GraqphQL のすゝめ

米藤智哉

2020 年 6 月 23 日

1　Web API 設計

Web API(Web Application Programming Interface) とは，どのように Web サービス上で外部のプログラムを呼び出して利用するかの仕組みである．現代では，Amazon や楽天での買い物や Twitter でのつぶやきなどネットワーク上で様々なデータのやり取りが行われている．では，こういったサービスを利用する上で自分の買い物履歴はどのように表示されていて、他人のつぶやきをどのように表示されているのでしょうか．そういったデータの表示のために様々な端末感でデータ (リソース) のやり取りを行うために Web API が存在します．それでは，これから Web API を設計するために登場した代表的な技術を古い順に紹介していこう．

1.1　RPC

RPC(Remote procedure call，遠隔手続き呼び出し) とは，ネットワークを通じて他の端末上のプログラム (Oracle だとストアド・サブプログラム) を実行させる手法である。クライアントとサーバー間には OS やハードウェアなどの差異は問題とならず，クライアント・サーバーシステムでは，クライアントからサーバーに対してリクエストを送り，サーバーからクライアントに対してレスポンスを返す．

1.2　SOAP

SOAP(Simple Object Access Protocol 諸説あり) とは，HTTP プロトコルを用い，XML ベースの WSDL(Web Services Description Language) にエンコードされたメッセージを伝送するプロトコルである．SOAP メッセージは，構造化された情報をまとめたエンベロープと呼ばれるタグ内に，ヘッダやボディなどの制御情報を付加し，異なる端末間で送受信することを可能とする．

1.3　REST

REST(Representational State Transfer) とは，Web を使ったクライアント・サーバーシステムの設計思想である．上記 2 つと異なり，REST は厳密な手法ではなく，こうすると良いという思想やスタイルに近いものだと考えてほしい．
REST はリソースを扱うための思想である．REST では，以下 4 つの HTTP メソッド (アクション) は状態を遷移させる操作であり，どのようなリソースを保持しているかまたは保持していない

かを状態として表している．

- GET リソースの取得
- PUT リソースの更新
- POST リソースの新規作成
- DELETE リソースの削除

2　複雑化する Web 開発に立ち向かう

前章で述べた技術は Web API 設計技術において大きな功績を作り上げた．しかし，それらの技術の登場には以前の技術に対しての不満があったからに他ならない．時代が進むにつれて，Web サービスはそのコンテンツを大きく増やし，複雑化させた．それによって，既存の手法では面倒なことが増えてしまった．SOAP で作られたサービスは REST に置き換わっていき，XML は書きにくいと JSON に置き換わった．
結局のところ，面倒なことは改善しようという考えが技術を発展させてきたのだ．

3　GraphQL の登場

GraphQL は 2015 年に Facebook によって仕様を公開されたクエリ言語である．Facebook は RESTful サーバーによって運用されていたが，モバイル版 Facebook を作成する際に十分な性能と安全性を出せずにいた．そのため，Facebook エンジニアたちは現行の RESTful サーバーの問題点を挙げた．

3.1　REST の短所

REST には以下の短所があった．

1. 複数のエンドポイント
2. オーバーフェッチ/アンダーフェッチ
3. バージョン管理
4. 弱い型付け
5. 不明なレスポンスデータ構成

3.1.1　複数のエンドポイント

RESTful サービスでは，リソースと URL は 1 対 1 対応の関係である．これにより，URL から端的に操作を予測することができるが同時にデメリットを生む．複数のリソースへと問い合わせる場合は複数のエンドポイントを呼び出す必要があるのだ．例えばではあるが，

```
# 記事の全取得
http:// ・・・ /articles
# 特定記事の取得
http:// ・・・ /articles/123
```

```
# 特定記事のコメントを全取得
http:// ・・・ /articles/123/comments
# 特定記事の特定コメントを取得
http:// ・・・ /articles/123/comments/1
```

のように，指定する URL によってリソースを管理できるが，articles の id 1 〜 5 のような部分取得は複数のエンドポイントを跨ぐ必要があり，URL も長くなってしまう．これはコンテンツの増加に伴い，必要なエンドポイント URL が増加すること，つまり API の維持の難化を示す．

3.1.2 オーバーフェッチ/アンダーフェッチ

3.1.3 バージョン管理

3.1.4 弱い型付け

3.1.5 不明なレスポンスデータ構成

4 GraphQL の仕様

内容はまだ

4.1 クエリとスキーマ

5 React + Apollo Cliente GraphQL

React 開発の話は別ドキュメント内容はまだ

5.1 GraphQL code generater

内容はまだ