

Symfony Framework

Santosh Kalwar, 30.04.2024

Symfony Topics

General Discussions

Object-relational mapping (ORM)

What is doctrine?

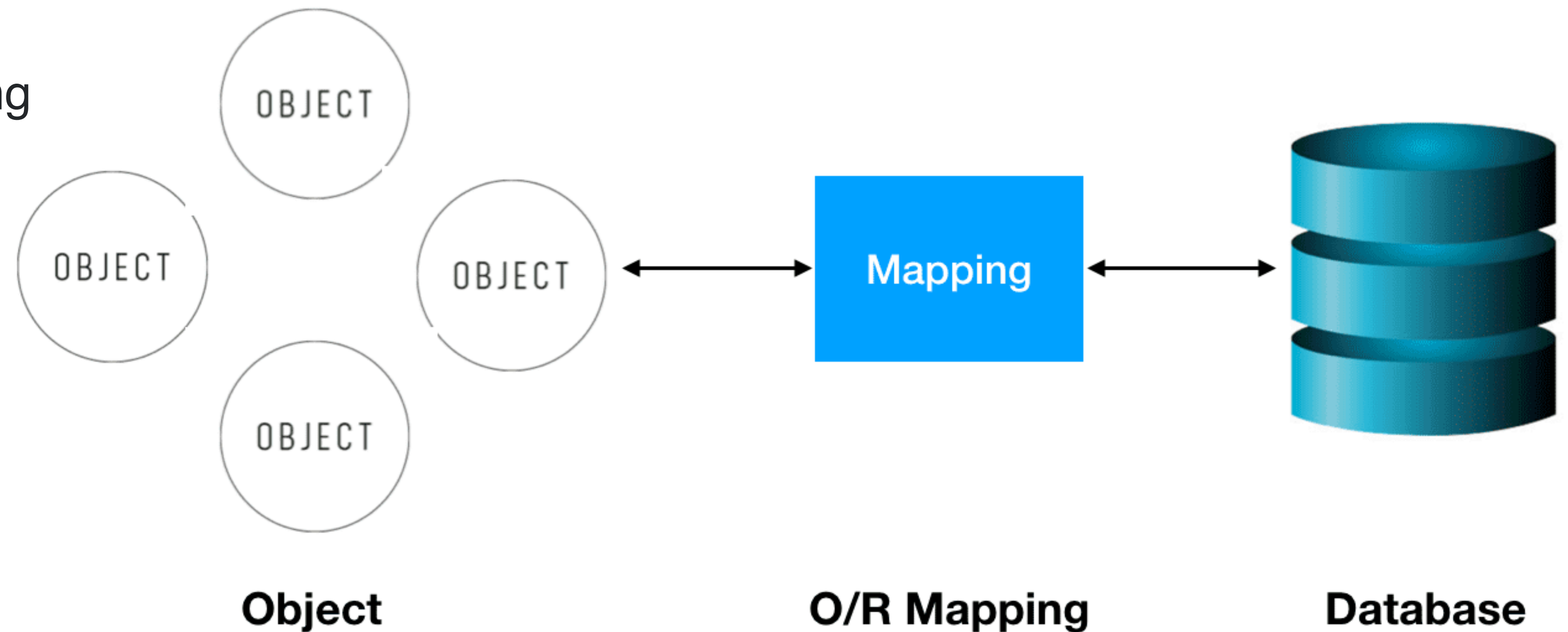
Why use doctrine?

Some Symfony DB terms

CRUD Task List App - Working Together / Learning DB Concepts

Object-relational mapping

Object-Relational Mapping (ORM) is a technique that lets you query and manipulate data from a database using an object-oriented paradigm. When talking about ORM, most people are referring to a *library* that implements the Object-Relational Mapping technique, hence the phrase "an ORM".



https://en.wikipedia.org/wiki/Object-relational_mapping

<https://stackoverflow.com/questions/1279613/what-is-an-orm-how-does-it-work-and-how-should-i-use-one>

Symfony Database Concepts

ORM - Object Relational Mapping

We write code relating to the collection of related objects, without having to worry about the way the data in those objects is actually represented and stored via a database.

