SYMFONY
INTRO

PRÉSENTATION

- Créé par Sensio Labs et Fabien Potencier
- Sortie en 2006
- Un des framework PHP historique
- Le plus populaire en France
  http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/

INSTALLATION

Outils indispensables

- Git (https://git-scm.com/downloads)
- Composer (https://getcomposer.org/download)
- Installeur symfony (https://symfony.com/download)

symfony.phar

```
symfony list
```

Default (-bash) ⌥⌘1

```
Usage:
  symfony [global options] <command> [command options] [arguments...]

Global options:
  --help, -h                        Show help
  --no-ansi                         Disable ANSI output
  --ansi                            Force ANSI output
  --no-interaction                  Disable all interactions
  --quiet, -q                       Do not output any message
  -v|vv|vvv, --verbose, --log-level Increase the verbosity of messages: 1 for normal output, 2 and 3 for more verbose outputs and 4 for debug
  -V                                Print the version

Available commands:
 account
  account:info                                                        Show info about your Symfony account
  account:ips                                                         List SymfonyCloud IPs for use in scripts
  account:login, login                                                Log in with your SymfonyConnect account
  account:logout, logout                                              Logout from your SymfonyConnect account
  account:ssh:key:add                                                 Add an SSH key
  account:ssh:key:remove                                              Remove an SSH key
  account:ssh:keys                                                    List project's SSH keys
 book
  book:check-requirements, book:check                                 Check that you have all the pre-requisites locally to code while
reading the "Symfony 5: The Fast Track" book
  book:checkout                                                       Check out a step of the "Symfony 5: The Fast Track" book reposito
ry
 domain
  domain:attach                                                       Attach a domain
  domain:cname, cname                                                 Display the CNAME value to use in DNS entries
  domain:default                                                      Display/Set the default domain
  domain:detach                                                       Detach a domain
  domain:get                                                          Show detailed information for a domain
  domain:list, domains                                                List domains
  domain:update                                                       Update a domain
 env
  env:activate                                                        Activate one or several environments
```

Installation

```
symfony new [nom de dossier]
```

Default (-bash) ⌥⌘1

```
* Creating a new Symfony project with Composer
  (running /usr/local/bin/composer.phar create-project symfony/skeleton /Users/seb/documents/formation/symfony/exos/forma
tion  --no-interaction)

* Setting up the project under Git version control
  (running git init /Users/seb/documents/formation/symfony/exos/formation)

 [OK] Your project is now ready in /Users/seb/documents/formation/symfony/exos/formation

mo-ghra-leictreonach:exos seb$
```

