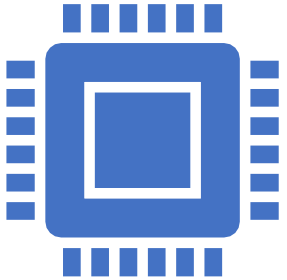
最近のモダンなREST API開発 ～APIファースト /OpenAPI/etc～



2021/3/3

第4回 みんなde課外活動（仮）

はじめに

- 本日のテーマ
  - 「最近のモダンなREST API開発」と題して、RESTAPIの仕様と実装の架け橋？繋ぎとして昨今、よく用いられるOpenAPIとは「なにものか？」「どんな風に使うのか？」「なにがうれしいか？」について解説させていただきます。

- 本日の狙いは！
  - 外部連携を行う開発では、接続条件であるプロトコル（メッセージ送信手順、メッセージフォーマット）を対向システムと如何に認識齟齬なく取り交わすかが1つのキモ
  - 外部連携方式の1つであるREST API開発でOpenAPIを用いることで、その辺がどのように正確性と効率化に寄与するかを理解してもらう

早速ですが・・

- こんな機能をREST APIとして作ってと言われたら、皆さんだったら仕様確定に向けまずはどんな資料や成果物を作りますか？

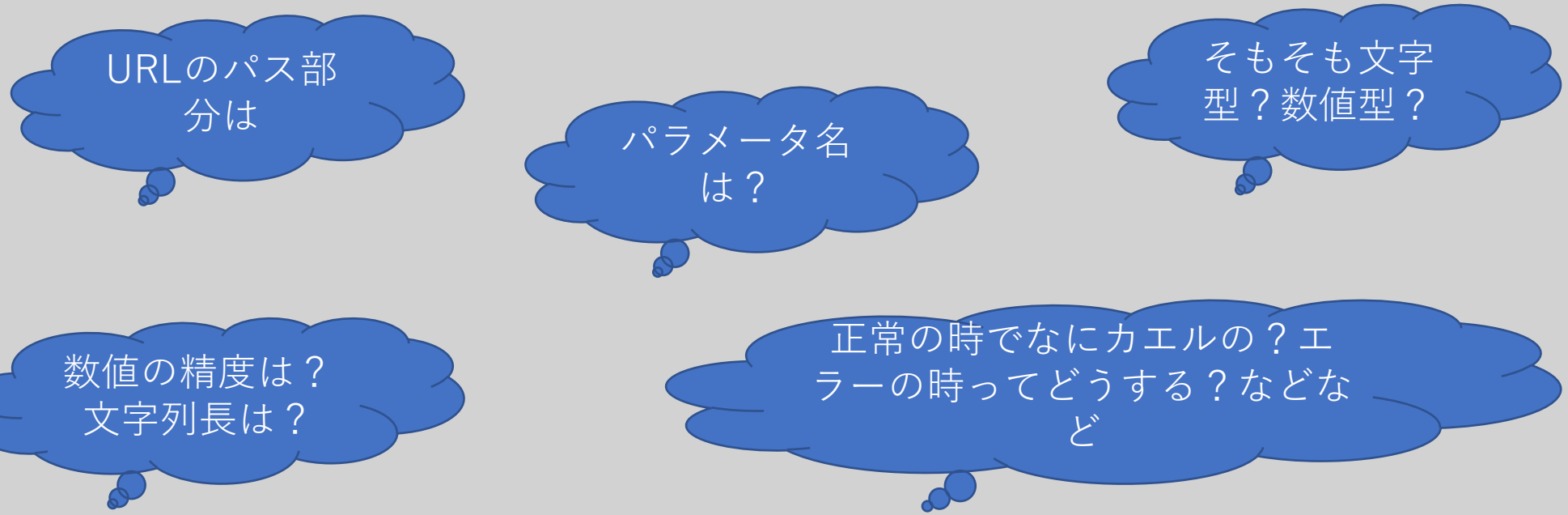
＜要求＞
・会員価格を計算するAPIをRESTで公開したい

＜前提＞
・会員には一意な会員番号が振られている
・会員には一般会員、シルバー会員、ゴールド会員の会員種別がある
・会員種別は会員番号だけで判別することができる
・会員種別ごとに割引率が設定されてる

＜API仕様＞
・会員番号と金額を入力された会員種別に応じた割引が適用された後の金額を返す
・割引率は、一般：10%、シルバー：20%、ゴールド：30%となっている
・端数は切り捨てでよく、消費税も考慮する必要なく、単純に入力金額 * 割引率の値を返せばよい
・なお、結果は割引後の金額の他に会員種別と割引率も合わせて返却する

