありました。

### チーム規模と構成

- アプリエンジニア3名、バックエンドエンジニア2名、プロジェクトマネージャ（PM）1名の合計6名のチームで、私はアプリチームのリーダー、サブリーダーのもとで開発を担当しました。

### 担当工程

- ☐ 企画
- ☐ 要件定義
- ☐ 設計
- ☒ コーディング
- ☒ テスト
- ☐ 運用/保守
- ☐ データ分析
- ☐ その他

### 背景・課題

- ウォーターフォール開発のため、仕様変更や修正があった場合の調整が難しく、納期遅延のリスクが存在していました。
- アトミックデザインを採用した設計方針に基づき、UIコンポーネントの一貫性と再利用性を高める必要があり、チーム内での設計規則の徹底と実装の効率化が課題でした。
- ユニットテストコードを書きながらの開発は初めてだったので、品質が担保されるコーディングが必要でした。
- アトミックデザインは初めての経験だったので、思想の理解とキャッチアップが必要でした。

### 実際の取り組み・工夫

- 設計書に基づいた実装では、細かい仕様の確認が必要であり、出戻りのリスクを減らすためデイリーミーティングやチャットで簡潔に質問することを心がけました。
- アトミックデザインの原則に従ったUIコンポーネントの実装において、既存ソースコードを見るだけでなく、自身でも勉強を行い理解を深めました。
- テストコードは仕様書と照らし合わせながら間違った動作をしていないか確認しつつ、実装コードが冗長化していないかも同時に確認しながらコーディングしました。
- 実装で詰まってしまい、スケジュール通りに進まない場合に開発リーダーやPMに状況を簡潔に説明するなど自身のアウトプットを細かく報告しました。

## 成果・価値

- 実装で詰まってしまい、スケジュール通りに進まない場合も問題を細かく報告することで、開発リーダーやPMが状況を理解し、意思決定を素早く行えました。
- チーム内のコミュニケーションとデモ共有により、タスクの重複や認識違いを防ぎ、効率的に作業を進めることができました。
- アトミックデザインやテストコードの実装により、整理され品質が担保されたソースコードが実現できました。

---

# 2022/12 - 2023/04 : 株式会社セイロップ、住宅見学アプリの新規開発

## プロジェクト概要

Flutterを用いた住宅見学アプリの設計とコーディングを担当し、少人数体制で迅速にリリースに向けて開発を推進しました。

## 役割・体制

### 自身のポジションと役割（学生インターン）

- Flutterを用いたアーキテクチャ設計とUI実装、API連携
- アプリ側の実装スケジューリング
- デザイナーとのUIUX設計

### チーム規模と構成

- PM兼バックエンドエンジニア1名、アプリエンジニア1名、デザイナー1名の合計3名の少人数チーム体制で開発を進めました。

## 担当工程

- ☐ 企画
- ☐ 要件定義
- ☒ 設計

- ☒ コーディング
- ☐ テスト
- ☐ 運用/保守
- ☐ データ分析
- ☐ その他

## 背景・課題

- 新規に開発される住宅見学アプリは、少人数の開発体制で迅速にリリースを行う必要があり、住宅見学をスムーズに行えるようにすることが最大の課題でした。
- さらに、アプリのUI改善や動線設計においても、少人数の中でどのように効率的にアイデアを提案し、実現していくかが重要な課題でした。

## 実際の取り組み

### 開発

- API連携はサーバー側がSwagger仕様書を提供していたため、その仕様に基づいてFlutter側での接続部分を設計・実装しました。
- UI/UX検討フェーズでは、ユーザビリティを重視し、デザイナーと密に連携しながらプロトタイプを作成し、『ユーザーの操作感や動線を意識した画面提案』を積極的に行いました。

### 工夫

- UI/UXの品質向上を目的に、疑問点や改善点を積極的に提案し、実装に反映させました。
- 特に、ユーザーの動線や画面遷移について、「この操作をもっとスムーズにできる」といった改善案を提案し、それが採用されました。
- さらに、状態管理は複数画面での一貫性が必要と判断し、Flutterの状態管理（Riverpod+Freezed）を活用し、動作の安定性とパフォーマンスを意識した設計を行いました。

