

2020年2月28日
株式会社インプレスR&D
<https://nextpublishing.jp/>

Firestore を実務で活用する！

『実践 Firestore』発行

技術の泉シリーズ、2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、『実践 Firestore』（著者：福田 雄貴）を発行いたしました。

最新の知見を発信する『技術の泉シリーズ』は、「技術書典」や「技術書同人誌博覧会」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。

『実践 Firestore』

<https://nextpublishing.jp/isbn/9784844378525>



著者：福田 雄貴

小売希望価格：電子書籍版 1600 円（税別）／印刷書籍版 2000 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 142 ページ

ISBN：978-4-8443-7852-5

発行：インプレス R&D

<<発行主旨・内容紹介>>

近年、Firebase は個人でのアプリケーション開発を中心に利用が広がっています。その中心的な機能のひとつである Firestore は新しい NoSQL マネージドデータベースであり、これまでのデータベースにはない特徴を数多く備えています。

本書には筆者が業務でFirestoreに向き合い続けて獲得した、Firestoreを有効かつ安全に利用するためのエッセンスが詰め込まれています。

(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。)

Firestoreの概要と特徴、データアクセスの基礎を解説

代表であるトランザクションを利用する場合には、関数がトリガーされる時にコンテキストを通じて渡されるイベントIDをチェックして、同じ処理が2回以上実行されないように制御します。

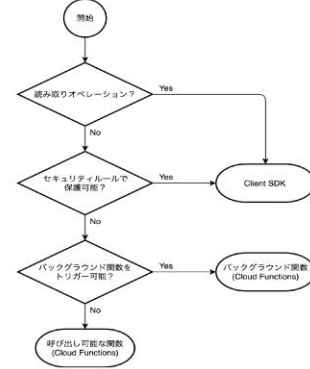
規模の大きなアプリケーションでは、たくさんのバックグラウンド関数が動作することになります。1回のFirestoreイベントが複数の関数をトリガーすることは全く問題ありません。一般的に、Firestoreへの書き込みが代表される副作用のある処理は呼び出し回数につき1回だけ呼び出すように設計すると、疎結合で複雑度の高いシステムを構築できます。

バックグラウンド関数は、開発者が指定した方法で呼び出されていることが保証されています。呼び出し可能な関数とは異なり、適切な権限をもったユーザーによって呼び出されたかどうかをチェックする必要があります。本質的なビジネスロジックのみを記述すればよく、余計なコードを記述しなくて良いという大きなメリットがあります。

1.4.2 関数の利用判断

図1.1にこれまでの議論をまとめたCloud Functions利用の指針を示します。

図 1.1: Cloud Functions利用のフローチャート



繰り返しになりますが、Firestoreの世界ではアプリケーションの要件を満たす最適なデータモデルを設計することが何よりも重要です。原則として、読み取りオペレーションでCloud Functionsを使ってはいけません。また、セキュリティルールによって機密性と完全性が担保できるかどうかはFirestoreの書き込み方法を決定する際の判断ポイントになります。つまり、「Cloud Functionsを使わずに読み取り可能であること」と「セキュリティルールでの扱いが容易であること」が良いデータモデルの必要条件となります。

呼び出し可能な関数によるFirestoreへの書き込み処理を使いたくないなら、一度データモデルの設計に問題がないかを見直してみてください。多くの要件はクライアントからの書き込みとバックグラウンド関数の利用により十分シンプルに実装できます。

複数のプラットフォームに対応したアプリケーションを構築する際に、ロジックの共通化を目的として呼び出し可能な関数を利用する例が散見されますが、このような理由でCloud Functionsを利用してはいけません。あくまで、理想的なデータモデルを採用した上でセキュリティルールが有

Firestoreのセキュリティルールを解説

4.5 アトミックオペレーションのセキュリティルール

アトミックオペレーションを利用するユースケースでは、複数のドキュメントを同時に書き込みオペレーションの対象とする場合があります。そのようなケースでは、それぞれの書き込みオペレーションに対するセキュリティルールで他のドキュメントが同時に更新されることをチェックする複合的なセキュリティルールによって整合性を担保します。

4.5.1 ルールを記述するための条件

これまでに紹介したような通常のcreateやupdateに対するルールを単独で用いただけでは、アトミックオペレーションによる更新で整合性を担保することを期待しているドキュメントが個別に更新されてしまいます。アトミックオペレーションに限定して書き込みを許可するルールを記述するためには、アトミックオペレーションに含まれるドキュメントのID (サブコレクションの場合は上位ドキュメントのIDも含む)、フィールドに持っているリファレンス、request.authから取得できる情報を組み合わせて、同時に書き込みされる他のドキュメントのパスを作成できるようにしなければなりません。

- たとえば、次のような関係にあるドキュメントAとドキュメントBは相互に関連を作成可能です。
- ドキュメントAがフィールドにドキュメントBのリファレンスを持ち、ドキュメントBがフィールドにドキュメントAのリファレンスを持つ
 - ドキュメントAがフィールドにドキュメントBのリファレンスを持ち、ドキュメントBがドキュメントAのサブコレクションに格納される (またはその逆)
 - ドキュメントAとドキュメントBが同一のIDを持つ