やっぱり

- エクセルですよね！でも・・



- が、どれも普通はエクセルでインタフェース仕様なりを作って確認してますよね

インターフェース仕様書					
会員価格を計算するREST API					
パス		/products/pricecalc?memberNo=会員番号&price=価格			
No	種別	項目名（論理）	項目名（物理）	型	説明
1	IN	会員番号	memberNo	文字列	・数値＋大文字アルファベットのみの5桁
2	IN	価格	price	整数	8桁の正数（マイナスはない）
3	OUT	結果リスト	リスト		
		価格	(1番目)	整数	割引適用後の価格
		会員種別	(2番目)	整数	1:一般、2:シルバー、3:ゴールド
		割引率	(3番目)	整数	パーセント表記の数値

みんな大好きエクセル仕様書

エクセルって・・

- どんなに頑張っても、、抜け漏れ、typo、実装側の読み間違え、書き間違えなどで、実際にテストするとアレヨアレヨと「認識齟齬！」と言う名の不具合がでてきます。
- が、これは仕様書を使った人がアレなのでしょうか？それとも実装している人の注意不足なのでしょうか？
- 人間なので、プログラムに近い詳細な情報を間違なく処理するのはやっぱり限界。
- で、よくよく考えれば、そんなに細かい情報を定義するなら、実装に繋げられるような成果物で仕様を作って確認した方がいいよね！と言うことで出てきたのが、

OpenAPI！

OpenAPIとはどのようなものか？

- 簡単にいうと、REST APIの仕様を語る上で必要となるような要素をモデル化し、その内容を表現するのに必要な構造とタグ名（正確には項目名）を規定し、それをyamlまたはjsonで作成する方法を定めたオープンな規格
  - モデル（次スライドの例を見ながら・・）
  - タグ名（次スライドの例を見ながら・・）

YAMLによるOpenAPISpecification(OAS)のサンプル

```yaml
info:
  contact:
    name: 豆ちゃん
    url: https://www.mamezou.com
  title: 施設予約システムの公開API
  version: 0.0.1-SNAPSHOT
openapi: 3.0.1
paths:
  /ers/equipments:
    post:
      description: 依頼された施設を登録する
      requestBody:
        content:
          application/json:
            schema:
              properties:
                capacity:
                  format: int32
                  type: integer
                extentionNumber:
                  type: string
                name:
                  type: string
              type: object
```

（続き）

```yaml
      responses:
        '400':
          $ref: '#/components/responses/ParameterError'
        '200':
          content:
            application/json:
              schema:
                items:
                  $ref: '#/components/schemas/Equipment'
                type: array
          description: 登録成功
      summary: 施設を登録する
```

結論：OpenAPIとは

「REST APIの仕様をOpenAPIで定義する」とは、つまるところOpenApiSpecificationの書式でRESTAPIの仕様をYAML(or JSON)で作成するということ

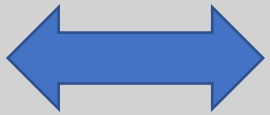
では先ほどの仕様をOASで作成してみよう！

ちょっとその前に少しだけYAMLの書き方の説明

- ファイルの拡張子は”.yml” or “.yaml”
- データの階層をインデント2つで表す
- 項目と値の書式は”項目名:値”となる
- 配列要素は項目名の前に”-” を付ける（ちょっと癖がある）

プロパティ形式

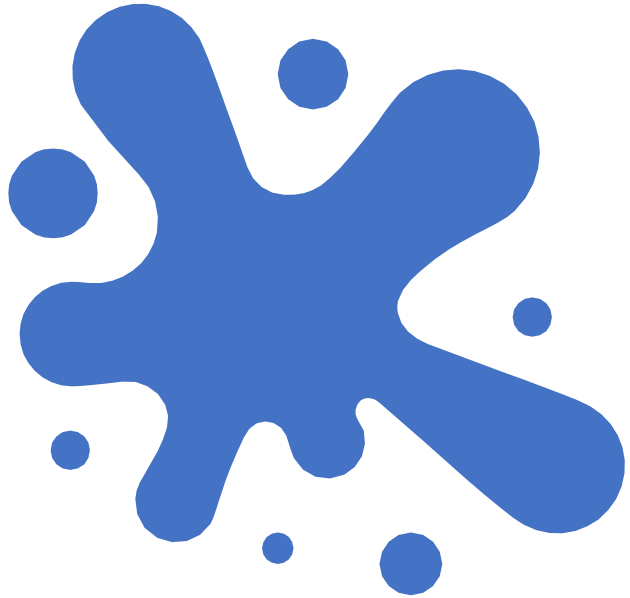
```
connect.dest=foo
connect. param=bar
connect. param.detail.0.key=key1
connect. param.detail.0.value=val1
connect. param.detail.1.key=key2
connect. param.detail.2.value=val2
```



等価

YAML形式

```
connect:
  dest: foo
  param: bar
  detail:
    - key: key1
      value: val1
    - key: key2
      value: val2
```