### 成果・価値

- チームの少人数体制にも関わらず、私自身はリリースまでは参画できなかったもの、主要な機能を一通り実装し、品質も一定水準を満たすことができました。
- さらに、実務経験を通じて、「UI/UXの改善点を提案し、実現に導くスキル」や、「少人数チームでの効率的な開発・運用」についての知見を深め、今後のプロジェクトに活かせる経験を積むことができました。

---

## 2022/10 - 2023/01 : 可茂IT塾合同会社、ライブ配信アプリの新規開発

### プロジェクト概要

ライブ配信アプリの新規開発を短期間で実施し、主要機能の実装とリリースを成功させました。

## 役割・体制

### 自身のポジションと役割（学生インターン）

- 開発のリーダーとして、Flutterを用いたアプリ開発とFirebaseによるNoSQLデータ設計とバックエンドの同時開発を担当しました。

### チーム規模と構成

- 3名のエンジニアと1名のデザイナー、1名のプロジェクトマネージャーからなるチーム体制の中で、私が開発リーダーとして、技術面の調整とタスク管理を主導しました。

### 担当工程

- ☐ 企画
- ☐ 要件定義
- ☒ 設計
- ☒ コーディング
- ☐ テスト
- ☐ 運用/保守
- ☐ データ分析
- ☐ その他

### 背景・課題

- 既存のライブ配信技術の調査と導入が最大の課題であり、高品質かつ低遅延な配信を実現するための技術選定と、その運用体制の構築に苦慮しました。
- プロジェクトは2022年頃の短期間（3ヶ月以内）で主要機能の実装を完了させる必要がありました。
- また、インターン生を中心とした開発体制のため、『技術的な指導とタスクの効率的な分配』も重要な課題でした。
- さらに、ライブ配信の安定性と品質を確保しつつ、ユーザビリティを損なわない設計と実装が求められ、技術的な難易度が高い状況でした。

## 実際の取り組み

### 開発

- Flutterを始めて触るメンバーもいたため、あらかじめ似たような画面や設計箇所を実装しておいて、既存実装を模範して実装できるように進めました。
- Firebaseをリアルタイム通信とデータ同期の基盤として採用し、ライブ配信中のチャット機能の実装を効率化しました。
- ライブ配信ツールの導入にあたっては、何度もデモを行い、技術的な実現可能性と運用上の課題を洗い出し、最適なソリューションとしてAgora SDKを利用しました。
- UIデザインをもとに使用するテキストスタイルや色などは静的に管理し、開発メンバーが統一した実装ができるように設計しました。



- NoSQLデータベースなので、できるだけアプリ側で取得しやすい形でデータを保存する必要がありました。コマンドクエリ分離原則（CQS）を採用することで、入力されたデータをバッチ処理でアプリ取得用に整形し、アプリ連携との効率化を図りました。

## 工夫

- 基本リモートでの開発なので、稼働中はバーチャルオフィスを活用し、気軽にチームメンバーと会話ができるよう心がけました。
- タスク分割はできるだけ細かい粒度にし、メンバーがタスクを長期保有し開発が進まなくなったり、コンフリクトの多発を防ぐよう心がけました。

## 成果・価値

- 予定通り3ヶ月以内でチームでの主要機能の設計・コーディングを完了しました。
- ライブ配信機能を初期段階で実現できたことにより、クライアントも安心してプロダクト進行することができ、チーム全体としても士気が高まりました。
- チャット機能においてFirebaseのリアルタイム通信を利用したことで、ライブ配信のストリーミングと同期したチャット表示を実現できました。

---

# 2022/03 - 2022/06 : 可茂IT塾合同会社、NFTアートのマーケットプレイスアプリの新規開発

## プロジェクト概要

NFTアートのマーケットプレイスアプリをFlutterで開発し、自身を含めた学生インターンメインで実装を推進しました。

## 役割・体制

### 自身のポジションと役割（学生インターン）

- バックエンドおよびアプリ開発者として、Flutterを使用したiOS・Android向けのアプリ画面作成とFirebaseを用いたNoSQLデータベースの設計を担当しました。

