職務経歴書

2021/8/29

佐佐木基顕

目次

1. 基本情報

- 氏名: 佐佐木基顕(Sasaki Motoaki)
- 誕生日: 1992年11月8日

ここ１年ほどは、サーバーサイド、インフラ、CI/CDなどをメインで担当。過去にはフロントエンドの実装経験もあり。

2. 職務経歴概要

2.1. 株式会社レスタス

- 設立: 2011年6月
- 資本金: 5億1,062万円（資本準備金を含む）
- 社員数: 43人
- 在職期間: 2020/4〜現在

自社ECサービスの開発に従事。これまでに、4つの新規開発、１つの追加開発のプロジェクトに参画し、内2つのプロジェクトでは開発リーダーを兼任。メインはバックエンドの開発だが、フロントエンド、インフラも必要に応じて担当。現在は、新規サービスの開発リーダー兼開発メンバーとして、要件定義〜テストを担当している。

2.2. 株式会社NSC

- 設立: 1971年3月11日
- 資本金: 4,550万円
- 社員数: 523人（2021年2月現在）
- 在職期間: 2018/4〜2020/1

化学メーカーの製造職として、社内の新規技術を用いた製品の製造に従事。独自の分析と工夫により、作業の正確性及び効率が工程内でトップであったほか、サブリーダーとしてマネジメントにも関わった。さらに、空いた時間を利用してデータ収集及び分析による提案や、現場の改善なども積極的に行った。

なお、本企業での経験の詳細については、業界業種ともに、異なるものであるため、以下の職務経歴詳細には記載していない。

3. 職務経歴詳細

3.1. 既存APIのリプレイス

3.1.1. 概要

既存のRubyで書かれたAPIをGoでリプレイスするプロジェクト。

3.1.2. 期間

2022/10/16現在継続中。トータル3ヶ月ほどで完了予定。

3.1.3. メンバー

プロジェクト全体:3人

役割	人数
サーバーサイドエンジニア兼インフラエンジニア	1(自分)
サーバーサイドエンジニア	2

3.1.4. 担当業務一覧

- 技術選定
- GitHub Actionsを用いたCIの設定
- 開発環境の整備
- 既存のORMのモデルからGoのORM(gorm)のモデルを生成するプログラムの作成
- 既存のOpenAPIドキュメントからGoのコードを生成するプログラムの作成
- 既存APIのGoへの書き換え
- Kubernetes(GKE Autopilot)を用いた、staging環境、production環境の構築
- コードレビュー
- reviewdogを用いた、レビュー半自動化の仕組み導入

3.1.5. 課題、対処方法、振り返りなど

リードエンジニアとしての参画。
他のエンジニアがGoの実務未経験であったため、コードの品質を担保するために、開発環境のコード化(dockerコンテナ化, VSCode設定の共有)や、CIの充実、実装の手本となるようAPIのいくつかを事前に移植、モデル変換プログラムの作成、などを行っており、現状プロジェクト進行上問題となるようなことは起こっていない。

技術選定では、可能な限りデファクトスタンダードなものや、Ruby on Rails経験者である他の開発者にとって学習コストの低いものを選んでいる。

3.1.6. 使用技術

- Ruby
- Go
- ShellScript
- gin
- Ruby on Rails
- gorm
- gin-swagger
- zap
- gin-zap
- air
- reviewdog
- staticcheck
- golangci-lint
- docker
- docker compose
- Kubernetes
- Cloud Build
- Google Container Registry
- Google Kubernetes Engine(Autopilot)
- GitHub
- GitHub Actions

3.2. 汎用ECシステム開発

3.2.1. 概要

汎用的なECシステムの構築プロジェクト。

3.2.2. 期間

参画~リリースまで: 7ヶ月
以後、改善・保守などを現在まで継続

3.2.3. メンバー

プロジェクト全体:5人

役割	人数
プロジェクトマネージャー兼サーバーサイドエンジニア兼インフラエンジニア	1
サーバーサイドエンジニア兼インフラエンジニア	1(自分)
フロントエンドエンジニア	1
デザイナー兼フロントエンドエンジニア	1
デザイナー	1

3.2.4. 担当業務

- 担当機能における、要件定義、テーブル設計、実装、テスト(単体、API)作成
- OpenAPIドキュメント生成機能付きFWの導入による、OpenAPIドキュメント自動生成の仕組みの構築
- APIの実装
- GitHub Actionsを用いたCIの設定
- Kubernetes(GKE Autopilot)を用いた、staging環境、production環境の構築
- Kubernetes(GKE Autopilot)を用いた、検証環境自動構築の仕組みの構築
- デプロイ属人化を防ぐslackアプリ(TypeScript)の作成
- 画像を透過処理して保存するAPIサーバー(Go)の作成