ではやってみましょう！

↓へアクセス！

https://editor.swagger.io/

今やったことの整理

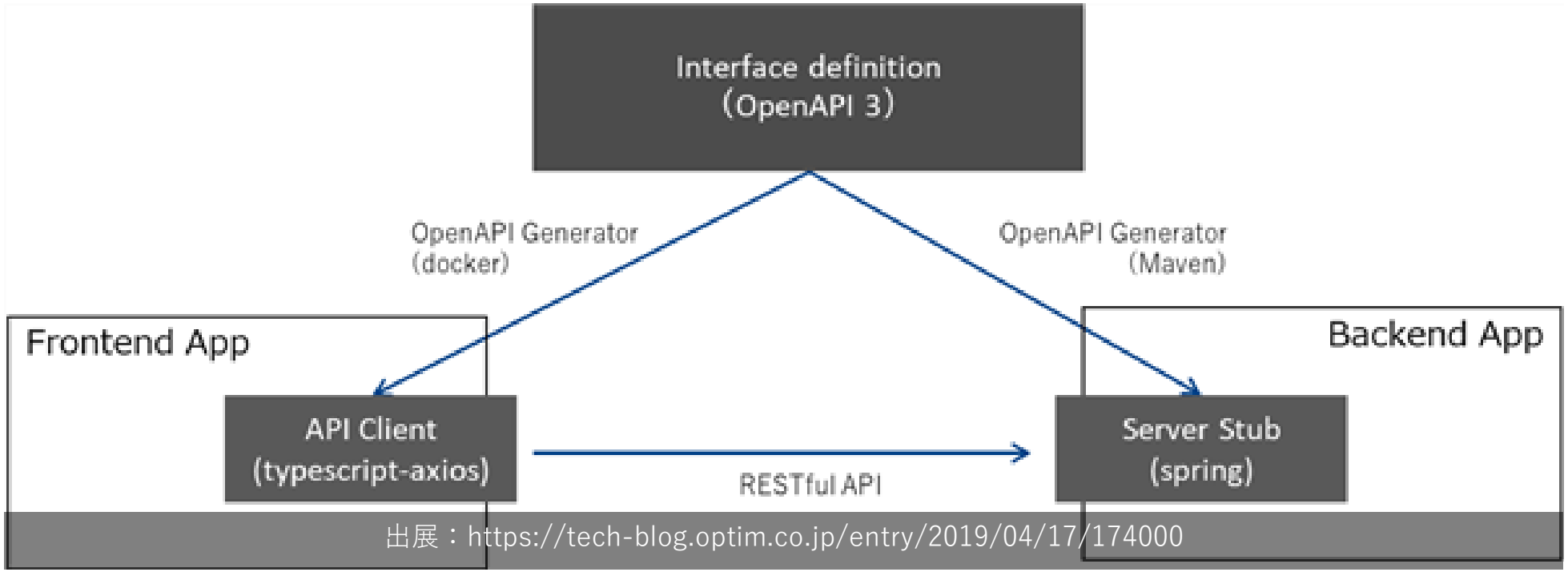
- REST APIの仕様をSwaggerEditorを使ってOpenAPIドキュメントとして（さわりだけ）作ってみた
  （仕様の提供側⇒サーバ側の人がやること）
- OpenAPIドキュメントを貰ったらSwaggerUIを使って確認してみた
  （仕様の受取側⇒クライアント側の人がやること）

改めてOpenAPISpecificationとは（固いお話し）

- Swaggerが主導で主にREST APIの仕様定義方法の標準化を行う標準化団体であるOpenAPI Initiative(OAI)を設立
- OAIの成果としてREST APIの仕様定義のOpenAPISpecification（OAS）を発表
- OASはREST APIの仕様をコンピュータでも処理できるようにYAMLまたはJSONで記述するのが特徴で、その記述を入力としてドキュメントやスケルトンコードを生成するツールが沢山出てきている
- また、RESTは異機種間通信が普通なので、OASで言語ニュートラルな仕様を定義し、それを入力しとして様々なプラットフォームや言語で使えるのが便利
  - OAS定義を入力としてクライアント向けにはJSのスケルトン、サーバ向けにはJavaのスケルトンを生成といったことができる
- と非常に便利なので、現在は仕様から~~エクセルではなく~~**OASを中心に開発を進めて行くことが一般的に行われている**

