Sponsored by Stripe
Simple payments for websites & mobile. Stripe powers payments for a variety of languages and platforms. Learn more about Stripe's API.
Free Signup
Free Upgrade to Pro, Elite
POSTS BY STACK
Linux | PHP | Node | Apache
Ubuntu | Nginx | MySQL | JavaScript
Debian | Redis | Python | Docker
POSTS BY PRODUCT
Vultr | Linode | Hetzner
DigitalOcean | Amazon AWS | Google
Cloudflare Edge | Varnish | Google Web Fonts
Google Cloud | Google Cloud Platform | Google Cloud Storage
CATEGORIES
Cache | Security and Hardening
Continuous Integration
Development Workflow | Plan Your Project
Deployment
Monitoring | Web Stacks | Tip Tuesday
Reviews
Sponsor Tutorial | Sponsor Review
Sign Up | Sign In
Developer Updates
Sign up for Developer Updates
Subscribe to Developer Updates for a monthly digest of all things Linode.
Enter your email
Sign Up
Your Email address will be used to send you the Developer Updates newsletter. You can unsubscribe at any time.
Tutorials
Tutorials
Tutorials and guides that show how to get started with a variety of languages, frameworks, and tools on Linode from scratch.
Get Started

Develop a GraphQL-Powered API With Symfony

For years, **REST** (Representational State Transfer) has been the de facto standard when designing APIs. That's quite understandable, as it is a very straightforward structure. When creating a request to an endpoint, a GETfor GRAB, a POSTfor CREATE, etc.

However, as applications become more complicated and a resource-driven structure is strange - multiple REST calls are required to populate a view (in Ember GraphQL With GraphQL, the sender of the request determines the structure of the response. Thus, providing more flexibility and efficiency for the consumer.

In this article, I will show you how to develop a GraphQL-powered API for an online book store with Symfony.

Prerequisites

To get the most out of this tutorial, you need the following:

- A basic understanding of GraphQL
- Basic experience with PHP and Symfony
- PHP 8.0
- Composer
- The Symfony CLI

Getting Started

Create a new Symfony application, and switch into the newly generated directory, by running the following commands:

```
symfony new graphql_demo --version=7.0
cd graphql_demo
```

To provide integration with GraphQL, the application will use a Symfony bundle named overblog/graphql-bundle. At the time of writing, the overblog/graphql-bundle could not be installed in a Symfony 6 project thus the use of Symfony 6.0.

Next, install the project dependencies. The project will use the following:

1. Doctrine. The Doctrine ORM helps manages the application database.
2. DoctrineFixtureBundle. This helps load authors and books into the database.
3. Faker. Generates fake data for the application.
4. Maker. Simplifies creating controllers, entities, and the like.
5. OverblogGraphQLBundle. This bundle provides integration with GraphQL.
6. OverblogGraphQLBundle. This bundle provides an integration of the GraphiQL interface.

To install them, run the two commands below.

```
composer require doctrine/orm symfony/orm-pack
composer require --dev doctrine/doctrine-fixtures-bundle fakerphp/faker
```

Accept the contrib recipes installation from Symfony Flex by pressing `y`. Then, create a .env.local file from the .env file, which Symfony generated during the creation of the project, by running the command below.

```
cp .env .env.local
```

The .env file is ignored by Git as it stores an existing pattern in .gitignore (which Symfony generated). One of the reasons for this file is to accomplish best practice of storing your credentials outside of code to keep them safe.

Next, update the DATABASE_URL parameter in .env.local so that the app uses an SQLite database instead of the PostgreSQL default. To do that, command-line users can run the command SYMFONY_DEV_MODE=1 symfony dotenv:dump, or find it inside the file manually below.

```
SYMFONY_DEV_MODE=1 symfony dotenv:dump
```

The database will be created in /var/app/data/app.db.

