

投稿日 2020/09/30 更新日 2025/05/30

SPA + REST API構成のサービス開発リファレンス

はじめに

SPA（シングルページアプリケーション）とREST APIで構成されるウェブアプリケーションを開発する際にリファレンスとして活用して頂ける方式設計のガイドと、アプリケーションのコード例を提供します。

このコンテンツのねらい

SI型のプロジェクトとサービス型のプロジェクトを比較すると、サービス型のプロジェクトはより良いUXを提供するためシームレスな操作感やインタラクティブなUIが必要とされる傾向にあります。ウェブアプリケーションでそういったUIを構築するためにはSPAが有効です。それからUIをSPAで構築する場合、サーバーサイドはREST APIを採用することが一般的です。

また、サービス型のプロジェクトは少人数かつ短期間で、動くものを重視して開発する傾向にあります。そのために技術選定や方式設計を十分に行う時間が確保できなかったり、開発のためのドキュメント作成を劣後させたりすることがあります。

こういった初期投資を削る行為は開発が進めば進むほど影響が大きくなります。また、サービス型のプロジェクトは「納品」のようなわかりやすいゴールはありません。リリースがスタートラインであり、そこからサービスが継続される限りアップデートを重ねて行きます。そういったときに開発のためのドキュメントが不足していると、徐々に開発が困難になって行くのは容易に想像できます。

これらの課題に対応するため、本コンテンツはSPAとREST APIで構成されるウェブアプリケーションの方式を設計する際にリファレンスとして活用して頂ける方式設計のガイドを提供します。また、理解を助けるためにコード例を提供します。コード例は断片的なものでなく、1つのアプリケーションとして動作するようなものとなっています。断片的なコード例だと実際のアプリケーションを開発する際、どこにどう適用すれば良いのか分かりづらいためです。

このコンテンツの対象者

このコンテンツはSPAとREST API、あるいはどちらか片方を使ったウェブアプリケーション開発を始めようとしているエンジニアを対象としています。

また、前のセクションではサービス型プロジェクトについて言及していますが、SI型のプロジェクトでの利用を制限するものではありません。SI型のプロジェクトであっても、有用だと感じて頂けるなら是非ともご活用ください。

コンテンツ紹介

- [方式設計ガイド](#)

SPAとREST APIで構成されるウェブアプリケーションを開発する際に有用だと思われるものをまとめました。REST APIの設計、SPAとREST APIの通信、SPAでのファイルダウンロードなどの方式に言及しています。

- [コード例](#)

方式設計ガイドに書かれていることを実装したコード例です。小さなチャットを題材にしたウェブアプリケーションとなっています。プログラミングを行う際、ご参考になさってください。また、必要に応じてソースコードを流用して再利用して頂いても構いません。

- [ハンズオンコンテンツ](#)

SPAやREST APIの開発が不慣れな方のためにハンズオンコンテンツを用意しました。テキストを読みながらコーディングをすることで学べるようになっていきますので、プロジェクトへの新規メンバー参入時などに是非ともご活用ください。※ コード例とは違い[Next.js](#)を使用してアプリケーションを作成しています（コード例は create-react-app + react-router）。

なお、SPA、REST APIに直接関連しない部分についての方式設計は本コンテンツのスコープ外です。それらの部分については[Nablarchのアプリケーション方式設計書サンプル](#)を参考になさってください。

本コンテンツではSPAを実現するためのライブラリとして[React](#)を選択しています。Reactを選択するにあたってはThoughtworksのTECHNOLOGY RADAR、Google Trends、GitHubでのStar数などを参考にしました(いずれも2020年2月時点)。言語は[TypeScript](#)です。

それからREST APIを実現するためのフレームワークとして[Nablarch](#)を選択しています。NablarchはTISがOSSとして公開している社内標準フレームワークです。言語はJavaです。

Reactを使って作られたSPAとNablarchを使って作られたREST APIは相互に依存はしていません。ですので、例えば次のような使い方も考えられます。

- SPAはチームが得意なVueを使いたい。REST APIはお任せ
 - →SPAの部分はVueに読み替えて（カスタマイズをして）使って頂いて、REST APIの部分はそのまま使って頂く
- REST APIのみを提供するプロジェクト
 - →SPAの部分は使用せず、REST APIの部分だけを使って頂く

JavaScriptのライブラリは移り変わりが激しいのでなるべく依存するライブラリは少なくするようにしています。

コンテンツの使い方

方式設計ガイドに記載されている内容はプロジェクトへそのまま適用して頂いても構いません。ただし、少なくとも一度は全量に目を通して内容を理解し、プロジェクトに合わない箇所や足りない箇所がないか確認をしてください。

コード例は開発中に参照して頂いたり、利用可能なコードを流用したりして活用してください。

詳しくは各Gitリポジトリ内に置かれたREADME.mdを参照してください。

今後の展開

本コンテンツへ頂いたフィードバックや本コンテンツに関連する技術のトレンドを見ながら継続的にアップデートを行ないます。

※ 本コンテンツに含まれるドキュメントは[Fintan コンテンツ 使用許諾条項](#)または[クリエイティブコモンズ\(Creative Commons\) 4.0 の「表示—継承」](#)に準拠しており、コード例は[Apache License 2.0](#)に準拠しています。

※ 本コンテンツ内に記載されている会社名、製品名は、各社の登録商標または商標です。

/* Recommend */

「ITアーキテクチャ」 のおすすめ記事はこちら

この記事に関連する記事もお読みください。

ITアーキテクチャ

サービス開発のエンジニアリング
ガイド

2018/10/01  6

最近投稿された記事も用意しました。

ITアーキテクチャ

NEW

生成AIの新潮流：AIエージェント
勉強会を開催しました

2025/05/02  11

ITアーキテクチャ

コーディング規約

2022/11/02  22

ITアーキテクチャ

Javaアプリケーションフレームワ
ークと設計・開発をサポートする
コンテンツの活用

2022/11/02  14

「ITアーキテクチャ」 で最も読まれている記事を以下にまとめています。

ITアーキテクチャ

Update

SPA + REST API構成のサービス
開発リファレンス

ITアーキテクチャ

コーディング規約

ITアーキテクチャ

NEW

生成AIの新潮流：AIエージェント
勉強会を開催しました

2020/09/30  15

2022/11/02  22

2025/05/02  11