Lancement du serveur

Dans le dossier
d'un projet Symfony

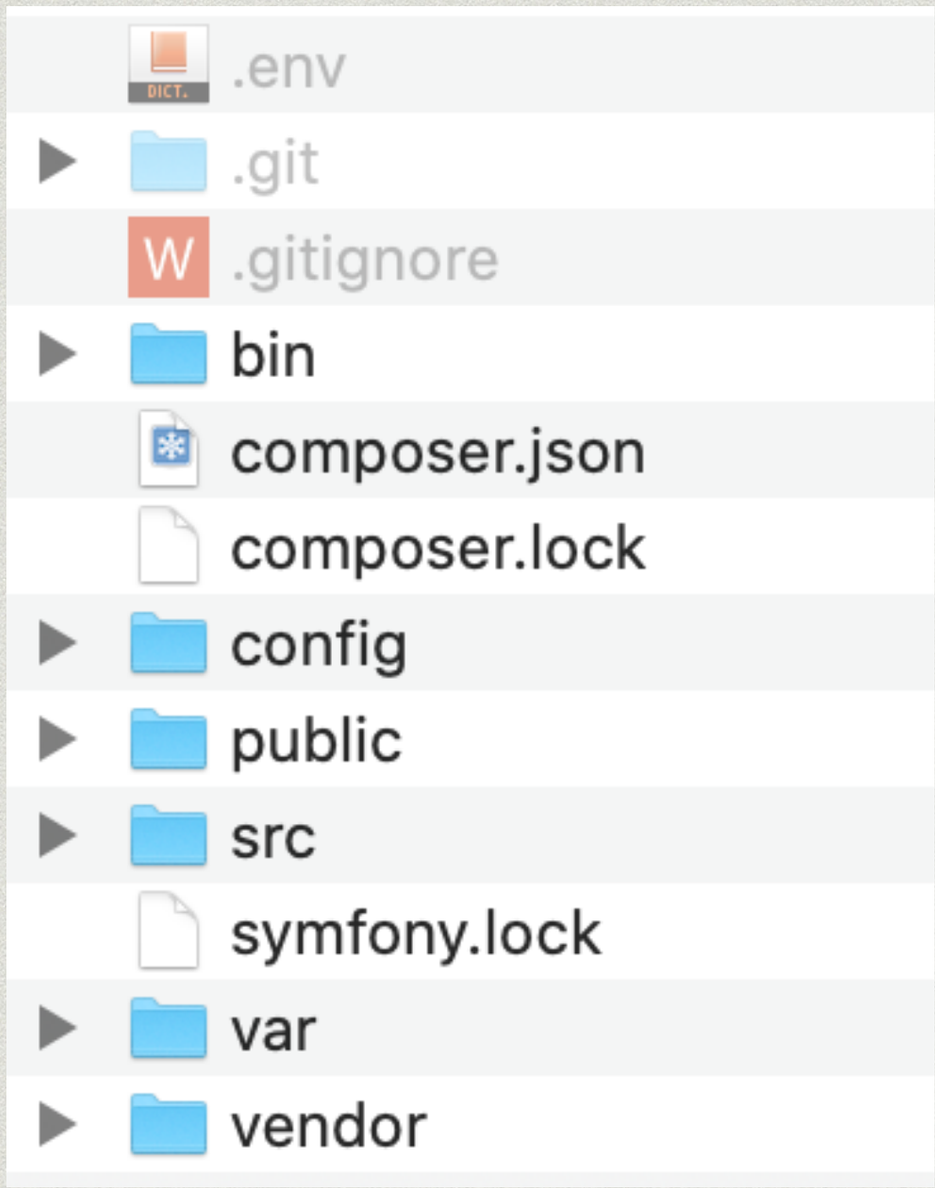
```
symfony server:start
```

ou

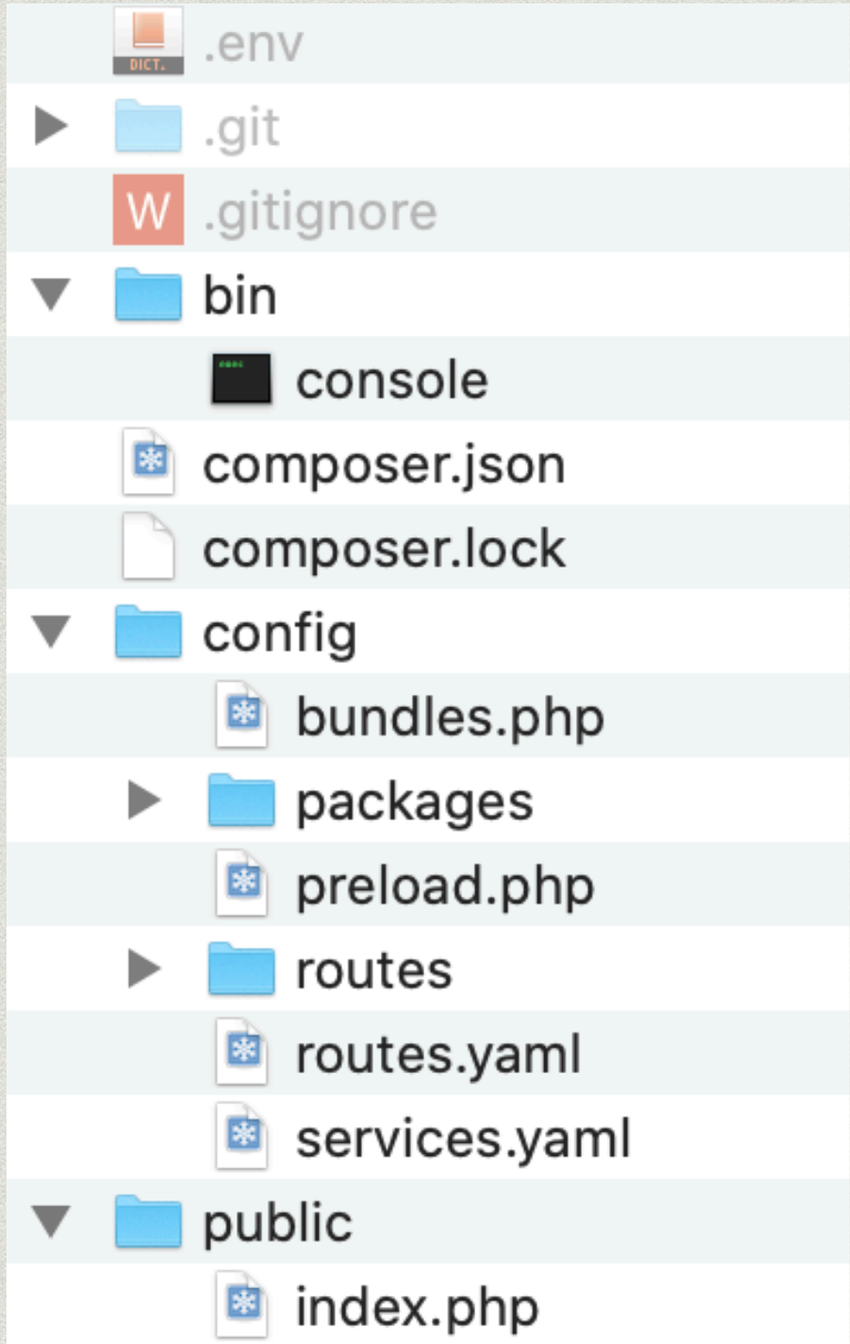
```
symfony serve -d
```

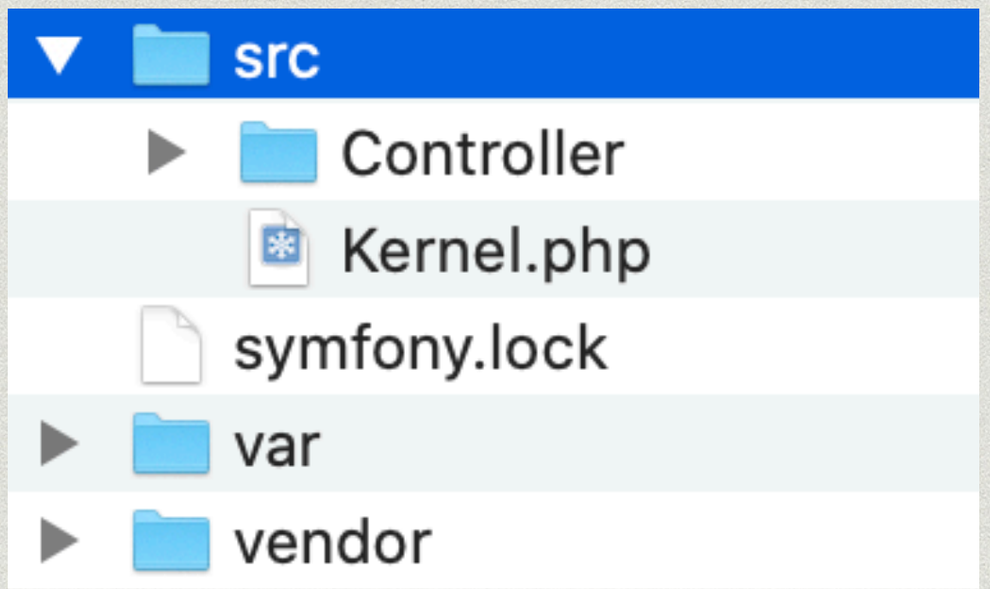
PRINCIPES

Arborescence

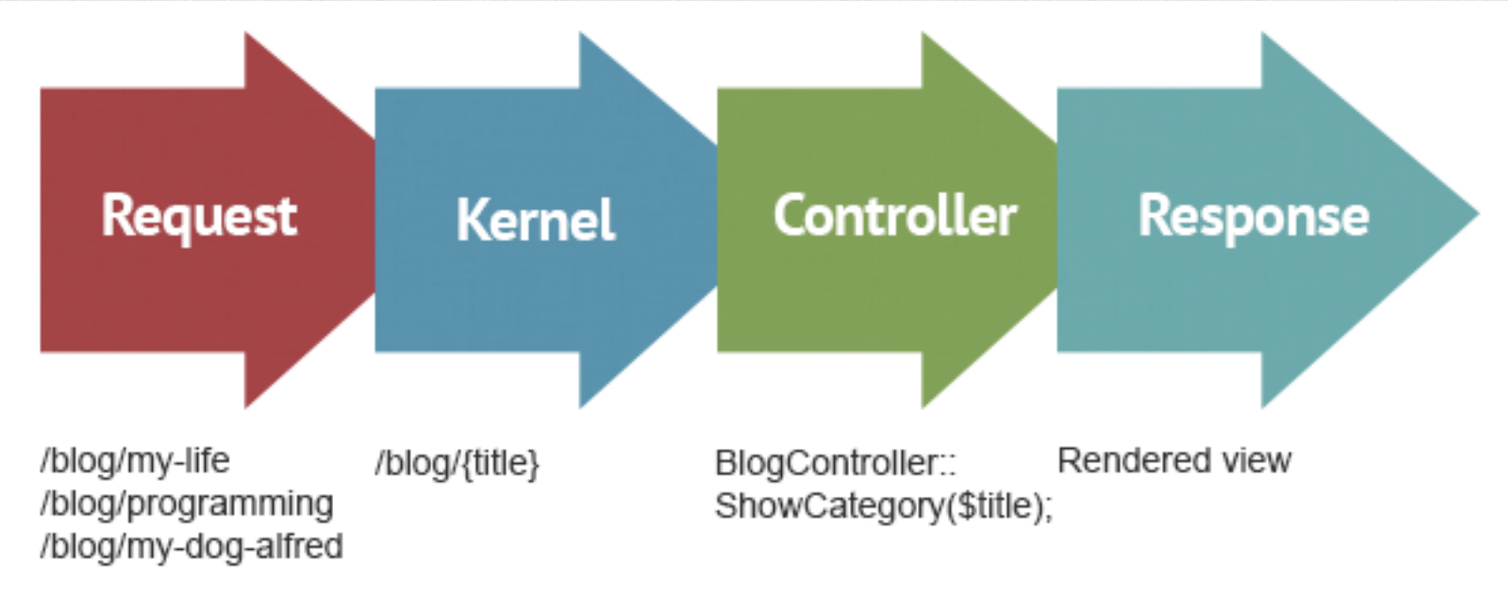


Sous-dossiers

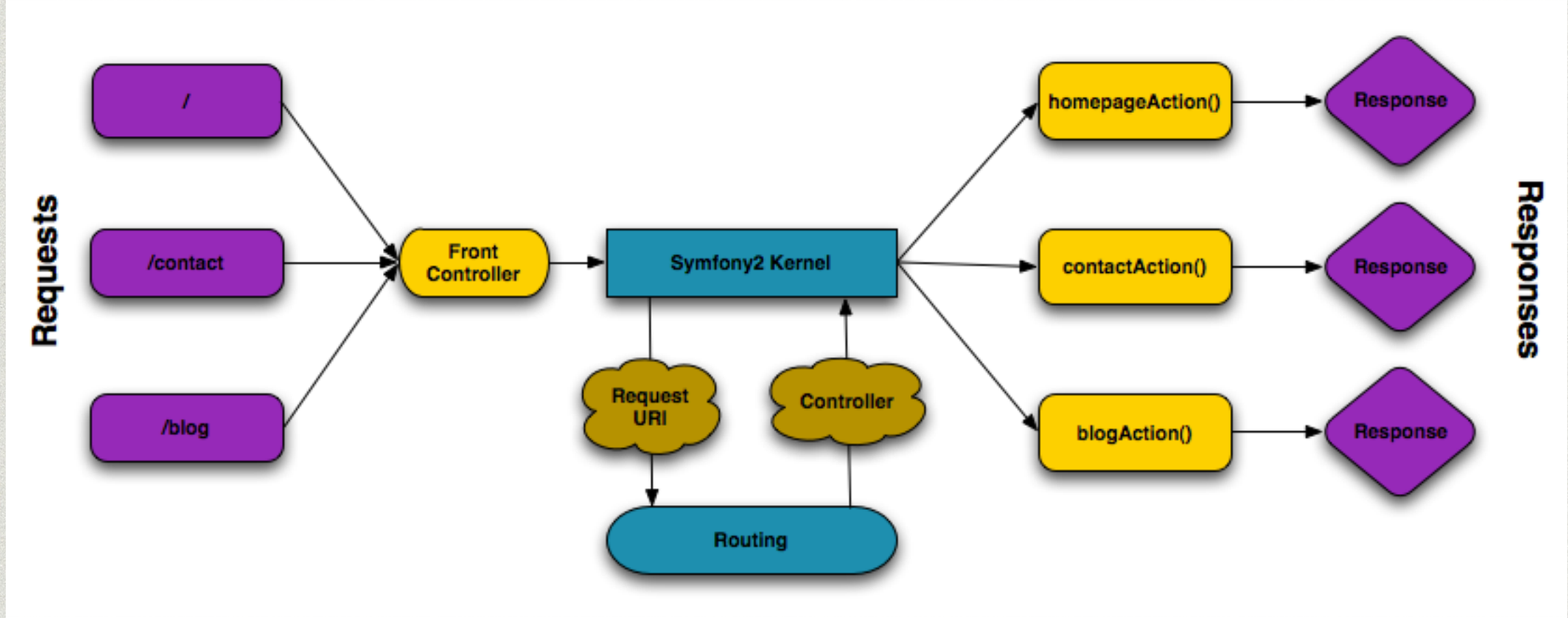




Framework http



Traitement de la requête

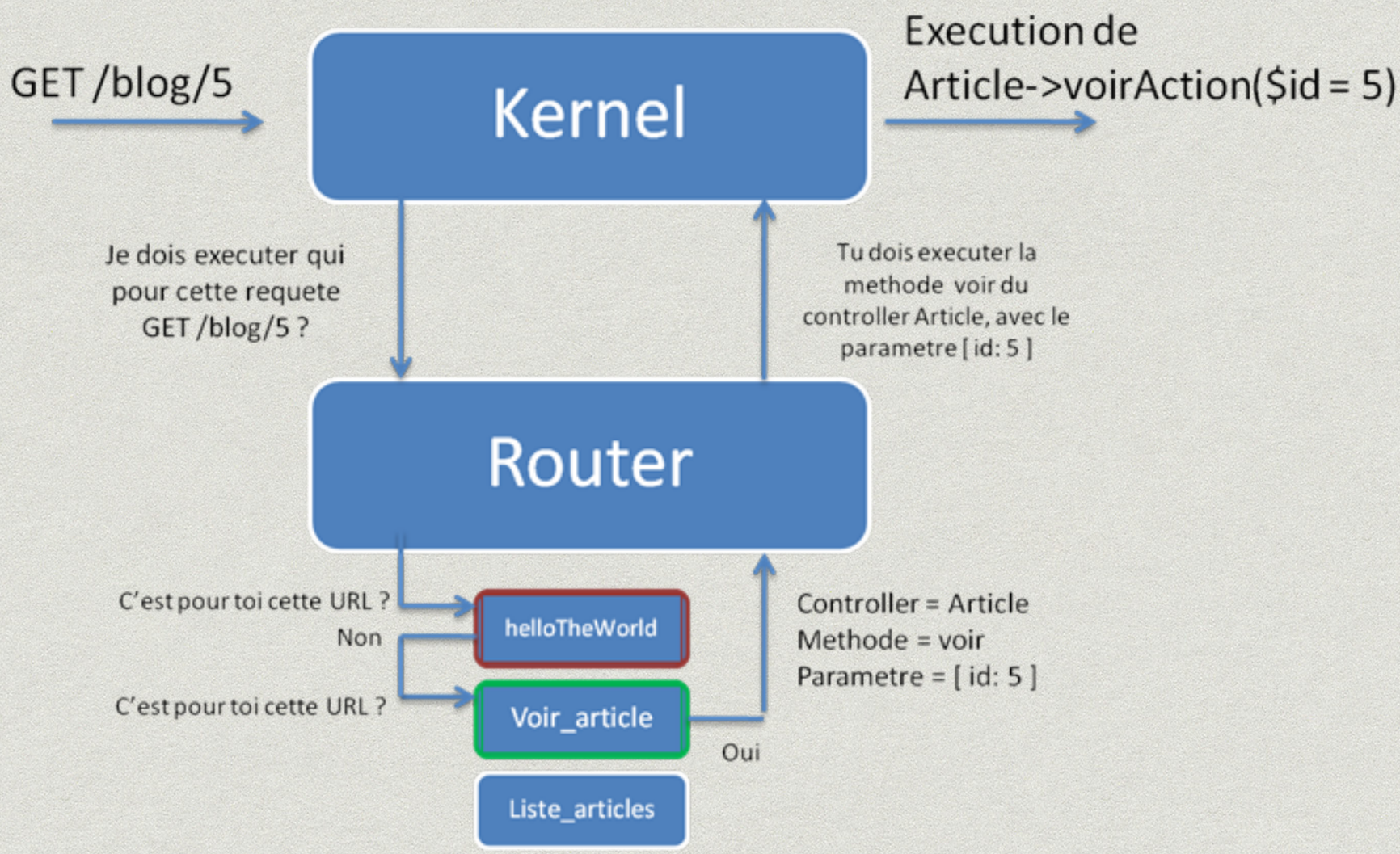


public/index.php

```php
<?php

use App\Kernel;
use Symfony\Component\Dotenv\Dotenv;
use Symfony\Component\ErrorHandler\Debug;
use Symfony\Component\HttpFoundation\Request;

require dirname(__DIR__).'/vendor/autoload.php';

(new Dotenv())->bootEnv(dirname(__DIR__).'/.env');

if ($_SERVER['APP_DEBUG']) {
    umask(0000);

    Debug::enable();
}

$kernel = new Kernel($_SERVER['APP_ENV'], (bool) $_SERVER['APP_DEBUG']);
$request = Request::createFromGlobals();
$response = $kernel->handle($request);
$response->send();
$kernel->terminate($request, $response);
```

ROUTES

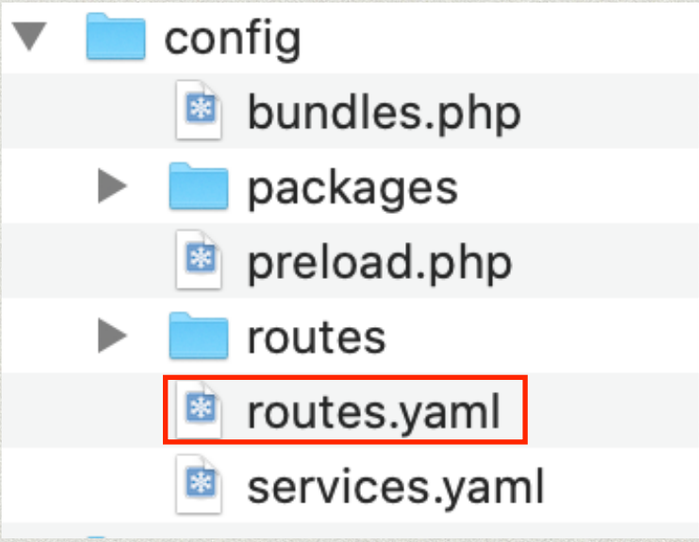
Routage


GET /blog/5
Kernel
Execution de
Article->voirAction($id = 5)
Je dois executer qui
pour cette requete
GET /blog/5 ?
Tu dois executer la
methode voir du
controller Article, avec le
parametre [id: 5]
Router
C'est pour toi cette URL ?
Non
helloTheWorld
C'est pour toi cette URL ?
Voir_article
Oui
Liste_articles
Controller = Article
Methode = voir
Parametre = [id: 5]

Création d'un contrôleur

```php
<?php
// src/Controller/LuckyController.php
namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;

class HomeController
{
    public function hello(): Response
    }
        return new Response(
            '<h1>Hello</h1>'
        );
    }
}
```

Définition de la route