3.2.5. 使用技術

- Ruby
- Go
- ShellScript
- JavaScript
- TypeScript
- Ruby on Rails
- grape
- goa
- rubocop
- eslint
- prettier
- jest
- zap
- make
- esbuild
- staticcheck
- docker
- docker compose
- Kubernetes
- Cloud Build
- Google Container Registry
- Google Cloud Storage
- Google Kubernetes Engine(Autopilot)
- GitHub
- GitHub Actions

3.2.6. 課題、対処方法、振り返りなど

サーバーサイドエンジニアとして参画。のちに希望してインフラも兼任。
これまでなかった、OpenAPIドキュメント自動生成の仕組みを構築することで、APIドキュメントの管理コストの削減、APIドキュメントが常に最新のコードと同じ状態が担保される、フロントエンドからのAPIの明確化、などに加え、生成されたOpenAPIドキュメントをもとに、プロキシサーバーの設定ファイルを自動生成する仕組みを構築することができた、などの間接的なメリットも感じることができ、成功だったといえる。

3.3. B2B向け新規自社グループECシステム構築

3.3.1. 概要

グループ会社の、B2B向け新規ECシステムの構築プロジェクト。

3.3.2. 期間

7ヶ月

3.3.3. メンバー

プロジェクト全体:10人

役割	人数
プロジェクトマネージャー	1
サーバーサイドエンジニア	2(自分含む)
フロントエンドエンジニア	1
デザイナー兼フロントエンドエンジニア	1
デザイナー	2

3.3.4. 担当業務

- 企画サイドと、エンジニア間での各種調整業務
- APIの実装
- 担当機能における、要件定義、テーブル設計、実装、単体テスト
- GitHub Actionsを用いたCIの設定

3.3.5. 使用技術

- Ruby
- JavaScript
- Node.js
- Ruby on Rails
- rubocop
- docker
- docker compose
- Google Cloud Functions
- GitHub
- GitHub Actions

3.3.6. 課題、対処方法、振り返りなど

サーバーサイドエンジニアとして参画。のちに、開発をまとめていたサーバーサイドエンジニアの退職に伴い、調整業務も兼任。
引き続き、Ruby on Railsを用いた開発をメインで担当。
今までに扱ったことのない技術(ActiveJob, Github Actionsなど)に積極的に挑戦したり、以前よりさらに保守性の高いコードを書くことを意識して開発をおこなった。

3.4. B2B向け新規自社ECサイト構築

3.4.1. 概要

リッチなフロントエンドアプリをベースとした、ECシステムの構築プロジェクト。

3.4.2. 期間

7ヶ月

3.4.3. メンバー

プロジェクト全体:7人

役割	人数
プロジェクトマネージャー	1
サーバーサイドエンジニア	1
フロントエンドエンジニア兼サーバーサイドエンジニア	1(自分)
フロントエンドエンジニア	2
デザイナー兼フロントエンドエンジニア	1
インフラエンジニア	1

3.4.4. 担当業務

- PMと各エンジニアとの各種調整(プロジェクト中盤以降)
- フロントエンドでの画像アップロード用コンポーネント作成
- 画像アップロードAPIサーバー(Go)の作成
- フロントエンドアプリのDocker化

3.4.5. 使用技術

- Go
- JavaScript
- gin
- Vue.js
- Nuxt.js
- docker
- docker compose
- Cloud Run
- GitHub
- GitHub Actions

3.4.6. 課題、対処方法、振り返りなど

フロントエンドエンジニア兼サーバーサイドエンジニアとして参画。のちに、PM~エンジニア間の調整業務を兼任。
Nuxt.js, Goを用いた開発は初めての挑戦だったが、事前に学習を行なっていたので、問題なくキャッチアップすることができた。
また、フロントエンド、サーバーサイドに横断するような機能の実装を初めて担当した。

3.5. 既存自社ECサイトの機能追加及び改善

3.5.1. 概要

既存自社ECサイトに対する、機能追加及び改善のプロジェクト

3.5.2. 期間

3ヶ月

3.5.3. メンバー

プロジェクト全体:4人

役割	人数
プロジェクトマネージャー	1
サーバーサイドエンジニア(実装)	1
サーバーサイドエンジニア(要件定義~設計)	1(自分)
インフラエンジニア	1

3.5.4. 担当業務

- 要件定義、テーブル設計、実装、テスト
- APIの実装

3.5.5. 使用技術

- Ruby
- JavaScript
- TypeScript
- Ruby on Rails
- Vue.js
- docker
- docker compose
- GitHub

3.5.6. 課題、対処方法、振り返りなど

実装担当のサーバーサイドエンジニアのサポート(主に設計面)として参画。のちに、実装担当のサーバーサイドエンジニア退職に伴い、実装も担当。
改修対象のアプリで、TypeScript、VueCompositionAPIなど、未経験の技術が使われていたが、難なくキャッチアップし、改修することができた。
また、エンジニアになって初めてマネージメントの立場を経験することができた。