Create entities

The application will have two entities, Author and Book. Create the Author entity first using the following command:

```
symfony console make:entity Author
```

The CLI will ask some questions and add fields for the entity based on the provided answers. Answer the questions as shown below.

```
$ bin/console make:entity Author

created: src/Entity/Author.php
created: src/Repository/AuthorRepository.php

Add your new property to your entity.

New property name (press <return> to stop adding fields):
> firstname

Field type (enter ? to see all types) [string]:
> string

Field length [255]:
>

Can this field be null in the database (nullable) [yes/no]:
> no

Add another property? Enter the property name (or press <return> to stop adding fields):
> lastname

Field type (enter ? to see all types) [string]:
> string

Field length [255]:
>

Can this field be null in the database (nullable) [yes/no]:
> no

Add another property? Enter the property name (or press <return> to stop adding fields):
> dob

Field type (enter ? to see all types) [datetime_immutable]:
> date

Can this field be null in the database (nullable) [yes/no]:
> no

Add another property? Enter the property name (or press <return> to stop adding fields):
> email

Field type (enter ? to see all types) [string]:
> string

Field length [255]:
>

Can this field be null in the database (nullable) [yes/no]:
> no

Add another property? Enter the property name (or press <return> to stop adding fields):
> country

Field type (enter ? to see all types) [string]:
> string

Field length [255]:
>

Can this field be null in the database (nullable) [yes/no]:
> no

Add another property? Enter the property name (or press <return> to stop adding fields):
>

updated: src/Entity/Author.php
updated: src/Repository/AuthorRepository.php

Next: when you're ready, create a migration file with this command:
php bin/console doctrine:migrations:diff

Notice: It looks like you've configured your ORM mapping to be stored in more than one directory.
Unfortunately, this feature is not supported with the ORM Doctrine Bundle.

Success!

Next: Add new properties to your entity, or run:
php bin/console make:entity
```

Press the `Enter` key to complete the creation process for the Author entity.

Next, create the Book entity using the following command:

```
symfony console make:entity Book
```

Respond to the CLI prompt as shown below.

```
$ bin/console make:entity Book

created: src/Entity/Book.php
created: src/Repository/BookRepository.php

Add your new property to your entity.

New property name (press <return> to stop adding fields):
> title

Field type (enter ? to see all types) [string]:
> string

Field length [255]:
>

Can this field be null in the database (nullable) [yes/no]:
> no

Add another property? Enter the property name (or press <return> to stop adding fields):
> author

Field type (enter ? to see all types) [string]:
> relation

What class should this entity be related to?:
> Author

In the Author entity what should this be called?:
> books

Is this an ORM OneToMany relation?:
> yes

Is this a ORM ManyToOne relation?:
> yes

Do you want to add a new property to Author that you can access/update Book objects from it - e.g. $author->getBooks()):
> yes

A new property will also be added to the Author class so that you can access the related Book objects from it.

New field name inside Author [books]:
> books

Do you want to generate an empty getter and setter for this new property? [yes]:
> yes

A new property will also be added to the Author class so that you can access the related Book objects from it.

New field name inside Author [books]:
> books

Do you want to generate an empty getter and setter for this new property? [yes]:
> yes

Add another property? Enter the property name (or press <return> to stop adding fields):
>

updated: src/Entity/Book.php
updated: src/Entity/Author.php
updated: src/Repository/BookRepository.php

Next: when you're ready, create a migration file with this command:
php bin/console doctrine:migrations:diff

Notice: It looks like you've configured your ORM mapping to be stored in more than one directory.
Unfortunately, this feature is not supported with the ORM Doctrine Bundle.

Success!

Next: Add new properties to your entity, or run:
php bin/console make:entity
```

Press the `Enter` key to complete the creation process for the Book entity.

