

ADRAR\o/ **PÔLE NUMÉRIQUE**

- > **INFRASTRUCTURES** SYSTÈMES & RÉSEAUX
- > **CYBERSÉCURITÉ** INFRASTRUCTURES & APPLICATIONS
- > **DEVOPS / SCRIPTING** & AUTOMATISATION
- > **DEVELOPPEMENT** WEB & MOBILE
- > **TRANSFORMATION NUMERIQUE DES ENTREPRISES**

www.adrar-numerique.com

Les Repositories Symfony

Vous les avez déjà vus sans même vous en apercevoir sûrement.
Les *repositories* sont des fichiers en lien avec vos entités.

Ce sont ces mêmes fichiers qui vont récupérer les informations depuis la BDD.

Les Repositories Symfony

Par défaut et dans tout *repository* généré, on retrouve 4 méthodes héritées et donc accessibles:

- `find`: retourne un utilisateur unique grâce à son ID, sinon null
- `findOneBy`: retourne un utilisateur unique en fonction d'un tableau de critères, sinon null
- `findAll`: retourne l'ensemble des utilisateurs
- `findBy`: retourne un ensemble d'utilisateurs en fonction d'un tableau de critères

En plus de ça, à chaque création, vous retrouvez en bas de la classe deux exemple qui utilisent le **QueryBuilder**

Syntaxes disponibles (1/4)

Symfony

En plus de ces méthodes héritées, nous avons la possibilité (et heureusement) de créer des requêtes plus personnalisées.

Il existe 4 syntaxes disponibles:

- DQL (Doctrine Query Language)
- QueryBuilder
- SQL (Structured Query Language)
- NSQL (Native Structured Query Language) (que nous ne verrons pas dans ce cours)

Nous allons prendre en exemple dans les slides suivantes la récupération de l'ensemble des utilisateurs avec les différentes syntaxes.

Sous DQL

```
/**                                     UserRepository.php
 * @return User[]
 */
public function findAllUsers(/* string $email */): array
{
    $entityManager = $this->getEntityManager();

    $query = $entityManager->createQuery(
        'SELECT u
        FROM App\Entity\User u
        -- WHERE u.email=:email
        ORDER BY u.email ASC'
    )
    /*->setParameter('email', $email)*;

    // Retourne un tableau d'objets "User"
    return $query->getResult();
}
```

Sous QueryBuilder

```
/**                                                                 UserRepository.php
 * @return User[]
 */
public function findAllUsers(/* string $email */): array
{
    // Retourne un tableau d'objets "User"
    return $this->createQueryBuilder('u')
        //     ->where('u.email = :email')
        //     ->setParameter('email', $email)
        //     ->orderBy('u.id', 'ASC')
        //     ->andWhere('u.active = TRUE')
        ->setMaxResults(10)
        ->getQuery()
        ->getResult();
}
```

Sous SQL

```
/**
 * @return User[]
 */
public function findAllUsers(/* string $email */): array
{
    $conn = $this->getEntityManager()->getConnection();

    $sql = '
        SELECT *
        FROM user u
        -- WHERE u.email=:email
        ORDER BY u.email ASC
    ';











    $resultSet = $conn->executeQuery($sql /*, ['email' => $email]*/);

    // returns un tableau de tableau SANS objet
    return $resultSet->fetchAllAssociative();
}
```

UserRepository.php

Exemple (1/4) Symfony

Prenons en exemple la table des utilisateurs ci-après:

  sf_adrar_classrooms user
 id : int
 email : varchar(180)
 roles : json
 password : varchar(255)
 nom : varchar(50)
 prenom : varchar(50)
 image : varchar(255)
 google_id : varchar(255)

Exemple (2/4)

Symfony

Dans mon contrôleur, je viens faire appel à mon *Repository* pour récupérer mes informations depuis ma BDD

```
#[Route(path: '/users', name: 'app_users')]
public function users(UserRepository $userRepository): Response
{
    $users = $userRepository->findAllUsers(); // Une des méthodes vues précédemment

    return $this->render('users/users.html.twig', ['users' => $users]);
}
```

UserController.php

Exemple (3/4)

Symfony

Je retrouve ici ma feuille Twig associée à mon contrôleur:

```
{% extends "base.html.twig" %}

{% block css %}
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
{% endblock %}

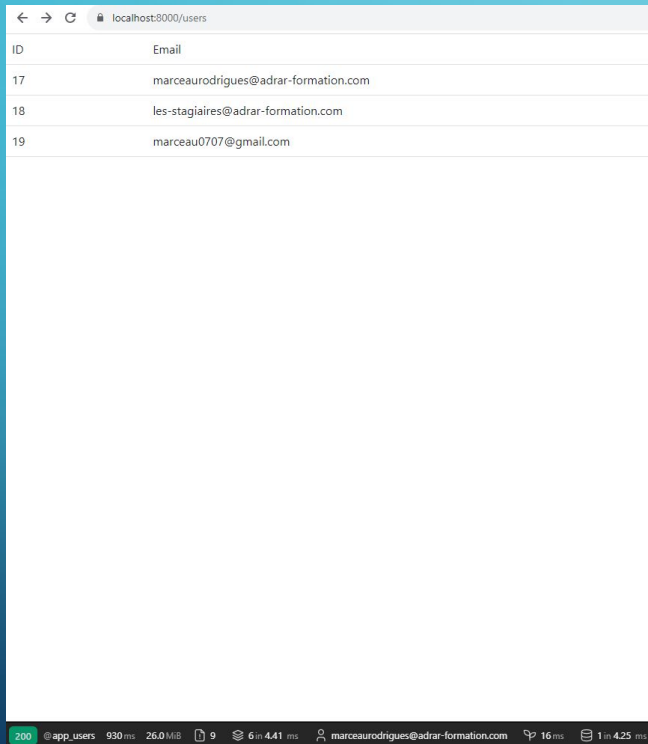
{% block js %}
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js"></script>
{% endblock %}

{% block body %}
<table class="table">
  <thead>
    <tr>
      <th>ID</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    {% for user in users %}
      <tr>
        <td>{{ user.id }}</td>
        <td>{{ user.email }}</td>
      </tr>
    {% endfor %}
  </tbody>
</table>
{% endblock %}
```

users/users.html.twig

Exemple (4/4) Symfony

Peu importe le type de requête utilisée, la page nous retournera toujours le résultat suivant:



ID	Email
17	marceaurodrigues@adrar-formation.com
18	les-stagiaires@adrar-formation.com
19	marceau0707@gmail.com