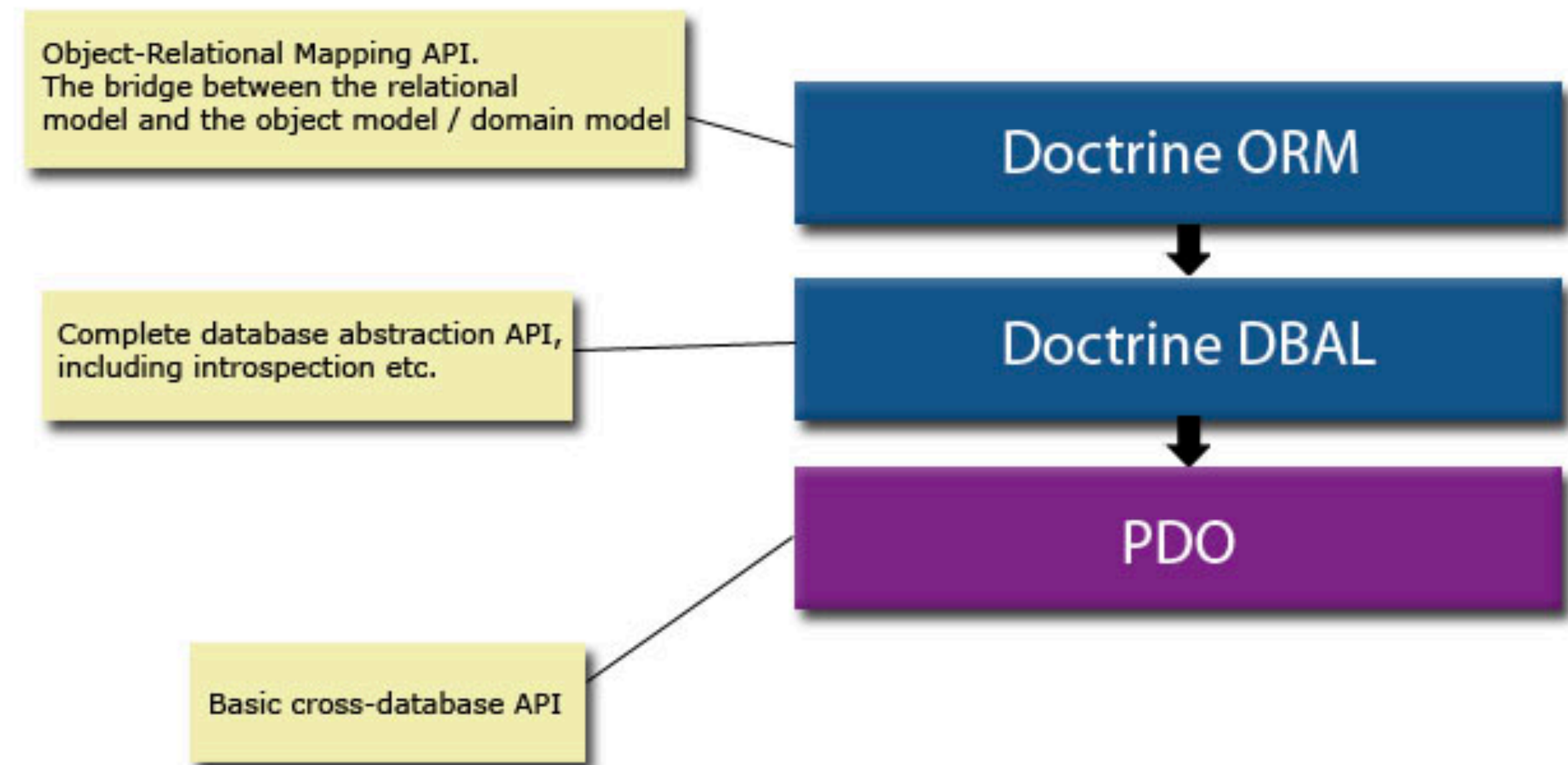
https://en.wikipedia.org/wiki/Object-relational_mapping

Symfony uses the Doctrine ORM system, and that offers DQL, the Doctrine Query Language.