```
hello:
     path: /hello
     controller:
App\Controller\HomeController::hello
```

Paramètres

controller

```php
<?php
namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;

class HomeController
{
    /**
     * @param $name
     * @return Response
     */
    public function hello($name){
        return new Response("<h1>Hello $name</h1>");
    }
}
```

route

```yaml
hello:
    path: /hello/{name}
    controller: App\Controller\HomeController::hello
```

Annotations

Définition des routes dans le contrôleur

```
composer require annotations
```

```php
<?php
namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class HomeController
{
    /**
     * @Route("/hello/{name}")
     * @param $name
     * @return Response
     */
    public function hello($name){
        return new Response("<h1>Hello $name</h1>");
    }
}
```

Requirements

Un expression régulière pour valider un paramètre

```
/**
 * @Route("/list/{page}", requirements={"page":"\d+"})
 * @param int $page
 * @return Response
 */
public function list(int $page): Response{
    return new Response("<h1>page : $page</h1>");
}
```

Nommer un route

Attribue un nom à la route
pour pouvoir y faire référence par la suite

```
/**
 * @Route("/hello/{name}", name="hello")
 * @param $name
 * @param Request $request
 * @return Response
 */
public function hello(string $name): Response{
    return new Response("<h1>Hello $name </h1>");
}
```

Valeur par défaut

```
/**
 * @Route("/hello/{name}")
 * @param $name
 * @return Response
 */
public function hello(string $name = "world"): Response{
    return new Response("<h1>Hello $name </h1>");
}
```

```
/**
 * @Route("/hello/{name}",
 *        defaults={"name"="World"})
 * @param $name
 * @return Response
