

## API REST



|                                                             |          |
|-------------------------------------------------------------|----------|
| <b>1 Mise en place des données .....</b>                    | <b>2</b> |
| 1.1 .....                                                   | 2        |
| 1.2 .....                                                   | 2        |
| 1.2.3 .....                                                 | 3        |
| 1.2.4 .....                                                 | 3        |
| <b>2 Déploiement d'une API REST avec API Platform .....</b> | <b>4</b> |
| 2.1 .....                                                   | 4        |
| 2.2 .....                                                   | 4        |
| 2.3 .....                                                   | 4        |
| 2.4 .....                                                   | 4        |
| 2.5 .....                                                   | 5        |
| 2.6 .....                                                   | 5        |

# 1 Mise en place des données

## 1.1

Dans un premier temps je créer un projet (symfony new garage -webapp) :

```
mlemoine@PC-F205-14:~/Documents/btssiolemoine/symfony$ symfony new garage -webapp
* Creating a new Symfony project with Composer
* Setting up the project under Git version control
  (running git init /home/mlemoine/Documents/btssiolemoine/symfony/garage)

[OK] Your project is now ready in /home/mlemoine/Documents/btssiolemoine/symfony/garage
```

Puis je créer une bdd (bin/console make entity) :

## 1.2

Je crée owner et car :

Success!

Il est important de créer owner avant car parce qu'il est utilisé dans la table car

### 1.2.3

```
mlemoine@PC-F205-14:~/Documents/btssiolemoine/symfony/garage$ bin/console make:fixture

The class name of the fixtures to create (e.g. AppFixtures):
> GarageFixture

created: src/DataFixtures/GarageFixture.php

Success!

Next: Open your new fixtures class and start customizing it.
Load your fixtures by running: php bin/console doctrine:fixtures:load
Docs: https://symfony.com/doc/current/bundles/DoctrineFixturesBundle/index.html
```

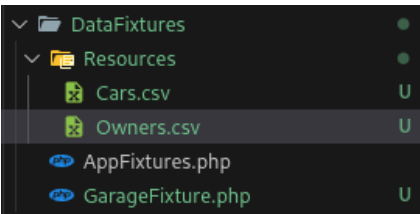
J'ai la créer la premier fixture

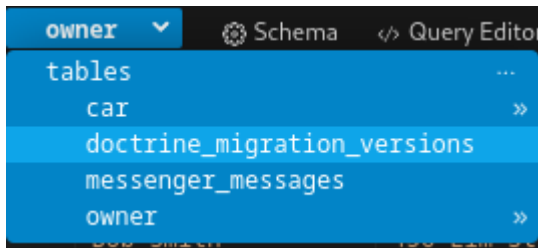
Le fichier qui a été créer est : GarageFixture.php son chemin est src/DataFixtures/GarageFixture.php

```
mlemoine@PC-F205-14:~/Documents/btssiolemoine/symfony/garage$ bin/console doctrine:fixtures:load
Careful, database "main" will be purged. Do you want to continue? (yes/no) [no]:
> yes

> purging database
> loading App\DataFixtures\AppFixtures
> loading App\DataFixtures\GarageFixture

mlemoine@PC-F205-14:~/Documents/btssiolemoine/symfony/garage$
```





Les table car et owner contiennent bien les données :