多くのケースでは、セキュリティルールのためにリファレンスフィールドを用意することで関連を作ることができますが、一般的にはセキュリティルールのためにドキュメントの設計を変えるのは避けたいでしょう。ルールで保護できない書き込みオペレーションは、Cloud Functionsなどを利用して更新を行います。

4.5.2 具体例

ここではショッピングサイトの商品レビュー機能を例に、クライアントからの一括更新のルールを解説します。

商品ドキュメントには、ユーザーによって投稿されたレビューの総数を保持するフィールド (reviews) を用意しています。Firestoreでは集計オペレーションがサポートされていませんが、一度件数をカウントするとドキュメントの件数分の読み取りオペレーションが発生してしまい、コストが膨大になってしまいます。Firestoreでは、集計の代替としてアトミックオペレーションによって集計結果を都度更新していく方式が用いられます。

商品: /products/product

```
<id>: {
  name: "高級猫大福",
  imageUrl: "https://cdn.shopify.com/...",
  description: "高級な猫大福です",
  price: 600,
  reviews: 10,
  createdAt: 2019-05-01 00:00:00,
  updatedAt: 2019-05-05 00:00:00
}
```

レビュードキュメントはユーザーが投稿したレビューを表します。1件の商品に対して複数のレビューが投稿されるため、レビュードキュメントは商品ドキュメントのサブコレクションに格納します。また、1人のユーザーは商品1件につき最大で1つまでのレビューを投稿できるようにします。この制約を表現するために、レビューのドキュメントIDにはAuthenticationのアカウントIDを利用します¹⁾。

レビュー: /products/product/reviews/uid

```
<id>: {
  user: {
    ref: /users/<uid>
  },
  comment: "最高の味",
  stars: 5,
  createdAt: 2019-05-01 00:00:00,
  updatedAt: 2019-05-05 00:00:00
}
```

新規のレビューを作成するときは、次のような操作をアトミックに実行します。

- 商品ドキュメントのレビュー数を1増やす
 - レビュードキュメントを作成する
- この操作に対するセキュリティルールはリスト4.26のようになります。

リスト 4.26: 新規レビュー投稿時の検証

```
service cloud.firestore {
  // .....
  // 商品のパスを作成する
```

1) ユーザーごとに1件のドキュメントを保持する場合はAuthenticationのアカウントIDをドキュメントIDとして利用すると便利です。

Firestore で実際にショッピングサイトを実装

また、ユーザー管理については第6章で説明した内容のものをそのまま流します。ユーザーの情報は、非公開の会員情報ドキュメントと公開プロフィールドキュメントによって管理されているものとします。

7.2 買い物かご

商品をかごに入れるまでの機能です。Firestore上の操作としては比較的シンプルなので、スキーマ設計に重点を置いて見ていきます。

一般的な要望として、かごの状態はマルチデバイスでの利用や複数タブで開かれたときでも最新のデータを同期したいというものがあります。ここでは、かごの中身データに対する読み取りオペレーションにリアルタイム・リスナーを採用することで対応します。この節では、リアルタイム・リスナーの利用を念頭に置いて設計を進めていきます。

7.2.1 『かごの中身』を設計する

かごの中身ドキュメントは「誰が」「どの商品を」「何個買うのか」という情報を管理するドキュメントです。

所有者の情報はドキュメント内には持たず、ユーザーアカウントに関連付けられたドキュメントのサブコレクションとすることで表現します。フィールドには購入する商品のリファレンスと数量を持ち、さらに取得時のソート用にドキュメントの更新時刻をもつことにします。ドキュメントのIDには商品ドキュメントのIDを流します。

かごの中身ドキュメント: `/users/{uid}/items-in-cart/{item}`

```
{<doc>: {
  product: {
    ref: /databases/default/documents/products/{<id>
  },
  quantity: 1,
  createdAt: 2019/08/12 10:00:00 UTC+9,
  updatedAt: 2019/08/12 10:00:00 UTC+9
}
```

コレクション

買い物かごはユーザー毎に存在します。また、かごの中には2種類以上の商品が同時に入っている状態があるため、ユーザーとかごの中身のリレーションは1:nとなります。ユーザーとの関連を表現しながら、かごの中身ドキュメントを保持するコレクションの配置に関しては次にあげる3つの候補があります。

- ・独立したルートコレクション (フィールドにユーザーを特定するためのリファレンスを持つ)
- ・会員情報ドキュメントのサブコレクション

・公開プロフィールドキュメントのサブコレクション

コレクショングループの登場により、クエリの観点でのルートコレクションの優位性は失われました。サブコレクションを利用した実装が可能であるにもかかわらず、ルートコレクションを選択するのはドキュメントIDの一意性に依存したロジックが存在する(つまり、サブコレクションを利用してコレクショングループに対するクエリを発行した際に万一でもドキュメントIDが重複すると困る)ケースに限定されます。