 */
public function hello(string $name): Response{
    return new Response("<h1>Hello $name </h1>");
}
```

Route globale

```php
<?php
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

/**
 * @Route("/contact")
 * Class ContactController
 * @package App\Controller
 */
class ContactController extends AbstractController
{

    /**
     * @Route("/")
     * @return Response
     */
    public function index(): Response
    {
        return new Response("<h1>liste des contact </h1>");
    }

    /**
     * @Route("/show/{id}")
     * @return Response
     */
    public function index($id): Response
    {
        return new Response("<h1>contact id : $id</h1>");
    }
}
```

AbstractController

Une classe abstraite
qui donne des outils pour les contrôleurs

```
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class ContactController extends AbstractController
{

    /**
     * @Route("/contact")
     * @return Response
     */
    public function index(): Response
    {
        return new Response("<h1>Liste</h1>");
    }
}
```

Analyse des routes

```
php bin/console debug:router
```

Default (-bash) ⌥⌘1

```
------------------- -------- -------- ------ --------------------------
 Name                Method   Scheme   Host   Path
------------------- -------- -------- ------ --------------------------
 _preview_error      ANY      ANY      ANY    /_error/{code}.{_format}
 app_contact_index   ANY      ANY      ANY    /contact
 app_home_hello      ANY      ANY      ANY    /hello/{name}
 app_home_list       ANY      ANY      ANY    /list/{page}
 hello               ANY      ANY      ANY    /hello/{name}
------------------- -------- -------- ------ --------------------------

mo-ghra-leictreonach:formation seb$
```