Update the entities

Open the Author entity in the src/Entity/Author.php file and update its content to match the following:

```
<?php
namespace App\Entity;

use App\Repository\AuthorRepository;
use Doctrine\Common\Collections\Collection;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\ORM\Mapping as ORM;

#[ORM\Entity(repositoryClass: AuthorRepository::class)]
class Author
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column(type: "integer")]
    private int $id;

    #[ORM\Column(type: "string", length: 255)]
    private string $firstname;

    #[ORM\Column(type: "string")]
    private string $lastname;

    #[ORM\Column(type: "datetime_immutable")]
    private DateTimeImmutable $dob;

    #[ORM\Column(type: "string")]
    private string $email;

    #[ORM\Column(type: "string")]
    private string $country;

    #[ORM\OneToMany(targets: Author::class, inversedBy: "books")]
    #[ORM\OrderBy(value: "ASC")]
    private Collection $books;

    public function __construct()
    {
        $this->id = null;
        $this->firstname = null;
        $this->lastname = null;
        $this->dob = null;
        $this->email = null;
        $this->country = null;
        $this->books = new ArrayCollection();
    }

    public function getId(): int
    {
        return $this->id;
    }

    public function setId(int $id): void
    {
        $this->id = $id;
    }

    public function getFirstname(): string
    {
        return $this->firstname;
    }

    public function setFirstname(string $firstname): void
    {
        $this->firstname = $firstname;
    }

    public function getLastname(): string
    {
        return $this->lastname;
    }

    public function setLastname(string $lastname): void
    {
        $this->lastname = $lastname;
    }

    public function getDob(): DateTimeImmutable
    {
        return $this->dob;
    }

    public function setDob(DateTimeImmutable $dob): void
    {
        $this->dob = $dob;
    }

    public function getEmail(): string
    {
        return $this->email;
    }

    public function setEmail(string $email): void
    {
        $this->email = $email;
    }

    public function getCountry(): string
    {
        return $this->country;
    }

    public function setCountry(string $country): void
    {
        $this->country = $country;
    }

    public function getBooks(): Collection
    {
        return $this->books;
    }

    public function setBooks(Collection $books): void
    {
        $this->books = $books;
    }

    public function addBook(Book $book): void
    {
        if (!$this->books->contains($book)) {
            $this->books[] = $book;
            $book->setAuthor($this);
        }
    }

    public function removeBook(Book $book): void
    {
        if ($this->books->removeElement($book)) {
            // set the owning side to null (unless already changed)
            if ($book->getAuthor() === $this) {
                $book->setAuthor(null);
            }
        }
    }
}
```

In addition to adding types for the fields of the Author entity, the constructor will add a constructor function that requires the authors's name, date of birth, and first, also modifies the getBooks() function to return a string corresponding to the date of birth, but in a human-readable format. Finally, it modifies the setter functions so that they return nothing, as they will not be used.

Lastly, open src/Entity/Book.php and update its content to match the following:

```
<?php
namespace App\Entity;

use App\Repository\BookRepository;
use Doctrine\ORM\Mapping as ORM;

#[ORM\Entity(repositoryClass: BookRepository::class)]
class Book
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column(type: "integer")]
    private int $id;

    #[ORM\Column(type: "string", length: 255)]
    private string $title;

    #[ORM\ManyToMany(targets: Author::class, inversedBy: "books")]
    #[ORM\OrderBy(value: "ASC")]
    private Collection $author;

    public function __construct()
    {
        $this->id = null;
        $this->title = null;
        $this->author = null;
    }

    #[ORM\Column(type: "string", length: 4)]
    private string $isbnformat;

    #[ORM\Column(type: "string")]
    private string $isbn;

    #[ORM\Column(type: "boolean")]
    private bool $outofprinting;

    #[ORM\Column(type: "text")]
    private ?int $copiessold;

    #[ORM\ManyToMany(targets: Author::class)]
    private Collection $coauthors;

    public function __construct(
        int $id,
        string $title,
        Collection $author,
        string $isbnformat,
        string $isbn,
        bool $outofprinting,
        int $copiessold,
        Collection $coauthors
    ) {
        $this->id = $id;
        $this->title = $title;
        $this->author = $author;
        $this->isbnformat = $isbnformat;
        $this->isbn = $isbn;
        $this->outofprinting = $outofprinting;
        $this->copiessold = $copiessold;
        $this->coauthors = $coauthors;
    }

    public function getId(): int
    {
        return $this->id;
    }

    public function setId(int $id): void
    {
        $this->id = $id;
    }

    public function getTitle(): string
    {
        return $this->title;
    }

    public function setTitle(string $title): void
    {
        $this->title = $title;
    }

    public function getAuthor(): Author
    {
        return $this->author;
    }

    public function setAuthor(Author $author): void
    {
        $this->author = $author;
    }

    public function getIsbnformat(): string
    {
        return $this->isbnformat;
    }

    public function setIsbnformat(string $isbnformat): void
    {
        $this->isbnformat = $isbnformat;
    }

    public function getIsbn(): string
    {
        return $this->isbn;
    }

    public function setIsbn(string $isbn): void
    {
        $this->isbn = $isbn;
    }

    public function isOutofprinting(): bool
    {
        return $this->outofprinting;
    }

    public function setOutofprinting(bool $outofprinting): void
    {
        $this->outofprinting = $outofprinting;
    }

    public function getCopiessold(): ?int
    {
        return $this->copiessold;
    }

    public function setCopiessold(?int $copiessold): void
    {
        $this->copiessold = $copiessold;
    }

    public function getCoauthors(): Collection
    {
        return $this->coauthors;
    }

    public function setCoauthors(Collection $coauthors): void
    {
        $this->coauthors = $coauthors;
    }
}
```