### チーム規模と構成

- チームは学生インターン生を中心に約5名とPM1名で構成されており、各メンバーは未経験の技術に挑戦しながら協力して開発を進めました。

## 担当工程

- ☐ 企画
- ☐ 要件定義
- ☒ 設計
- ☒ コーディング

- ☐ テスト
- ☐ 運用/保守
- ☐ データ分析
- ☐ その他

## 背景・課題

- 受託案件としてNFTアートの売買を目的としたマーケットプレイスアプリの開発に取り組み、短期間で高品質なプロトタイプを作成し、クライアントの要望に応える必要があった。
- チームは学生インターン生主体であり、技術的な経験が乏しいメンバーも多く、新しい技術やツールの習得と実践を同時に進める必要がありました。
- さらに、未経験者主体のチームであったため、技術面だけでなく、『コミュニケーションや役割分担の調整も重要な課題』となっていました。

## 実際の取り組み

### 開発

- Flutterを用いて、NFTアートを一覧で表示する画面のUI作成・Firestoreデータ連携を行いました。
- Node.jsを用いて、NFTの売買に必要なデータ同期やバッチ処理のシステムを構築し、アプリのパフォーマンス向上を意識しました。

### 工夫

- チームの士気を高めるため、私は積極的に開発に参加し、コードの改善や新機能の提案を行うことで、チーム全体のモチベーション向上に役立ちました。
- 開発期間が短い中で、私は自発的にサーバーサイドのNode.jsコードやFirebase設定も担当し、多方面にわたる技術的な貢献を実現しました。

## 成果・価値

- 3ヶ月以内の開発期間で、チーム一丸となりNFTアートのマーケットプレイスアプリを完成させました。

---

## スキル

### プログラミング言語

言語	実務	個人開発
Dart	3年半	4年
TypeScript	1年	3年

## フレームワーク・ライブラリ

フレームワーク	実務	個人開発
Flutter	3年半	4年
Next.js	1年未満	1年
Vue.js	1年未満	-
Node.js	1年未満	3年

## データベース

データベース	実務	個人開発
NoSQL/Firestore	1年	4年

## インフラ・ツール

ツール・サービス	実務	個人開発
Firebase	3年半	4年
Git/GitHub	3年半	4年
AWS	-	1年未満
Figma	3年	

# キャリアビジョン

## 1. 個人開発と本業の両立

### 現在

- 個人開発で既に経験していた技術や知見を、実務でも活用できると判断して提案しています。実際に、Flutterでのユニットテストの構築やFirebase App Distributionを使ったテストアプリ配布の仕組みを個人開発で経験していたため、実務プロジェクトに導入を提案・実現しました。
- 個人開発で試した新しい技術や設計パターンを、本業プロジェクトに応用することで、技術選定や開発効率の向上に貢献してきました。
- 逆に、実務経験が個人開発の視野を広げることも実感しています。実務で得た知見（スクラム開発、アーキテクチャ設計など）を個人開発にも活かすことで、より実践的で実務に近い開発ができるようになりました。

- この双方向の好循環を通じて、技術力と視野を相互に高めています。自分が主体的に学び試したことが、チームの成果に還元されることにやりがいを感じています。

#### 今後

- 今後も、個人開発で培った技術や知見を実務に活かし、実務経験が個人開発の視野を広げるという双方向の好循環を継続していきたいと考えています。

## 2. チーム開発・スクラム改善

#### 現在

- 2週間スプリントのスクラム開発において、リファインメント、プランニング、デイリースクラム、レビュー、レトロスペクティブの各イベントに参加し、改善提案を実践してきました。
- 実際に、プランニング後の追加ミーティングやデイリースクラム前の「アプリデイリー」を導入し、タスク詳細化や影響範囲の明確化を行う運用を構築しました。これにより、スプリント完了率が安定し、手戻りや遅延を大幅に削減できました。
- レトロスペクティブでの改善提案を積極的に実践し、技術的負債の削減（リファクタリング、テスト基盤構築）と機能開発のバランスを取りながら、チームの自己組織化に貢献してきました。