ところで・・

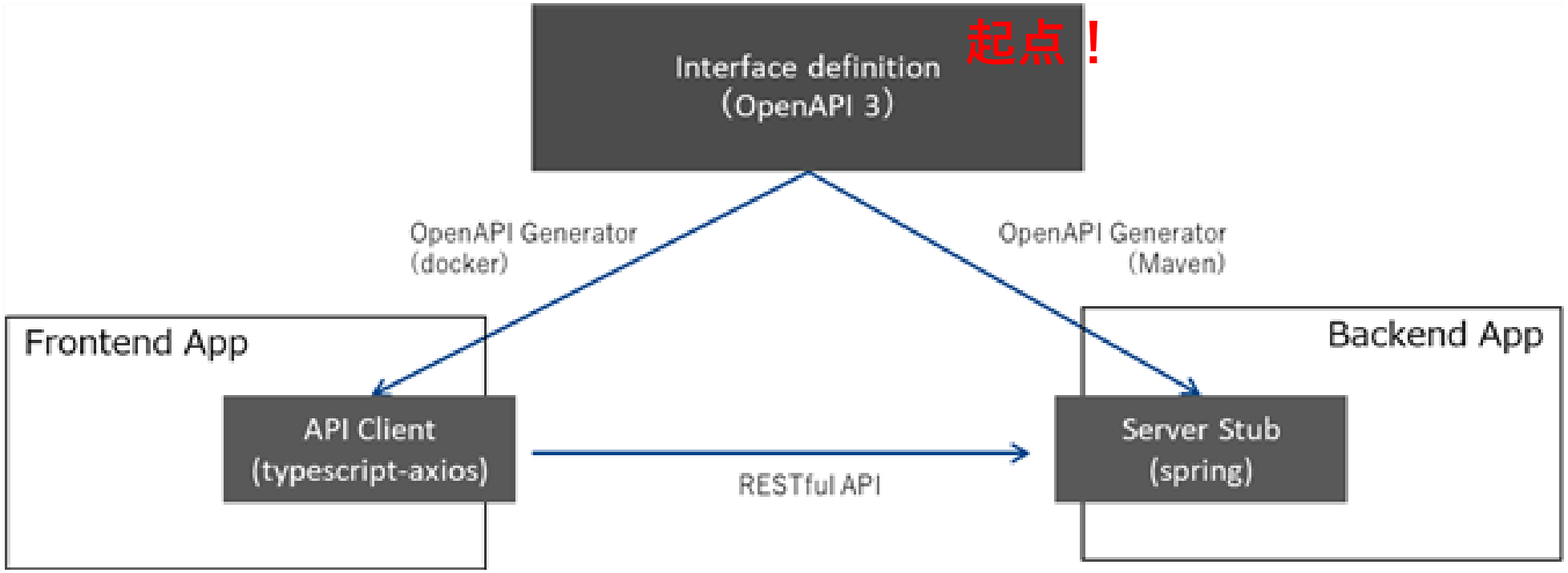
- 正確な仕様記述に寄与するのは分かったけど、なにが効率的になったの？
- OpenAPIドキュメントはオープンな仕様なので、それを入力としてJavaインタフェースやスケルトンコード、とりあえず何か応答を返すモックサーバを生成するツールがいくつかあり、それを使うことでコードの自動生成が可能です！
  - swagger codegen（←オワコン）
  - OpenAPI Generator（←今はコッチ）
    - https://github.com/OpenAPITools/openapi-generator



OpenAPITools で出来ること

- OpenAPITools はOpenAPIドキュメントを入力としてクラアントとサーバそれぞれの実装をサポートする任意の言語で生成することが可能
- ・なので1つのOpenAPIドキュメントを入力として、クラアントサイドはJavaScript、サーバサイドはJavaで生成などと言ったことも可能

まとめ



再び改めてOpenAPIドキュメントとは

- クライアントとサーバが同じOpenAPIドキュメントを起点として開発を行うことで、ズレなく効率的な開発をすることができる！🙌

こぼれ話・・

- 用語の補足
  - OpenAPISpecification(OAS)に準拠したファイルを一般的にOpenAPIドキュメントと呼びます
  - ですので、正確にはOASは仕様で、それを実現したファイルがOpenAPIドキュメントとなります。が、、一般的には区別することなく双方の意味で双方が使われています
- そんな素敵なOpenAPIドキュメントですが
  - 実際に開発で使えるレベルの精度の良いコードを生成させるには、プログラムと等価なレベルでOpenAPIドキュメントを定義する必要がありすごく大変です。（なので、無邪気なOAS原理主義は危険）
  - クライアントとサーバの双方が同じ言語なら、Javaのインタフェース相当のものを双方で共有した方が効率的
- モデルドリブンは幻想か？

そういえばタイトルのもう一つは？

- APIファースト
  - 実は何をもってAPIファーストと言うことは、なにをもって「アジャイル」と言うかと同じくらい、その定義はなく、いろいろな人が色々なコンテキストでいろいろなことを言っている（と思っている）
  - とは言え、つまるところみんな「APIを中心にその組み合わせでアプリを作っていこう！」的なことを言っているので、多分そのような気持ちのことを言うと思う
  - その実践にはOpenAPIはキッと必要で役に立つと思います！

ご静聴ありがとうございました