<https://www.doctrine-project.org/>

[https://en.wikipedia.org/wiki/Doctrine_\(PHP\)](https://en.wikipedia.org/wiki/Doctrine_(PHP))

<https://github.com/doctrine>



What is doctrine?

Doctrine is the best set of PHP libraries to work with databases.

Doctrine supports relational databases like MySQL and PostgreSQL and also NoSQL databases like MongoDB.

Why use Doctrine DBAL and PDO in doctrine?

We use Doctrine DBAL so that we do not have to re-write SQL queries for other databases.

PDO is a "data-access layer", not a "database abstraction layer". This means that you can switch databases and still make the same method calls, but PDO will not re-write SQL queries to match the selected database or emulate any database functionality.

<https://symfony.com/doc/current/doctrine.html>

<https://www.doctrine-project.org/>

<https://stackoverflow.com/questions/9244300/for-doctrine-orm-why-is-dbal-needed-in-addition-to-pdo>

Why use doctrine?

- Around since 2006 with a very stable, high-quality codebase.
 - Extremely flexible and powerful object-mapping and query features.
 - Support for both high-level and low-level database programming for all your use-cases.
 - Large Community and integrations with many different frameworks
 - ([Symfony](#), [Laravel](#) and more).
-
- *Trivia*: Let us say you have a million posts in your DB and you want to display this in the newsfeed section of your website, how would you do that?

<https://www.doctrine-project.org/>

<https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/introduction.html>

Configuring the database

The database connection information is stored as an environment variable called `DATABASE_URL`. For development, you can find and customize this inside `.env`:

Run `php bin/console list doctrine` to see a full list.

Symfony DB terms/concepts

Entity: a type of object that is used to hold data.

EntityManager API is used to create and remove the persistent entity

The repository is a doctrine repository for retrieving a collection of objects. This repository has access to main objects e.g. EntityManager and QueryBuilder.

Migration: Migration helps to automatically create a new table in DB. In general, they are like a type of version control for your database

flush: The flush operation tells the Doctrine ORM to "now apply the changes".

CRUD Task App

In this app, we will learn the following:

- Create a record in a database
- Read a record from a database
- Update a record
- Delete a record



CREATE



READ



UPDATE



DELETE

C

R

U

D

Symfony commands and steps

1/2

In this CRUD app, following commands must be used to create CRUD App

```
php bin/console make:controller CRUDListController
```

```
composer require symfony/maker-bundle - - dev
```

```
php bin/console make:controller CRUDListController
```

```
composer require symfony/twig-bundle
```

```
composer require doctrine
```

Inside docker, type following commands

```
# pwd (present working directory)
```

```
/var/www
```

```
# cd web (go to the “web” folder inside docker)
```

```
# ls -la
```

Symfony commands and steps

2/2

In this CRUD app, following commands must be used to create CRUD App

Inside docker container, under “web” folder terminal run:

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

```
docker-compose up --build (if database name is not shown in phpmyAdmin)
```

After database is created, you can create a model or entity in Symfony (**outside docker**)

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

Entity should be e.g. class name with data types

After entity is created, you will need to do the migration from **inside docker terminal**

```
php bin/console make:migration
```

```
php bin/console doctrine:migration:migrate
```

Symfony-MAMP

IMPORTANT NOTE:

Every time you want to create a new Symfony application, you can **rename** “web” folder to something else and then create new application using following command:

```
symfony new web
```

Reason for this is docker looks for “web” folder inside Symfony-MAMP

If you don't intend to use **databases, mysql or phpmyadmin** then Symfony-MAMP is not needed you can create normal symfony application with

```
symfony new react24k_symfony_app
```

<https://github.com/kalwar/Symfony-MAMP>