今回の要件ではそもそもコレクショングループの利用は発生しませんし、ルートコレクションを利用すると、クエリによるフィルタリングのために余計なフィールドを持たせなければなりません。不必要にフィールドが増えちゃう一方で、特段のメリットがないためルートコレクションの利用は避けず。

会員情報と公開プロフィールのどちらのサブコレクションにするかは、データの性質と他の業務を考慮したデータの関連で判断します。

サブコレクションに対するセキュリティルールは上位コレクションと独立して記述することができますが、かごの中身は他のユーザーに公開しない情報ですので、同じく非公開情報である会員情報ドキュメントのサブコレクションに保持するほうが直感的です。

また、後述する商品購入フローにおいて「購入された商品を発送する宛先」を取得する必要があります。かごの中身と会員情報はより業務上の関連が強いと考えられるため、この観点でも会員情報ドキュメントのサブコレクションの利用が適していると考えられます。

かごの中身のドキュメントID

「商品をかごに追加する」という操作は次のような手順になります。

- ・商品が既にかごに入っている場合は、かごの中身ドキュメントの購入数量を増加させる
- ・そうでなければ新たにかごの中身ドキュメントを作成する

この操作には、サブコレクション内のすべてのドキュメントを見て「既にかごに商品が入っているかどうか」を調べる必要があります。しかし、「商品がすでにかごに入っているかどうか」はセキュリティルールによって走査することができます。先述の単純な実装では安全なcreate/updateオペレーションを記述することができます。Cloud Functionsを利用する必要が出てきてしまいます。

この問題は、ある商品をかごに追加しようとしたときに更新される可能性があるドキュメントを1つに絞ることで解決できます。「同一IDを用いた1:nリレーション」パターンを採用し、ある商品をかごに追加する際のかごの中身ドキュメントのIDを固定します。かごに入れる商品ドキュメントのIDからかごの中身ドキュメントに割り当てるIDを決定できるようにすることで、商品がすでにかごに入っているかどうかの判定に際してコレクション内のドキュメント全てを走査する必要がなくなり、オペレーションをセキュリティルールで安全に保護することができます。

商品データの非正規化の検証

かごの中身一覧画面や商品購入フローの確認画面などでは、かごの中身ドキュメントから実際に購入する商品の情報を取得することになります。かごにいれた商品の情報は、購入プロセスにおいて価格や在庫数などその商品の最新の状態を参照する必要があります。したがって、かごの中身

<<目次>>

第1章 Firestoreの正体

- 1.1 Firestoreとはなにか
- 1.2 Firestoreがもたらす変化
- 1.3 特徴的な機能
- 1.4 Cloud Functions
- 1.5 ロケーション

第2章 データアクセスの基礎

- 2.1 Firestoreのデータモデル
- 2.2 単一のドキュメントに対する操作
- 2.3 クエリによるデータの取得
- 2.4 リアルタイム・リスナー
- 2.5 アトミックオペレーション

第3章 オフラインモード

- 3.1 オフラインモードの有効化
- 3.2 オフライン時の書き込みオペレーション
- 3.3 オフライン・データの読み取り
- 3.4 リアルタイム・リスナー
- 3.5 キャッシュ優先読み取り

第4章 セキュリティルール

- 4.1 安全なセキュリティルールを記述するための原則
- 4.2 ユーザー認証

- 4.3 スキーマ検証
- 4.4 データのバリデーション
- 4.5 アトミックオペレーションのセキュリティルール
- 4.6 セキュリティルールのテスト
- 第5章 Firestore データモデリング
 - 5.1 ドキュメント設計の原則
 - 5.2 Firestore での 1:1 リレーション
 - 5.3 Firestore での 1:n リレーション
 - 5.4 コマンドクエリ責務分離
- 第6章 Firestore でユーザーを管理する
 - 6.1 ユーザー管理機能概要
 - 6.2 サインアップ
 - 6.3 会員情報を更新する
 - 6.4 ユーザーの退会処理
- 第7章 Firestore でショッピングサイトを実装してみる
 - 7.1 アプリケーション要件概説
 - 7.2 買い物かご
 - 7.3 商品購入
 - 7.4 レビュー機能

<<著者紹介>>

福田 雄貴

Firestore を完全に理解した Web アプリケーションエンジニア。レガシーコードを片っぱしから爆破したり、技術スタックのモダン化に取り組んだり、マネージド・サービスに最適化した Web アプリケーションアーキテクチャをひらめいたりする仕事をしている。最近のテーマは「大規模アプリケーションの開発・運用にかかるリソースの最小化」。Firestore をはじめとする新しい技術を積極的に取り入れながら、テクノロジーの変化に伴うフロントエンドとバックエンドの境界の再定義に思いを馳せている。

<<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 Kinopyy、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たい開始されます。

※ 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ィア&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