Analyse des routes

```
php bin/console debug:router --show-controllers
```

Default (-bash) ⌥⌘1

```
------------------- -------- -------- ------ -------------------------- ------------------------------------------
 Name                Method   Scheme   Host   Path                       Controller
------------------- -------- -------- ------ -------------------------- ------------------------------------------
 _preview_error      ANY      ANY      ANY    /_error/{code}.{_format}   error_controller::preview()
 app_contact_index   ANY      ANY      ANY    /contact                   App\Controller\ContactController::index()
 app_home_hello      ANY      ANY      ANY    /hello/{name}              App\Controller\HomeController::hello()
 app_home_list       ANY      ANY      ANY    /list/{page}               App\Controller\HomeController::list()
 hello               ANY      ANY      ANY    /hello/{name}              App\Controller\HomeController::hello()
------------------- -------- -------- ------ -------------------------- ------------------------------------------

mo-ghra-leictreonach:formation seb$
```

A vous

- Créer un contrôleur CalcController
- Créer des routes pour les 4 opérations (addition, soustraction, multiplication, division) admettant deux arguments n1 et n2

REQUEST / RESPONSE

Symfony Demo application

localhost:8000/fr

Bienvenue sur l'application **Symfony Demo**

Parcourir la **section publique** de l'application de démonstration.

Parcourir l'**interface d'administration** de l'application de démonstration.

2.7.6 | php | e3e970 | 200 | TemplateController :: templateAction | 0 | 542 ms | 6.5 MB | 0 | 5 | 24 ms | anon. | 0

Inspecteur | Console | Débogueur | Éditeur de s... | Éditeur de s... | Canvas | Performances | Réseau | Stockage | Web Audio

Méthode	Fichier	Domaine	Type
200 GET	fr	localhost:8000	html
200 GET	app.css	localhost:8000	css
200 GET	app.js	localhost:8000	js
200 GET	e3e970	localhost:8000	html
200 GET	Lato-Regular.woff	localhost:8000	octe...
200 GET	Lato-Bold.woff	localhost:8000	octe...
200 GET	fontawesome-webfont.woff2?v=4.3.0	localhost:8000	octe...
200 GET	Lato-Regular.woff	localhost:8000	octe...
200 GET	Lato-Bold.woff	localhost:8000	octe...

En-têtes | Cookies | Paramètres | Réponse | Délais | Aperçu

URL de la requête : http://localhost:8000/fr
Méthode de la requête : GET
Adresse distante : 127.0.0.1:8000
Code d'état : 200 OK
Version : HTTP/1.1

Modifier et renvoyer | En-têtes bruts

Filtrer les en-têtes

En-têtes de la réponse (0,243 Ko)

Cache-Control : "no-cache"
Connection : "close"
Content-Type : "text/html; charset=UTF-8"
Date : "Sun, 31 Jan 2016 14:12:29 GMT"
Host : "localhost:8000"
X-Debug-Token : "e3e970"
X-Debug-Token-Link : "/_profiler/e3e970"
X-Powered-By : "PHP/5.6.7"

En-têtes de la requête (0,446 Ko)

Host : "localhost:8000"
User-Agent : "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:43.0) Gecko/20100101 Firefox/43.0"
Accept : "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8"
Accept-Language : "fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3"

Tout | HTML | CSS | JS | XHR | Polices | Images | Médias | Flash | Autre

9 requêtes - 728,25 Ko - 1,39 s | Effacer

Request

$request->query	$_GET
$request->request	$_POST
$request->cookies	$_COOKIE
$request->server	$_SERVER
$request->files	$_FILES
$request->headers	en têtes http
$request->attributes	info sur le routage

Récupération de l'objet

```
public function indexAction(Request $request){
  …
}
```

OU

```
public function indexAction(){
  $request = $this->get('request_stack')
              ->getCurrentRequest();
}
```

Collections

`$request->query`	$_GET
`$request->request`	$_POST
`$request->server`	$_SERVER
`$request->files`	$_FILES
`$request->getSession`	$_SESSION
`$request->cookies`	$_COOKIES

Méthodes des collections

`get(string $key)`	la valeur d'une clef
`all()`	toute la collection
`keys()`	la liste des clefs
`add(array $params)`	ajoute à une collection
`set(string $key, mixed $val)`	modifie la valeur d'une clef
`replace(array $params)`	remplace toute la collection
`has(string $key)`	test l'existence d'une clef
`remove(string $key)`	supprime une clef

Response

```
$response = new Response(
    '<h1>Hello</h1>',
    200,
    array(
        'content-type' => 'text/html'
    )
);

return $response;
```

Response

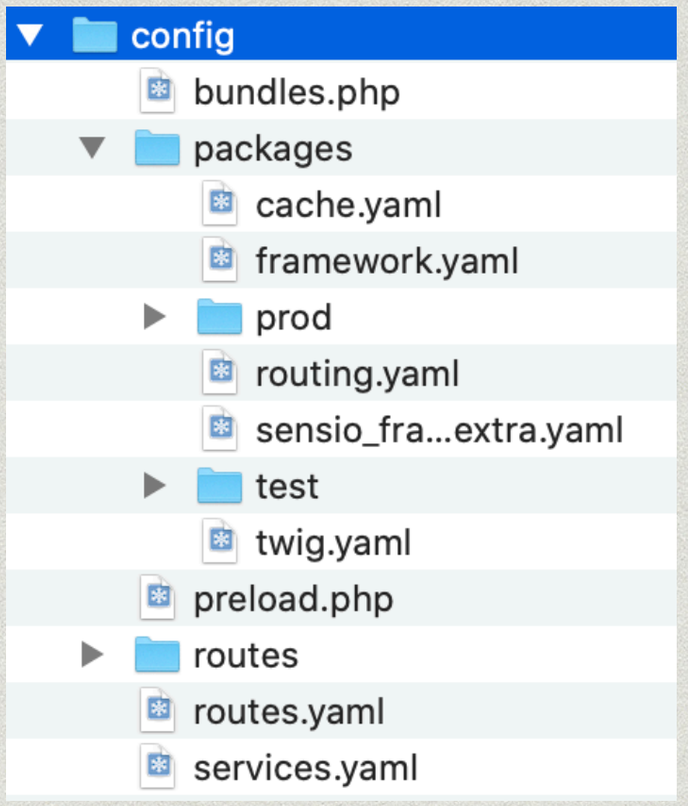
```
$response = new Response();
$response->headers->set('content-type', 'application/json');
$response->setStatusCode(200);

$response->setContent(json_encode($data));

return $response;
```

TWIG

Installation

```
composer require twig
```

Créer un nouveau dossier templates et un fichier de configuration



Contrôleur

```
/**
 * @Route("/show/{id}")
 * @return Response
 */
public function show($id): Response
{
    return $this->render("contact/show.html.twig", ["id" => $id]);
}
```

