

Symfony4とAPI Platformを使って5分でAPIサーバを作る

PHP Symfony Symfony4 API-Platform

この記事は最終更新日から3年以上が経過しています。

API PlatformはオープンソースのAPIフレームワークです。Symfony上で動作し、ちょっとした設定で簡単にAPIサーバを構築することができます。ここではSymfony4とAPI PlatformでAPIサーバを構築する手順をご紹介します。タイトルの通り、サクッと5分ほどで構築できます。

1-2分：インストール

インストールはcomposerを利用します。APIサーバを構築するだけであればapiのみで必要なものは全てインストールできますが、ここでは動作確認用と作業効率向上のため、makerとserverを追加でインストールしています。

もろもろインストール

Symfony4プロジェクトを作成

```
composer create-project symfony/skeleton symfony4-api-sample  
cd symfony4-api-sample
```

API Platformフレームワークインストール

```
composer req api
```

Entityをコマンドラインから作成するためにmakerをインストール

```
composer req maker
```

動作確認用のサーバコマンドをインストール

```
composer req server --dev
```

3分 : DB設定

今回はSQLiteを利用しました。ので、.envをちょっと修正します。

.env

```
###> doctrine/doctrine-bundle ###
```

```
... # Format described at http://docs.doctrine-project.org/project
```

```
... # For an SQLite database, use: "sqlite:///kernel.project_dir%
```

```
... # Configure your db driver and server_version in config/packages
```

```
... - DATABASE_URL=mysql://db_user:db_password@127.0.0.1:3306/db_name
```

```
+ DATABASE_URL=sqlite:///kernel.project_dir%/var/data.db
```

```
###< doctrine/doctrine-bundle ###
```

修正後、DBを作成します。

DB作成

```
bin/console doctrine:database:create
```

4分：Entity作成

今回はProductというEntityを作成しました。フィールドは

- name(string)
- price(integer)
- isAvailable(bool)

を用意しました。

Entity作成

```
bin/console make:entity
```

```
Class name of the entity to create or update (e.g. OrangeChef  
> Product
```

```
created: src/Entity/Product.php
```

```
created: src/Repository/ProductRepository.php
```

Next: When you're ready, create a migration with `make:migration`

作成したEntityにAnnotationを追加します。

Product.php

```
<?php
```

```
namespace App\Entity;
```

```
use Doctrine\ORM\Mapping as ORM;
```

```
+ use ApiPlatform\Core\Annotation\ApiResource;
```

```
/**
```

```
 * @ORM\Entity(repositoryClass="App\Repository\ProductRepository")
```

```
+ * @ApiResource
```

```
 */
```

```
class Product
```

```
{
```

