報道各位

2019 年 5 月 17 日
株式会社インプレスR&D
https://nextpublishing.jp/

動画投稿サイトの制作を題材に学ぶ Firebase！

『Firebase によるサーバーレスシングルページアプリケーション』発行

技術の泉シリーズ、5 月の新刊

インプレスグループで電子出版事業を手がける株式会社インプレス R&D は、『Firebase によるサーバーレスシングルページアプリケーション』(著者:小島 佑一)を発行いたします。

最新の知見を発信する『技術の泉シリーズ』は、「技術書典」をはじめとした各種即売会や、勉強会・LT 会などで頒布された技術同人誌を底本とした商業書籍を刊行し、技術同人誌の普及と発展に貢献することを目指します。

『Firebaseによるサーバーレスシングルページアプリケーション』
https://nextpublishing.jp/isbn/9784844398998

著者:小島 佑一
小売希望価格:電子書籍版 1600 円(税別)／印刷書籍版 1800 円(税別)
電子書籍版フォーマット:EPUB3／Kindle Format8
印刷書籍版仕様:B5 判／カラー／本文 112 ページ
ISBN:978-4-8443-9899-8
発行:インプレス R&D

＜＜発行主旨・内容紹介＞＞

本書は、動画投稿サイトの制作を題材とした React によるフロントエンド開発と Firebase の主要な機能である Authentication、 Firestore、Cloud Storage、Cloud Functions の各機能を理解しつつ、Firebase によるシングルページアプリケーションの開発について学ぶことができる入門書です。

〈本書の対象読者〉

・Firebase は名前は聞いたことがあるが、実際に触ったことはない人
・JavaScript だけで、Web アプリケーションを開発してみたい人

・Firebase と 何かしらのモダンな JS フレームワークを組み合わせてアプリケーションを開発してみたい人
・サーバーサイドだけでなく、フロントエンドの開発にも興味がある人
（本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。）

Firebase のセットアップから丁寧に解説

```
yarn start

Happy hacking!
```

生成が完了したら、ローカルで正しくアプリケーションが動作することを確認してください。

```
npm start
```

Reactの Welcome 画面が表示されます。

図2.1: welcome_to_react

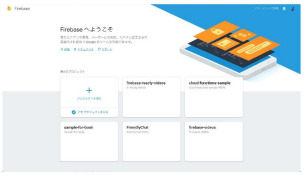
To get started, edit src/App.js and save to reload.

2.1.2 Firebaseのセットアップ

Firebaseプロジェクトの新規作成

https://console.firebase.google.com/から、「プロジェクトを追加」の部分をクリックし、新規プロジェクトを作成します。

図2.2: create_new_firebase_project



次に、プロジェクト名とリージョンを選択します。今回のプロジェクト名は `firebase-youtube-clone` にします。Firestore のリージョンは、東京リージョンを指す `asia-northeast1` を選択します。

図2.3: setting_firebase_project

NoSQL データベースの考え方を RDBMS と比較しつつ、Firebase のデータベース管理の仕組みを紹介

第5章 Firestoreによるデータベース管理

5.1 NoSQLデータベースとFirestore

Firestoreは、NoSQLデータベースと呼ばれるものです。MongoDBのようなドキュメント指向型のNoSQLデータベースと同じカテゴリーにあたります。NoSQLのデータベースであるため、MySQLやPostgreSQLのようなRDBMS（リレーショナルデータベース）とは、かなり趣の異なるデータベースです。

恐らく、RDBMSを使ったことはあれどNoSQLデータベースを使ったことがある人は少ないのではないでしょうか。ここでは、RDBMSと比較しつつNoSQLデータベースであるFirestoreとは何かについて解説をしていきます。

5.1.1 NoSQLデータベースについて

まず、RDBMSでデータ管理する場合は、テーブルの行はスキーマとテーブルを用いて、全て厳密に何かの型が定義されている必要があります。例えば、`Users`と`Videos`というデータを管理するときは、表5.1や表5.2のようなテーブルを作ることになるでしょう。

`Users`と`Videos`を連携したい場合、`Videos`は、一意のIDを含む外部キーを`Users`テーブルに紐付けるというのがよくある方法でしょう。

表5.1: Usersテーブル

primary_key	username	age	login_date	is_admin
123	"Mike"	22	20180101	1
345	"Tom"	30	20180202	0

表5.2: Videosテーブル

primary_key	title	content_type	download_url	foreign_key
777	"Firebase 入門"	"mp4"	https://hoge.com	123
888	"すべらない話"	"mov"	https://xxxx.xxxx	345

複数テーブルにまたがる条件で目的のデータを取得したい場合、基本的には外部にキーをもたせて、SQLで`JOIN`を用いて目的のデータを取得していくのがRDBMSでは王道でしょう。