La vue

```
<h1>contact id : {{ id }}</h1>
```

Variables composites

```php
/**
 * @Route("/")
 * @return Response
 */
public function index(): Response
{
    return $this->render("contact/index.html.twig", [
        "fruits" => ["Pommes", "Poires", "Oranges"],
        "contact" => ["name" => "Seb", "age" => 50]
    ]);
}
```

```
{{fruits[0]}}

{{contact.name}}
```

Filtres

http://twig.sensiolabs.org/doc/filters/index.html

```
{{ name | upper }}
```

```
{{ name | upper | reverse }}
```

```
{{
  9800.333|
  number_format(2, ',', ' ')
}}
```

Tags

http://twig.sensiolabs.org/doc/tags/index.html

```
{% if age <18 %}
  mineur
{% else %}
  majeur
{% endif %}
```

```
{% filter upper %}
    hello {{name}}
{% endfilter %}
```

```
{% for element in liste %}
    <li>
       {{ element }}
    </li>
{% endfor %}
```

LOOP

loop.index	position actuelle du curseur de la boucle en commencant par 1
loop.index0	position actuelle du curseur de la boucle en commencant par 0
loop.revindex	nombre d’itérations restantes avant la fin de la boucle en finissant par 1
loop.revindex0	nombre d’itérations restantes avant la fin de la boucle en finissant par 0
loop.length	nombre d’éléments dans la séquence
loop.first	true si le curseur est sur le premier élément
loop.last	true si le curseur est sur le dernier élément

Boucle et variables

```
{% set total=0 %}

{% for produit in catalogue if produit.prix < 30 %}
   {% set total = (total + produit.prix) %}
   …
{% else %}
   Le catalogue produit est vide
{% endfor %}

{{ total | number_format(2, ',', ' ') }}
```

Fonctions

http://twig.sensiolabs.org/doc/functions/index.html

```
{% for item in itemList %}
<tr class="
 {{ cycle(['odd', 'even'], loop.index0) }}">
   …
</tr>
{% endfor %}
```

Tests

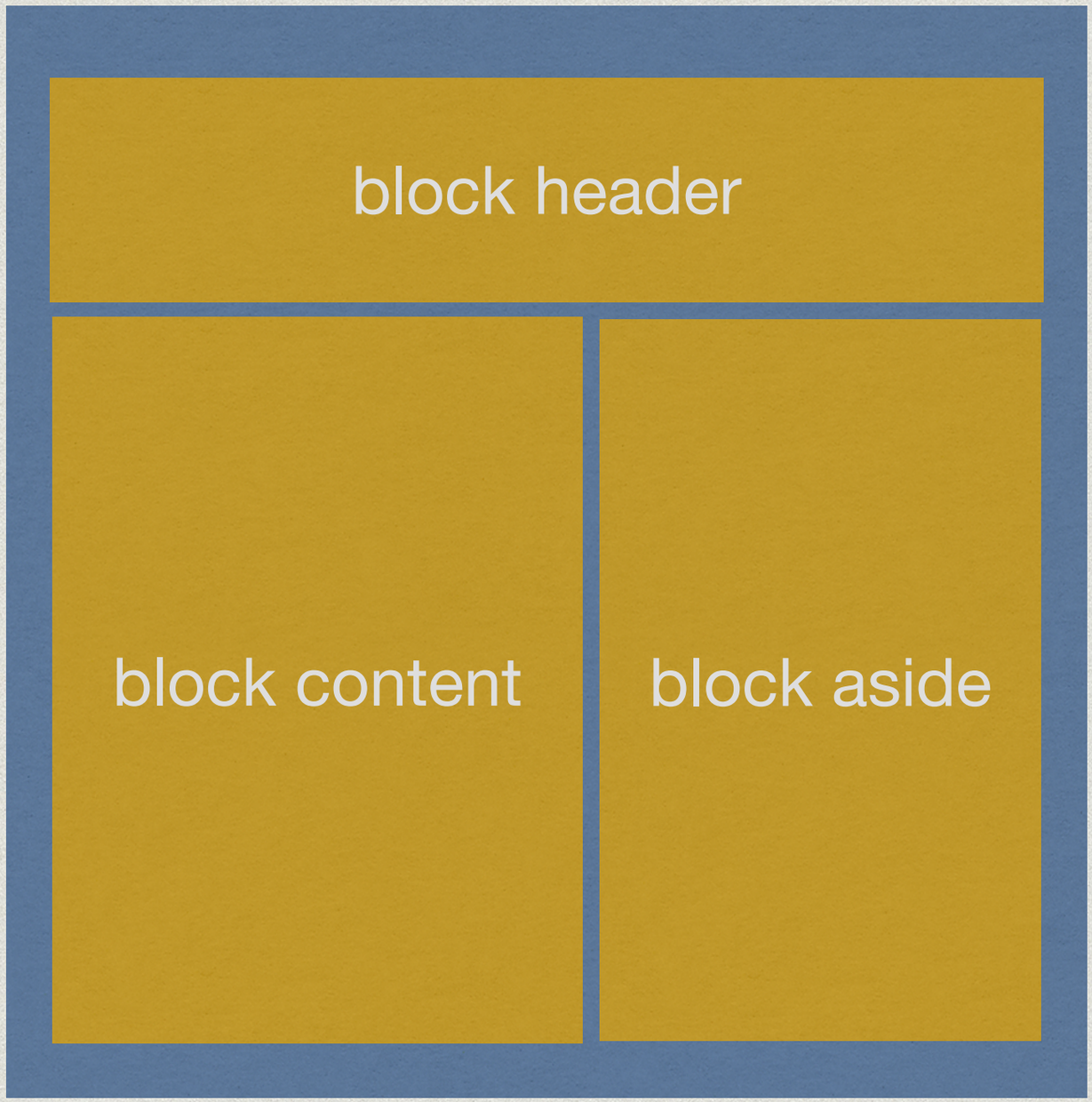
http://twig.sensiolabs.org/doc/tests/index.html

```
{% if defined itemList %}
  {% for item in itemList %}
    {% if loop.index is divisible by(3) %}
      ...
    {% endif %}
  {% endfor %}
{% endif %}
```

Liens

```
{# commentaire #}
{{ path('route', {'param':'valeur'}) }}
```

Héritage et blocks



Layout

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8" />
        <title>{% block title %}Welcome!{% endblock %}</title>
        {% block stylesheets %}{% endblock %}
    </head>
    <body>
        <header> {% block header %}
          <h1>Mon Blog</h1>
        {% endblock %} </header>
        {% block content %}{% endblock %}
        <aside> {% block aside %}{% endblock %} </aside>
        {% block javascripts %}{% endblock %}
    </body>
</html>
```

Template

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

{% block title %}MON BLOG{% endblock %}

{% block header %}
  {{ parent() }}
  <nav>…</nav>
{% endblock %}

{% block content %}…{% endblock %}
{% block aside %}…{% endblock %}
```