The modifications to the Book entity are similar to the ones made for the Author entity. Types for the fields have been added, and the constructor function has been updated to accept the required parameters. The getAuthor() function returns type `Collection`. In addition, a magic function for the Book entity has also added to support updating multiple fields in one request.

Next, update the database schema by running the following command:

```
symfony console doctrine:schema:update --force
```

Create Fixtures

To give an interesting data when we're developing the API, let's create some fixtures which will load data into the database. Start by creating a fixture to load authors by running the following command:

```
symfony console make:fixtures AuthorFixtures
```

Then, open the newly created src/DataFixtures/AuthorFixtures.php file and update its content to match the following:

```
<?php
namespace App\DataFixtures;

use App\Entity\Author;
use Doctrine\Bundle\FixtureBundle\Fixture;
use Doctrine\Persistence\ObjectManager;
use Faker\Factory;

class AuthorFixtures extends Fixture
{
    /* Create a new Faker instance */
    private $faker = Faker\Factory::create();

    public const NUMBER = 'NUMBER_NUMBER';

    private $numbers = 10;

    public function __construct()
    {
        $this->faker = Faker\Factory::create();
    }

    public function load(ObjectManager $manager): void
    {
        for ($i = 1; $i <= $this->numbers; $i++) {
            $author = new Author();
            $author->setId($i);
            $author->setFirstname($this->faker->firstname());
            $author->setLastname($this->faker->lastname());
            $author->setDob($this->faker->dateTime());
            $author->setEmail($this->faker->email);
            $author->setCountry($this->faker->country);

            $manager->persist($author);
            $manager->flush();
        }
    }

    public function getDependencies(): array
    {
        return array(
            \App\DataFixtures\BookFixtures::class,
            \App\DataFixtures\CoauthorFixtures::class,
        );
    }
}
```

The fixture adds 10 authors to the database using details generated by Faker. Faker saves the changes to the database the `persist()` function is called, passing in a the `$author` content and an author. This provides access to the Author in the fixture for loading books.

Create a new fixture for the Book entity using the following command:

```
symfony console make:fixtures BookFixtures
```