しかし、Firestoreのような NoSQLデータベースでは事情が異なります。Firestoreでは、RDBMSのようにきっちりとデータ構造を厳密に定義してデータを格納しません。キーと値がセットになったJSONオブジェクトをそのまま格納することが一般的です。また、異なるキーバリューのデータをFirestoreに保存することも可能です。

例えば、`Video`をFirestoreに格納する場合は、次のようなイメージでJSONオブジェクトをツリー構造で格納します。

図5.1: Firestoreのデータ構造の例

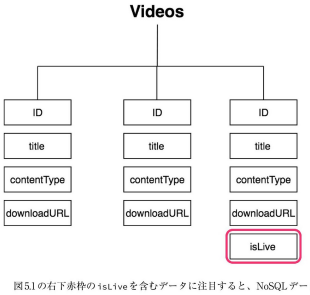


図5.1の右下赤枠の`isLive`を含むデータに注目すると、NoSQLデータベースはRDBMSと違い、一部のデータにだけ異なるフィールドを追加することができる、といえます。このような特徴があるため、FirestoreのようなNoSQLデータベースは**スキーマレス**なシステムと呼ばれ、事前に厳密なテーブル設計をしなくてもデータを格納できるメリットがあります。

では、`Videos`を`Users`に属するデータとして格納したい、という場合を考えてみましょう。RDBMSであれば、`Videos`テーブルに外部キーを持たせて正規化をしますが、NoSQLデータベースではSQLの`JOIN`という概念すら存在しません。

仮に、`Videos`に`user_id`という外部キーのようなフィールドを追加しても`JOIN`ができません。特定ユーザーの動画だけで、さらに他の条件に合致するようなクエリを発行するのが難しいのです。

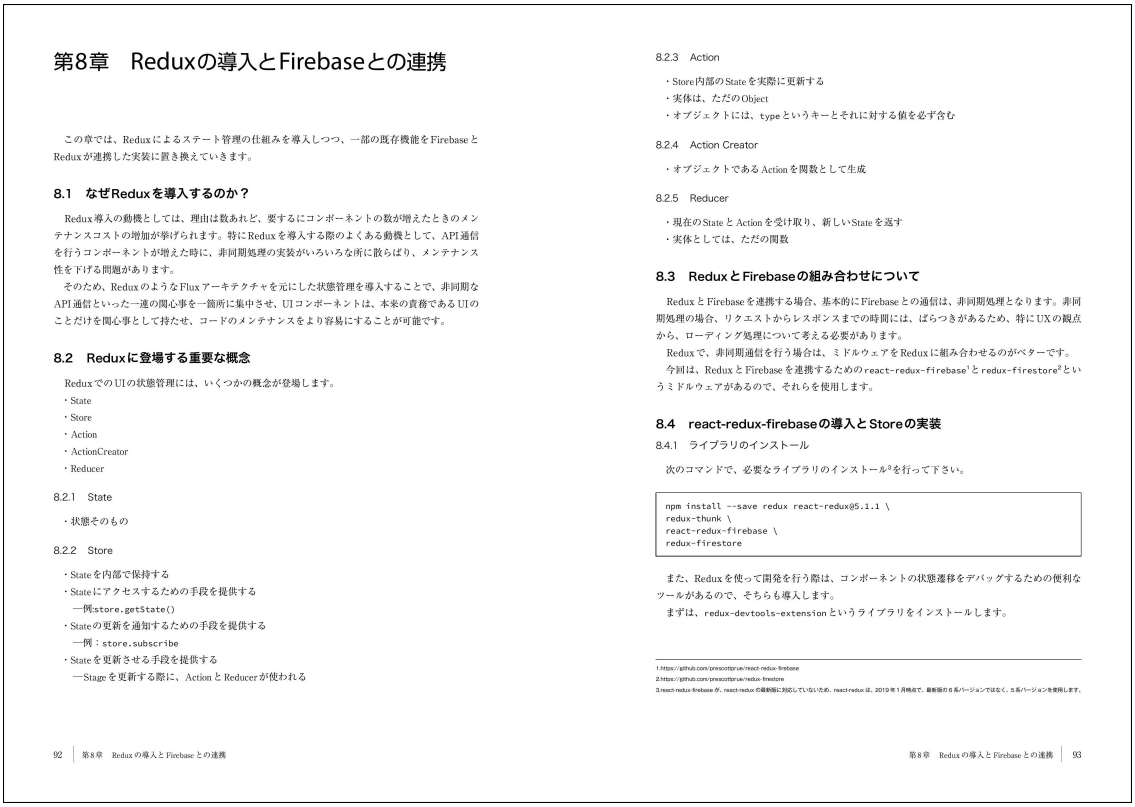
そのため、NoSQLデータベースであるFirestoreでは、データの重複を許容して、他のデータをコピーする方法があります。

RDBMSに慣れ親しんだ方にとっては、データの重複を許すというのはかなり気持ち悪く感じることでしょう。実際、この方法を採用する場合、重複元のデータが変更された場合は重複データもその変更を反映させないと、データの整合性を保てないというデメリットがあります。

5.1.2 なぜNoSQLデータベース?