作成したEntityを利用してDBにテーブルを作成しましょう。

テーブル作成

```
bin/console doctrine:schema:create
```

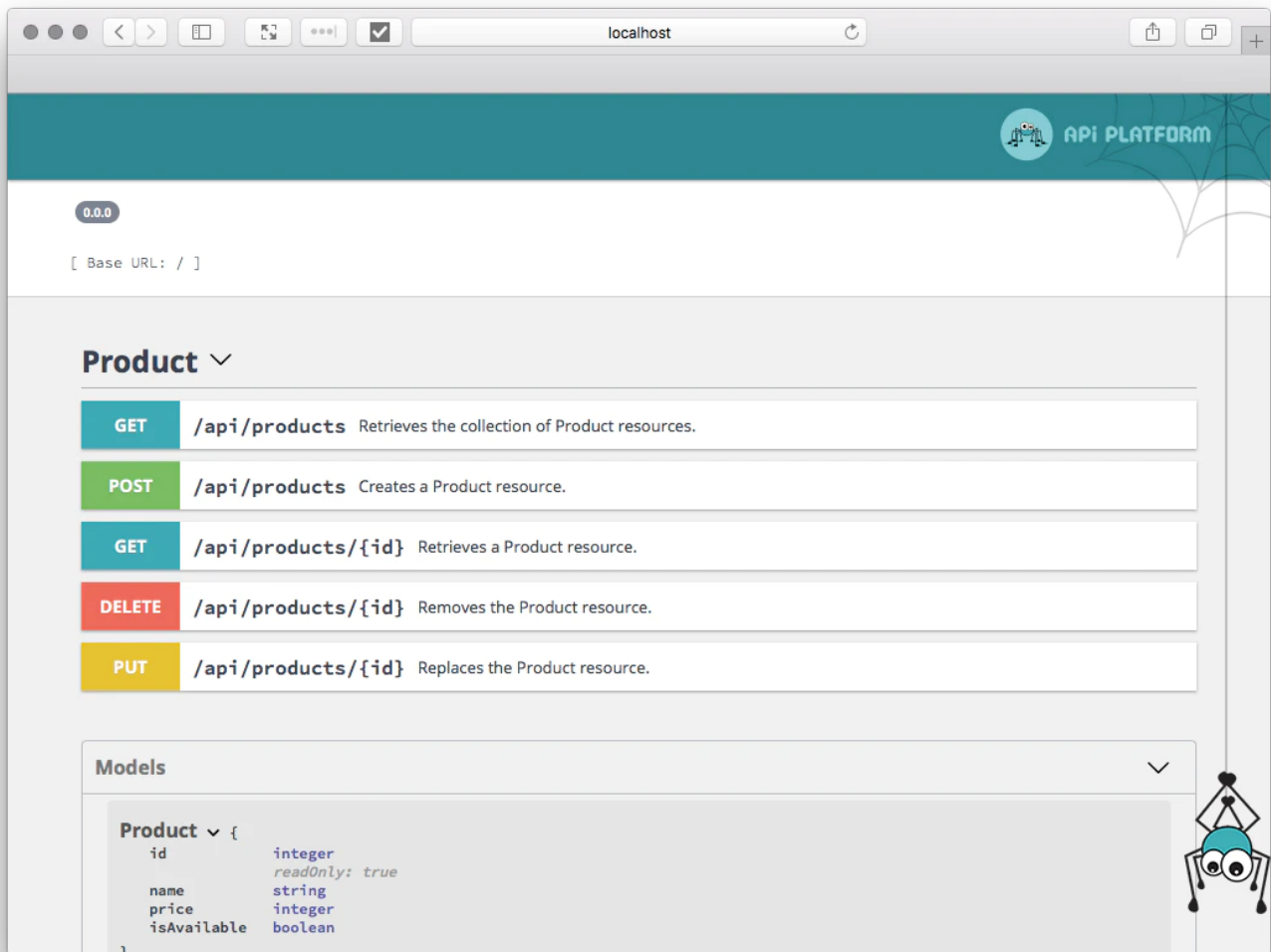
5分：実行

これでProductのデータを制御するAPI群が完成しました。実際に動かしてみましょう。

サーバ起動

```
bin/console server:run
```

<http://localhost:8000/api> にアクセスすると...



できました！Productを制御するためのAPIのリストと利用方法を記載したページができ上がりました。では、実際にAPIを実行してみましよう。ここでは登録とリスト取得を実行してみます。

curlで登録

```
curl --request POST \  
  --url http://localhost:8000/api/products \
```

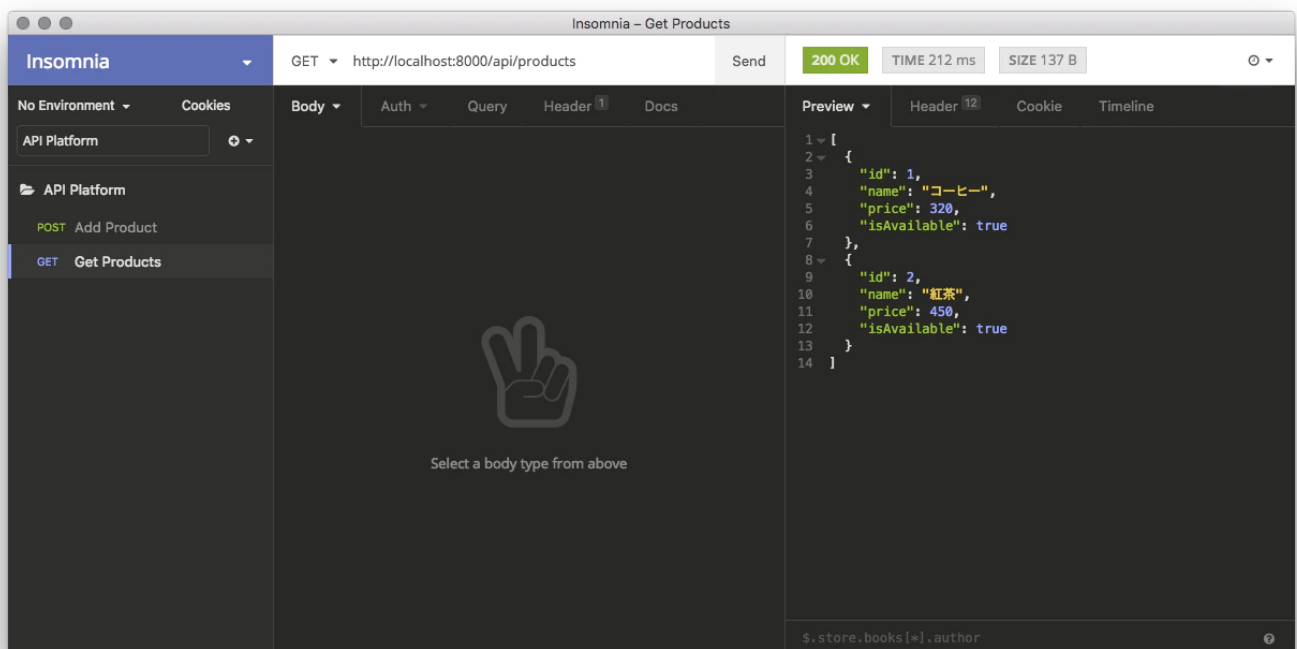
```
--header 'accept: application/json' \  
--header 'content-type: application/json' \  
--data '{  
    "name": "紅茶",  
    "price": 450,  
    "isAvailable": true  
}'
```

response

```
{  
    "id": 2,  
    "name": "紅茶",  
    "price": 450,  
    "isAvailable": true  
}
```



```
{  
    "id": 2,  
    "name": "紅茶",  
    "price": 450,  
    "isAvailable": true  
}  
]
```



まとめ

こんな感じでサクッとAPIサーバが構築できます。設定などでOAuth2を仕込んだりバリデーションを設定したりなどカスタマイズできるので、ぜひ一度お試しください。ここまでのプロジェクトをGitHubに用意しましたので、こちらをご覧ください。

<https://github.com/lppey/symfony4-api-sample>