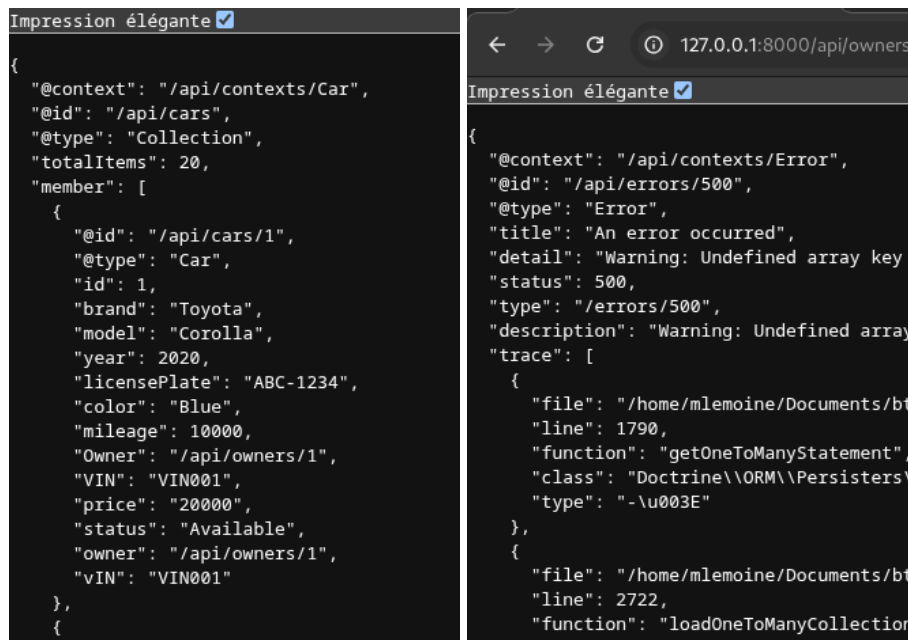
| SELECT * FROM owner |               |                  |                      |                      |              |               |              |                |   |
|---------------------|---------------|------------------|----------------------|----------------------|--------------|---------------|--------------|----------------|---|
|                     | id            | name             | address              | email                | phone        | date_of_birth | national_id  | license_number |   |
|                     | INTEGER       | VARCHAR(255)     | VARCHAR(255)         | VARCHAR(255)         | VARCHAR(255) | DATE          | VARCHAR(255) | VARCHAR(255)   |   |
|                     | PRIMARY KEY   | NOT NULL         | DEFAULT              | DEFAULT              | DEFAULT      | DEFAULT       | DEFAULT      | DEFAULT        | + |
|                     | AUTOINCREMENT |                  |                      |                      |              |               |              |                |   |
| 1                   | 1             | Alice Johnson    | 123 Main St CityA    | alice.johnson@ema... | 123-456-7890 | 1985-01-15    | NID001       | LN001          |   |
| 2                   | 2             | Bob Smith        | 456 Elm St CityB     | bob.smith@email.c... | 234-567-8901 | 1990-02-25    | NID002       | LN002          |   |
| 3                   | 3             | Charlie Williams | 789 Maple St CityC   | charlie.williams@... | 345-678-9012 | 1980-03-10    | NID003       | LN003          |   |
| 4                   | 4             | Dana Clark       | 101 Pine St CityD    | dana.clark@email...  | 456-789-0123 | 1995-04-05    | NID004       | LN004          |   |
| 5                   | 5             | Ethan Brown      | 202 Cedar St CityE   | ethan.brown@email... | 567-890-1234 | 1988-05-20    | NID005       | LN005          |   |
| 6                   | 6             | Fiona Davis      | 303 Birch St CityF   | fiona.davis@email... | 678-901-2345 | 1992-06-18    | NID006       | LN006          |   |
| 7                   | 7             | George Evans     | 404 Oak St CityG     | george.evans@ema...  | 789-012-3456 | 1983-07-23    | NID007       | LN007          |   |
| 8                   | 8             | Hannah Foster    | 505 Redwood St Ci... | hannah.foster@ema... | 890-123-4567 | 1997-08-12    | NID008       | LN008          |   |

## 1.2.4

L'intérêt de la variable est de stocker les objets, Ils servent uniquement à faire correspondre les relations.

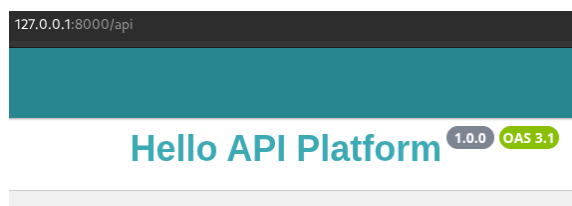
## 2 Déploiement d'une API REST avec API Platform

### 2.1



On découvre les info de la base de données

### 2.2

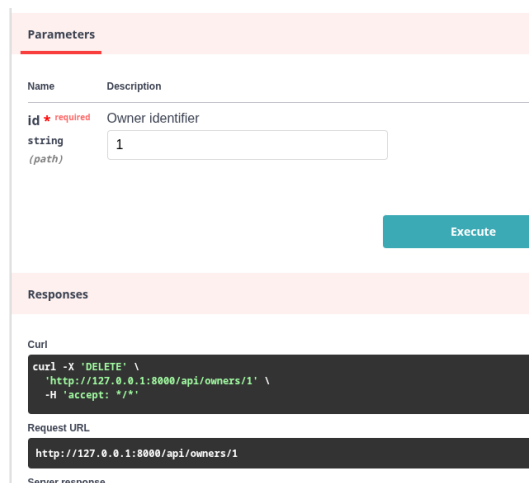


On tombe sur l'api platform qui permet de modifier en direct la base de données

### 2.3

Les 5 types de requête possible via API REST sont : GET, POST, PUT, PATCH et DELETE

### 2.4



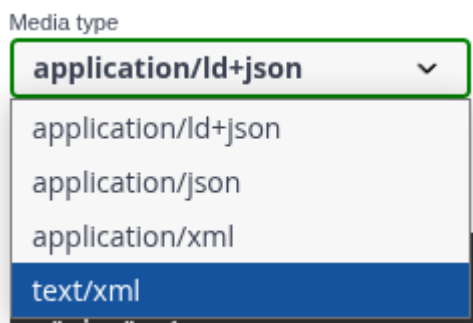
La requête est : `curl -X 'DELETE' \ 'http://127.0.0.1:8000/api/owners/1' \ -H 'accept: */*'`

## 2.5

|                                           |                                                   |
|-------------------------------------------|---------------------------------------------------|
| <b>Car</b> Resource 'Car' operations.     |                                                   |
| GET                                       | <code>/api/cars</code> Retrieves the collection   |
| GET                                       | <code>/api/cars/{id}</code> Retrieves a Car       |
| <b>Owner</b> Resource 'Owner' operations. |                                                   |
| GET                                       | <code>/api/owners</code> Retrieves the collection |
| GET                                       | <code>/api/owners/{id}</code> Retrieves a Owner   |

On passe uniquement en guete ce qui sécurise l'api

## 2.6



Cela permet de l'avoir soit en json ou xml sans rien faire juste en configurant