RDBMSは、データが正規化され、重複データが存在しないクリーンな世界でした。NoSQLデータベースは、重複データを許容し、データの整合性を保つためのコストがRDBMSと比べると大きいという明らかな欠点があります。

Firebase と Redux の連携による利点を解説



第8章　Reduxの導入とFirebaseとの連携

この章では、Reduxによるステート管理の仕組みを導入しつつ、一部の既存機能をFirebaseとReduxが連携した実装に置き換えていきます。

8.1　なぜReduxを導入するのか？

Redux導入の動機としては、理由は数あれど、要するにコンポーネントの数が増えたときのメンテナンスコストの増加が挙げられます。特にReduxを導入する際のよくある動機として、API通信を行うコンポーネントが増えた時に、非同期処理の実装がいろいろな所に散らばり、メンテナンス性を下げる問題があります。

そのため、ReduxのようなFluxアーキテクチャを元にした状態管理を導入することで、非同期なAPI通信といった一連の関心事を一箇所に集中させ、UIコンポーネントは、本来の責務であるUIのことだけを関心事として持たせ、コードのメンテナンスをより容易にすることが可能です。

8.2　Reduxに登場する重要な概念

Reduxでの UI の状態管理には、いくつかの概念が登場します。

・State
・Store
・Action
・ActionCreator
・Reducer

8.2.1　State

・状態そのもの

8.2.2　Store

・Stateを内部で保持する
・Stateにアクセスするための手段を提供する
　一例:`store.getState()`
・Stateの更新を通知するための手段を提供する
　一例：`store.subscribe`
・Stateを更新させる手段を提供する
　—Stageを更新する際に、Actionと Reducerが使われる

8.2.3　Action

・Store内部のStateを実際に更新する
・実体は、ただのObject
・オブジェクトには、typeというキーとそれに対する値を必ず含む

8.2.4　Action Creator

・オブジェクトであるActionを関数として生成

8.2.5　Reducer

・現在のStateとActionを受け取り、新しいStateを返す
・実体としては、ただの関数

8.3　Reduxと Firebaseの組み合わせについて

Reduxと Firebaseを連携する場合、基本的に Firebaseとの通信は、非同期処理となります。非同期処理の場合、リクエストからレスポンスまでの時間には、ばらつきがあるため、特にUXの観点から、ローディング処理について考える必要があります。

Reduxで、非同期通信を行う場合は、ミドルウェアをReduxに組み合わせるのがベターです。

今回は、Reduxと Firebaseを連携するための`react-redux-firebase`[1]と`redux-firestore`[2]というミドルウェアがあるので、それらを使用します。

8.4　react-redux-firebaseの導入と Storeの実装

8.4.1　ライブラリのインストール

次のコマンドで、必要なライブラリのインストール[3]を行って下さい。

```
npm install --save redux react-redux@5.1.1 \
redux-thunk \
react-redux-firebase \
redux-firestore
```

また、Reduxを使って開発を行う際は、コンポーネントの状態遷移をデバッグするための便利なツールがあるので、そちらも導入します。

まずは、`redux-devtools-extension`というライブラリをインストールします。

1.https://github.com/prescottprue/react-redux-firebase
2.https://github.com/prescottprue/redux-firestore
3.react-redux-firebaseが、react-reduxの最新版に対応していないため、react-reduxは、2019年1月時点で、最新版の6系バージョンではなく、5系バージョンを使用します。

＜＜目次＞＞

＜＜著者紹介＞＞

小島 佑一(こじま ゆういち)

Rails に飽きてきたので、プライベートや業務を含め、React, Vue などのフロントエンドまわりをやるようになった意識低い系エンジニア。インフラまわりが弱いエンジニアでも、迅速にサービスを構築できる可能性を秘めた Firebase に夢中になり、現在は、Firebase を用いたサーバーレスな構成の新規プロダクトを開発中。

＜＜販売ストア＞＞

電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※　各ストアでの販売は準備が整いしだい開始されます。

※　全国の一般書店からもご注文いただけます。

【インプレス R&D】 https://nextpublishing.jp/

株式会社インプレス R&D（本社：東京都千代田区、代表取締役社長：井芹昌信）は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【インプレスグループ】 https://www.impressholdings.com/

株式会社インプレスホールディングス(本社:東京都千代田区、代表取締役:唐島夏生、証券コード:東証 1 部 9479)を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「旅・鉄道」「学術・理工学」を主要テーマに専門性の高いメデイア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター
TEL 03-6837-4820
電子メール: np-info@impress.co.jp