3.6. B2B向け新規自社ECサイト構築

3.6.1. 概要

商材に依存しないECシステムの構築プロジェクト

3.6.2. 期間

3ヶ月

3.6.3. メンバー

プロジェクト全体:4人

役割	人数
プロジェクトマネージャー	1
サーバーサイドエンジニア	1(自分)
デザイナー兼フロントエンドエンジニア	1
インフラエンジニア	1

3.6.4. 担当業務

- 要件定義、テーブル設計、実装、テスト
- 外部APIと連携した定期実行処理の設計・実装
- インフラエンジニアとの調整

3.6.5. 使用技術

- Ruby
- JavaScript
- Ruby on Rails
- docker
- docker compose
- GitHub

3.6.6. 課題、対処方法、振り返りなど

サーバーサイドエンジニアとして参画。
要件定義~設計や、テーブル設計などを本格的に行なったのは初めてであった。
Kubernetesの機能を用いた定期実行処理や、外部APIと連携した処理を初めて実装するなど、技術的にも挑戦することができた。

3.7. B2B向け新規商材向け自社ECサイト構築

3.7.1. 概要

B2B向け新規商材を対象とした自社ECシステム構築

3.7.2. 期間

4ヶ月

3.7.3. メンバー

プロジェクト全体:6人

役割	人数
プロジェクトマネージャー	1
サーバーサイドエンジニア	2(自分含む)
フロントエンドエンジニア	1
デザイナー兼フロントエンドエンジニア	1
インフラエンジニア	1

※プロジェクトの途中からサーバーサイドエンジニアとして参画。

3.7.4. 担当業務

- バックエンド機能の開発・テストを担当。
- 一部機能は、仕様書を元に、自ら設計〜開発まで担当。

3.7.5. 使用技術

- Ruby

- JavaScript
- Ruby on Rails
- Vue.js
- docker
- docker compose
- GitHub

3.7.6. 課題、対処方法、振り返りなど

サーバーサイドエンジニアとしてプロジェクトの途中から参画。
初めての実務レベルの開発を通じて、Ruby on Railsを用いた基本的な開発力を培うことができた。
Git, GitHubを用いたプルリクエストベースのチーム開発や、Dockerを用いた開発に慣れることができた。

4. スキルシート

4.1. 技術

カテゴリー	スキル	経験期間	備考
OS	macOS	1年3ヶ月	難なく使用可能。業務で常に使用しており、 プライベートでは７年前から使用。
OS	Linux	1年3ヶ月	難なく使用可能。Dockerを用いた開発や、production、 staging環境のサーバーなどで経験している。
言語	Ruby	1年3ヶ月	難なくプログラミングが可能。 株式会社レスタスでのバックエンド開発で主に使用。
言語	Go	2ヶ月	基本的なプログラミングが可能。株式会社レスタスでBFF構築の際に使用。
言語	JavaScript	1年	難なくプログラミングが可能。 株式会社レスタスでのフロントエンド開発で主に使用。
フレームワーク	Ruby on Rails	1年3ヶ月	難なくプログラミングが可能。メタプログラミング、クラスの継承、 APIを用いた開発なども可能。
フレームワーク	Gin	2ヶ月	基本的なプログラミングが可能。APIの作成、 Errorインターフェースを用いたエラーハンドリングや、 エラー監視ツールSentryの導入などの経験がある。
フレームワーク	Vue.js	1年	OptionsAPIを用いたプログラミングが難なく可能。 単一ファイルコンポーネント、VueRouter、 Vuexなどを用いたSPAを作成可能。 CompositionAPIは基本的なコードの読み書きができる。
フレームワーク	Nuxt.js	3ヶ月	難なく開発可能なほか、設定ファイルの編集も可能。
データベース	SQL(PostgreSQL)	1年3ヶ月	ORMを用いた、クエリの組み立て、調整などが可能。素のSQLを用いた、 基本的なCRUD操作が可能。
インフラ	Kubernetes	1ヶ月	k9sといったコマンドラインツールを用いたアプリケーションのデバッグや、 マニフェストファイルを記述し、 Kubernetesリソースを追加した経験がある。

カテゴリー	スキル	経験期間	備考
DevOps	GitHubActions	1ヶ月	特定のブランチへのpushを起点として、自動でテストを行うような、 CIの構築経験あり。
クラウド	GCP	1年3ヶ月	いくつかのマネージドサービス(特に、CloudFunctions、CloudRun、 CloudStorageなど)を難なく使用可能。

4.2. その他

カテゴリー	スキル	経験期間	備考
マネジメント	教育	3ヶ月	質問しやすい空気作りや、自発的な思考を促せるよう意識しながら、 後輩エンジニア教育をした経験がある。
マネジメント	開発リーダー	5ヶ月	開発以外の担当者との調整や、機能要件のドキュメント化、 メンバーの進捗管理などを経験。

5. アカウント

サービス	アカウント
GitHub	https://github.com/msasaki666
Qiita	https://qiita.com/motoakii