Open the newly created src/DataFixtures/BookFixtures.php file and update its content to match the following:

```
<?php
namespace App\DataFixtures;

use App\Entity\Book;
use App\Entity\Author;
use Doctrine\Bundle\FixtureBundle\Fixture;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Persistence\ObjectManager;
use Faker\Factory;

class BookFixtures extends Fixture
{
    /* Create a new Faker instance */
    private $faker = Faker\Factory::create();

    public const NUMBER = 'NUMBER_NUMBER';

    private $numbers = 10;

    public function __construct()
    {
        $this->faker = Faker\Factory::create();
    }

    public function load(ObjectManager $manager): void
    {
        for ($i = 1; $i <= $this->numbers; $i++) {
            $book = new Book();
            $book->setId($i);
            $book->setTitle($this->faker->sentence(3));
            $book->setAuthor(new ArrayCollection([$this->getReference('author_'.$i)]));
            $book->setIsbnformat('isbn-13');
            $book->setIsbn($this->faker->isbn13);
            $book->setOutofprinting(false);
            $book->setCopiessold(null);
            $book->setCoauthors(null);

            $manager->persist($book);
            $manager->flush();
        }
    }

    public function getDependencies(): array
    {
        return array(
            \App\DataFixtures\AuthorFixtures::class,
            \App\DataFixtures\CoauthorFixtures::class,
        );
    }
}
```

```
<?php
namespace App\Models;
use App\Models\Author;
use App\Models\Book;
use App\Models\Publisher;
use App\Models\AuthorBookPublisher;
use App\Models\AuthorBookPublisherPivot;
use App\Models\Genre;
use App\Models\GenreCollection;
use App\Models\GenreCollectionPivot;
use App\Models\Review;
use App\Models\ReviewCollection;
use App\Models\ReviewCollectionPivot;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Illuminate\Database\Eloquent\Relations\HasOneThrough;
use Illuminate\Database\Eloquent\Relations\HasManyThrough;
use Illuminate\Database\Eloquent\Relations\BelongsToMany;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class BookDetailsPivot
{
    private $id;
    private $book_id;
    private $publication_date;
    private $publication_place;
    private $publication_company;
    private $publication_price;
    private $publication_currency;
    private $publication_isbn;
    private $publication_pages;
    private $publication_language;
    private $publication_edition;
    private $publication_publication_id;
    private $publication_publication_name;
    private $publication_publication_address;
    private $publication_publication_phone;
    private $publication_publication_email;
    private $publication_publication_website;
    private $publication_publication_contact_person;
    private $publication_publication_contact_phone;
    private $publication_publication_contact_email;
    private $publication_publication_contact_website;
    private $publication_publication_contact_address;
    private $publication_publication_contact_notes;
    private $publication_publication_contact_created_at;
    private $publication_publication_contact_updated_at;
    private $publication_publication_contact_deleted_at;
    private $publication_publication_contact_deleted_by;
    private $publication_publication_contact_deleted_reason;
    private $publication_publication_contact_deleted_reason_other;
    private $publication_publication_contact_deleted_reason_other_text;
    private $publication_publication_contact_deleted_reason_other_text_id;
    private $publication_publication_contact_deleted_reason_other_text_name;
    private $publication_publication_contact_deleted_reason_other_text_created_at;
    private $publication_publication_contact_deleted_reason_other_text_updated_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_by;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_id;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_name;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_created_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_updated_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_by;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_id;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_name;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_created_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_updated_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_by;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_id;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_name;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_created_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_updated_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_by;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_id;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_name;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_created_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_updated_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_by;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_id;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_name;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_created_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_updated_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_by;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_id;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_name;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_created_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_updated_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_by;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_id;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_name;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_created_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_updated_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_by;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_id;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_name;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_created_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_updated_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_by;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_id;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_name;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_created_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_updated_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_at;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_by;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text;
    private $publication_publication_contact_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text_deleted_reason_other_text
```

Create an author

Here, we will start with mutations by using the following query to create a new author:

```
mutation createAuthor {
  createAuthor(author: {name: "Amazia Author", dateOfBirth: "01/11/1990", bio: "Great Author"}){
    id
    name
    dateOfBirth
    bio
  }
}
```

That's how to develop a GraphQL-Powered API with Symfony

In this article, you built a GraphQL-powered API using Symfony. Using GraphQL, the client is able to specify the structure of the response (based on a previously agreed schema) thereby dealing with the issue of over or under-fetching data. Additionally, we are able to leverage the type checking provided out of the box to prevent unwanted bugs.

Tweet
Share
Share