DOCTRINE

Installation

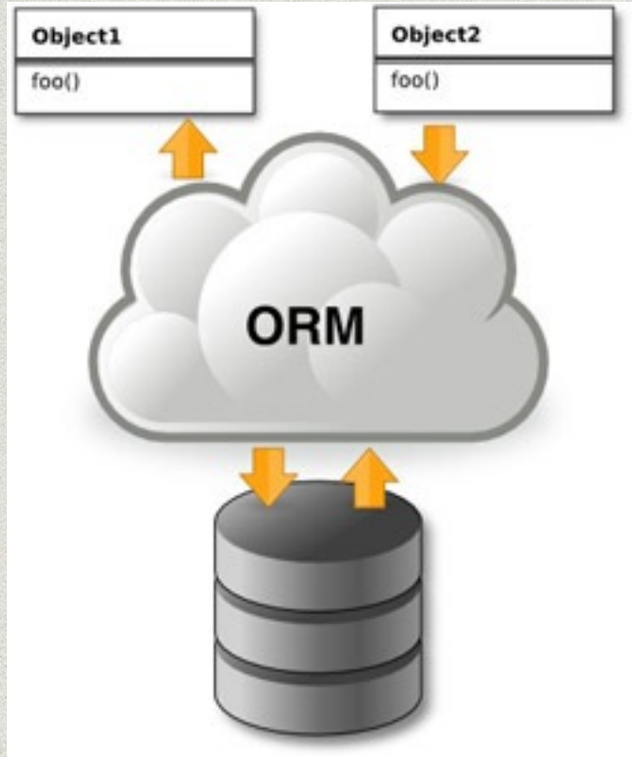
```
composer require doctrine
```

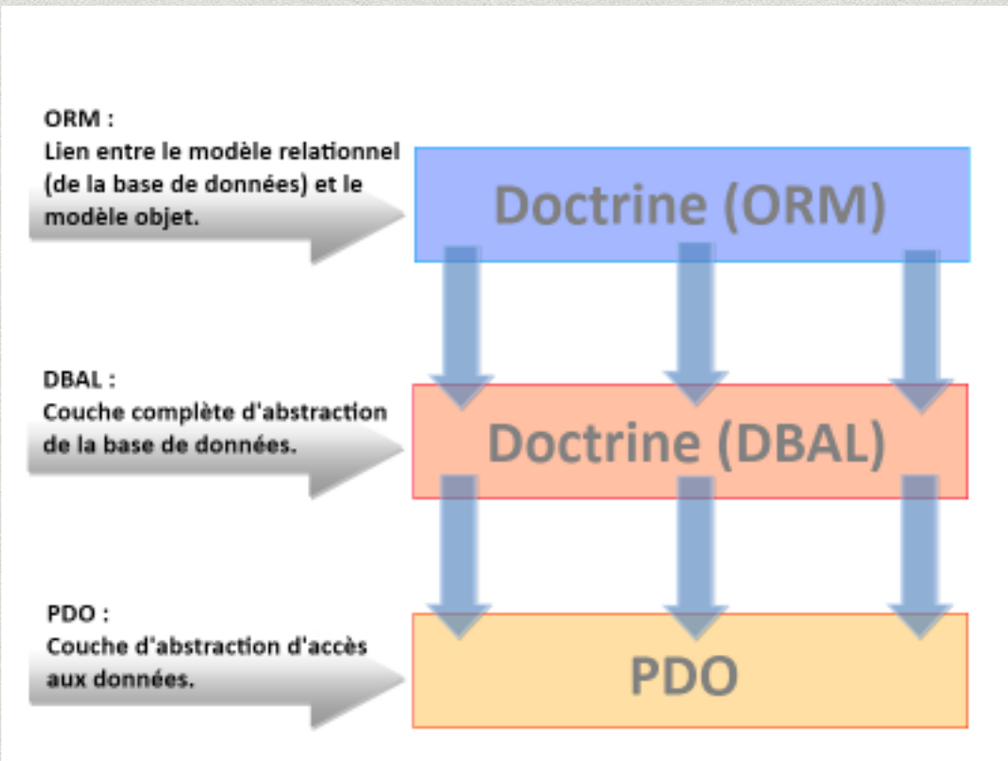
```
composer require --dev maker
```

Attention bug
Il faut utiliser la version 2.2 de doctrine sinon maker ne pourra pas créer les annotations pour les entités

```
"doctrine/doctrine-bundle": "2.2",
```

ORM


Object1
foo()
Object2
foo()
ORM

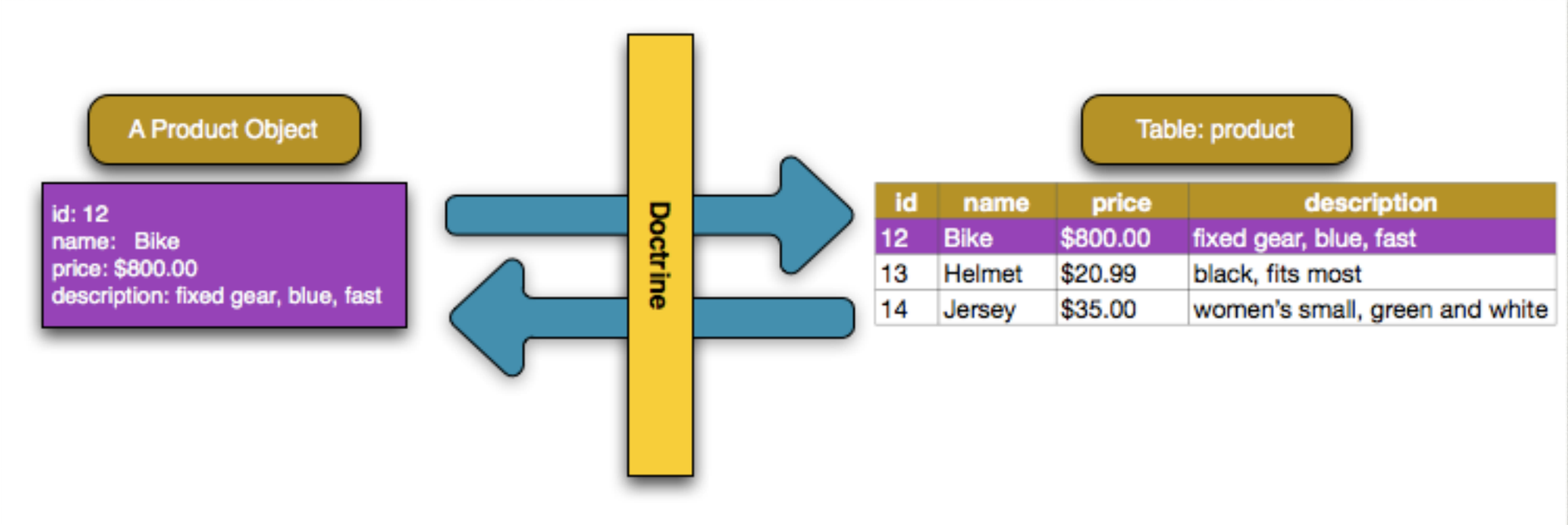

ORM :
Lien entre le modèle relationnel (de la base de données) et le modèle objet.
Doctrine (ORM)
DBAL :
Couche complète d'abstraction de la base de données.
Doctrine (DBAL)
PDO :
Couche d'abstraction d'accès aux données.
PDO

