

Repositories

One repository manager for the entities of Doctrine and the data sources of derafu/repository

Anonymous · 26/10/2026

Repositories of Derafu over Doctrine

`DoctrineRepositoryFactory` creates a repository of `derafu/repository` for a class that Doctrine maps. Registered in the repository manager under those classes, one manager gives the repository of every class, whether its data is in a table or in a data source:

```
use Derafu\Orm\Factory\DoctrineRepositoryFactory;
use Derafu\Repository\Service\DataProvider;
use Derafu\Repository\Service\RepositoryManager;

$manager = new RepositoryManager(
    dataProvider: new DataProvider([App\Entity\Country::class => __DIR__ .
    '/countries.php']),
    factories: [App\Entity\Invoice::class => new
    DoctrineRepositoryFactory($entityManager)],
);

$invoices = $manager->getRepository(App\Entity\Invoice::class); // Doctrine.
$countries = $manager->getRepository(App\Entity\Country::class); // Data source.
```

Both are a `RepositoryInterface`, with the same queries (see [querying](#)). An entity in the database can refer to one that is not by its identifier, in a plain column, and get it through the manager.

The factory takes any `EntityManagerInterface`, not only the one of `EntityManagerFactory`. It rejects an identifier that is not a class that Doctrine maps with a `ManagerException`, and each class has to be registered, as the manager does not look for the classes that Doctrine maps by itself.

`DoctrineRepository` gives the queries of `RepositoryInterface` over a Doctrine repository. The lock arguments of `find()` reach Doctrine, which requires a transaction for a pessimistic lock, as when it is used directly. The methods that are only of Doctrine's repository, such as `createQueryBuilder()`, are available through the entity manager.

Last updated on 26/10/2026