#### 今後

- スクラムの各イベントを活用し、チームの開発効率やアウトプットの品質向上に継続的に取り組んでいきたいと考えています。
- 技術だけでなく、チーム運営や開発プロセスの改善でも価値を提供していきたいです。

## 3. 長期的なキャリアビジョンと想い

#### 技術的な成長

- 将来的には、API設計・認証基盤・DB設計・パフォーマンスチューニングなど、バックエンド・インフラ領域の知見を深め、アプリからインフラまで一貫して改善提案ができるエンジニアを目指しています。
- 技術的リーダーシップ（技術選定、アーキテクチャ設計）、チームマネジメント、プロダクトマネジメントなど、様々な方向性を経験しながら、自分に最も適したキャリアパスを見つけていきたいと考えています。

#### プロダクト・チームへの想い

- 自分が成長させたい、社会に広げたいと思えるサービスに、本気で貢献したいと考えています。技術だけでなく、プロダクトの価値やユーザーへの影響を実感できる環境で、情熱を持って取り組みたいです。
  - 感情移入でき、本気で挑戦できるチーム、プロダクト、企業とともに、自身も成長していきたいです。お互いを高め合い、共に目標に向かって進めるチームで、プロダクトの成功と自分の成長を両立させていきたいと考えています。
-

# 今後身に付けたい技術

## バックエンド (Go / API)

- 個人開発で Go によるAPIサーバー実装 (Gin、レイヤード設計、PostgreSQL/Supabase、Cognito、Lambda + API Gateway など) に触れてきたため、今後は「**業務レベルで再現性のある設計・運用**」まで深めたいと考えています。
- 深めたいテーマ：
  - **API設計**：リソース設計、エラーハンドリングの統一、バージョンニング、OpenAPI/Swaggerでの仕様管理
  - **認証・認可**：Cognito等を前提に、RBAC/権限境界、トークン運用
  - **非同期処理/バッチ**：集計・通知などをジョブとして分離し、失敗時のリトライや幂等性を担保
  - **DB設計**：正規化とクエリ性能のバランス、インデックス設計、マイグレーション運用
  - **パフォーマンス/信頼性**：ボトルネックの可視化、キャッシュ戦略、レート制御などの設計
  - **テスト**：ユニット/統合テストを前提に、変更に強いAPIの設計と回帰の仕組みづくり

## インフラ (AWS / Terraform)

- Firebase は個人・実務ともに利用してきましたが、個人開発では Terraform によるAWS構築 (S3、Cognito、Lambda、API Gateway、DynamoDB、SNS など) にも触れてきました。今後は **IaC前提で運用できるインフラ設計** を深めたいです。
- 深めたいテーマ：
  - **環境分離/運用**：dev/staging/prodの分離、設定管理、リリース手順の標準化
  - **権限設計/セキュリティ**：最小権限、シークレット管理、監査性の担保
  - **観測性**：ログ/メトリクス/アラート設計と、障害対応を回せる仕組み
  - **コスト最適化**：見える化と継続的な改善 (無駄なリソースの抑制)

## データ/分析 (ログ収集・BigQuery)

- 個人開発でログ収集・分析 (Firebase Analytics、BigQuery等) を実装してきたため、今後は「**計測→分析→改善**」を**継続的に回す設計**を強化したいです。
- 深めたいテーマ：
  - **イベント設計**：KPIに紐づくログ設計、命名規則、スキーマ管理
  - **集計・可視化**：定常レポート化、ダッシュボード、異常検知の仕組み
  - **改善サイクル**：施策→計測→検証の高速化 (プロダクト改善に直結させる)

## フロントエンド

### Web (Next.js / Vue.js)

- 管理画面/オーナー向けコンソールなど **複数フロントを運用する前提** で、中〜大規模開発に必要な設計・運用のノウハウを積みたいです。
- 深めたいテーマ：
  - **構成・状態管理**：ディレクトリ構成、状態管理、APIクライアント設計の標準化

- **BFF/API Routes**：フロントに最適化したAPI（認証・権限・キャッシュ）設計
- **レンダリング戦略**：SSR/SSG/ISRの使い分けとパフォーマンス最適化
- **運用**：エラー監視、ログ、権限周りの保守性

## Flutter

- Flutterでの開発経験を活かしつつ、プロダクト品質を高めるための習慣や仕組みづくりを強化していきたいです。
  - **E2Eテスト**：手動テスト削減・リリースサイクル高速化
  - **Widget Book**：UIコンポーネントのカタログ化・品質の平準化
  - **Crashlytics活用**：レポート分析～改善サイクルの仕組み化

## ネイティブ開発・プラグイン開発（Swift / Kotlin）

- 実務でAndroid TVアプリ開発を経験したことをきっかけに、Flutterの可能性を広げるため、ネイティブ開発にも挑戦したいと考えています。個人開発ではまだ経験が少ないため、今後実務で深めていきたいです。
- 深めたいテーマ：
  - **Flutterプラグイン開発**：Swift/Kotlinによるネイティブ機能の実装と、Flutter側への橋渡し
  - **デバイス拡張**：Apple Watch / Vision Pro / CarPlay など、モバイル以外のデバイスでの開発

---

## 成果物・アウトプット

- [GitHub](#)
- [Zenn](#)

## GitHub

主にUI/UXやライブラリの技術検証

## 技術ブログ

- コミュニティ 学生時代よりエンジニアコミュニティに所属し、定期的に技術ブログを執筆しています。  
[コミュニティプロフィール](#)
- Zenn 主に個人開発で試したことなどを執筆
- X 主に技術記事を引用した感想や意見を投稿

## 個人開発アプリ：Vintage Tracker

- **概要**  
日本国内の古着屋をマップで探せるアプリの開発・運用

- URL (iOS版) : <https://apps.apple.com/jp/app/vintage-tracker-%E5%8F%A4%E7%9D%80%E5%B1%8B%E3%83%9E%E3%83%83%E3%83%97/id6447299193>
- URL (Android版) : [https://play.google.com/store/apps/details?id=com.noriaki.vintage\\_tracker&pcampaignid=web\\_share](https://play.google.com/store/apps/details?id=com.noriaki.vintage_tracker&pcampaignid=web_share)
- URL (Web) : <https://vintage-tracker-web.vercel.app/>

※アプリのソースコードは公開しておりませんが、実際の動作や機能は上記リンクから確認可能です。

- 技術的ポイント / 実装例
  - モバイルアプリ (Flutter)
    - サブスクリプション機能 (RevenueCat)
    - 地図表示 (Google Maps / Mapbox 両方対応経験)
    - GPT API × Supabaseベクトル検索による自然言語検索チャットボット
    - プッシュ通知 (Firebase Messaging)
    - ログ収集 / 分析 (Firebase Analytics、API Gateway + Lambda + BigQuery)
  - Web版 (Next.js)
    - App RouterやUIフレームワークの活用
    - API Routesによるフルスタック開発 (フロントエンドとAPIを同一プロジェクトで開発)

---

## 働き方について

### 1. 成長機会と評価制度

- 昇給制度が整備されており、自身の向上意欲や成長が評価される環境を希望しています。
- 技術力の向上や貢献が適切に評価され、キャリアアップの道筋が明確であることを重視しています。

### 2. ワークライフバランス

- 子育てに寛容である環境を希望しています。
- プライベートと仕事の両立が可能で、柔軟な働き方ができる環境を重視しています。具体的には、残業時間が適切に管理されていること、リモートワークやフレックスタイムなどの柔軟な働き方が可能であること、有給休暇が取得しやすい環境であることを希望しています。

### 3. 技術的挑戦の機会

- 新しい技術やアプローチに挑戦できる環境を希望しています。

### 4. チーム開発とコミュニケーション

- オープンなコミュニケーションが取れる環境を希望しています。
  - チーム全体で成長し合える文化を重視しています。
-