Configuration

fichier .env

```
# MySQL
DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/db_name?serverVersion=5.7

# MariaDB
DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/db_name?serverVersion=mariadb-10.5.8"
```

Entité



- Plain old PHP Object (POPO)
- Stockage des données

Créer une entité

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

Default (-bash) ⌥⌘1

```
Help:
  The make:entity command creates or updates an entity and repository class.

  php bin/console make:entity BlogPost

  If the argument is missing, the command will ask for the entity class name interactively.

  You can also mark this class as an API Platform resource. A hypermedia CRUD API will
  automatically be available for this entity class:

  php bin/console make:entity --api-resource

  You can also generate all the getter/setter/adder/remover methods
  for the properties of existing entities:

  php bin/console make:entity --regenerate

  You can also *overwrite* any existing methods:

  php bin/console make:entity --regenerate --overwrite

mo-ghra-leictreonach:formation seb$
```

Annotations

```
use Doctrine\ORM\Mapping as ORM;

/**
 *  @ORM\Table(name="blog_auteur")
 *  @ORM\Entity
 */
class Auteur {

     /**
      * @ORM\Column(name="id", type="integer")
      * @ORM\Id
      * @ORM\GeneratedValue(strategy="AUTO")
     */
    private $id;

/**
 * @ORM\Column(name="name",
 * type="string", length=50)
*/
    private $name;
}
```

Options de Colonnes

doc annotation @Columm

- nullable
- options (default, unsigned)
- unique
- length (string)
- scale et precision (float)

index

```
/**
 * Auteur
 *
 * @ORM\Table(name="auteurs",
 * indexes={
    @Index(name="idx_email", columns={"email"}),
    @Index(name="idx_nom", columns={"nom"}),
    @Index(name="idx_email_nom", columns={"email", "nom"})
 * }
 *
 * )
 * @ORM\Entity
 */
```

Migrations

Création des classes de migration

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

Génération des tables de la BD en fonction des entités

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

Persistence

```
//Gestionnaire d’entité
$em = $this->getDoctrine()->getManager();

//hydratation de l’entité
$auteur = new Auteur();
$auteur->setNom('Maloron');
$auteur->setPrenom('Sébastien');

$em->persist($auteur);

//Commit sur la transaction
$em->flush();
```

Repository

```
$repository = $this->getDoctrine()
                ->getRepository('AppBundle:Produits');

//Collection d’entités
$produits = $repository->findAll();

//Une seule entité
$pk = 5;
$produit = $repository->find($pk);
```

findBy

```
public function findBy(
        array $criteria,
        array $orderBy = null,
        $limit=null,
        $offset = null
    );
```

```
// Récupère les 5 derniers articles de l'auteur Seb
$articles = $repository->findBy(
    array ('auteur' => 'Seb'),
    array ('date_parution', 'desc'),
    5
);
```

Méthode magique

```
// Récupère les 3 premiers articles d'un auteur
$articles = $repository->findByAuteur(
                    'seb',
                    array('date_parution' => 'asc'),
                    3
                );
```

Correspondances

```
app/console doctrine:mapping:convert yml src/
AppBundle/Resources/config/ORM/Mapping
--from-database --force
```

Génération

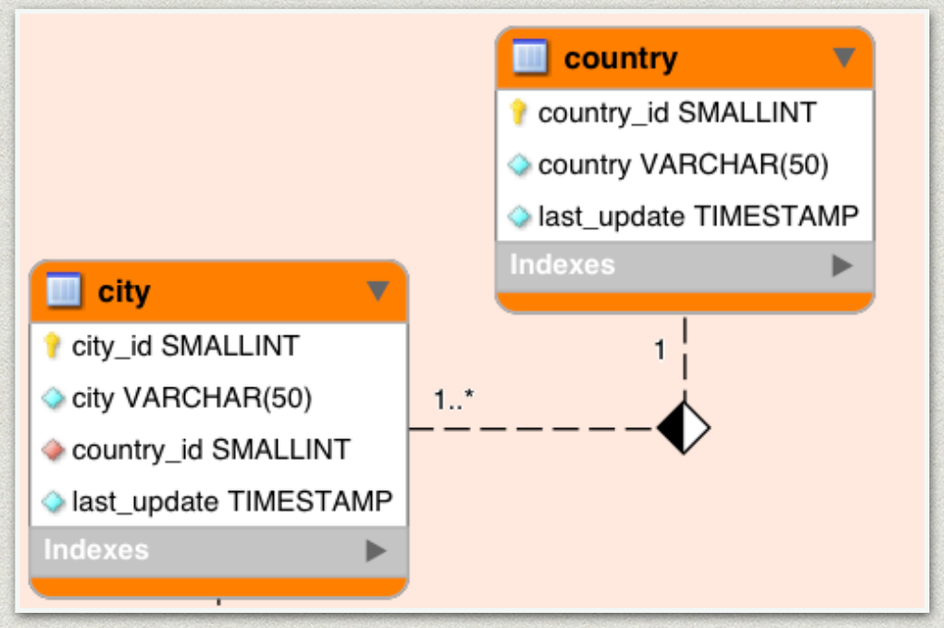
```
app/console doctrine:mapping:import
AppBundle annotation
```

ASSOCIATIONS

OneToOne

```
class Auteur
{
    /**
     * @var Image
     *
     * @ORM\OneToOne(targetEntity="Image")
     */
    private $photo;
}
```

ManyToOne




```
class City
{
    /**
     * @var Country
     *
     * @ORM\ManyToOne(targetEntity="Country")
     */
    private $country;
}
```

ManyToOne bidirectionnel

```
class City
{
    /**
     * @var Country
     *
     * @ORM\ManyToOne(
            targetEntity="Country",
            inversedBy="cities")
     */
    private $country;
}
```

entité

variable sur
l’entité inverse

OneToMany bidirectionnel

```
class Country
{
    /**
     * @var ArrayCollection
     *
     * @ORM\OneToMany(
            targetEntity="City",
            mappedBy="country")
     */
    private $cities;

  public function __construct(){
    $this->cities = new ArrayCollection();
  }
}
```

entité

variable sur l’entité inverse

Colonne de jointure

```
/**
* @var Visiteur
*
* @ORM\ManyToOne(
*   targetEntity="Visiteur",
*   inversedBy="commentaires")
* @ORM\JoinColumn(
*     name="auteur_commentaire_id",
*   referencedColumnName="id")
*/
private $visiteur;
```

clef étrangère

clef primaire

cascade

```
class Auteur
{
    /**
     * @var Image
     *
     * @ORM\OneToOne(targetEntity="Image",
         cascade={"persist", "remove"}
        )
     */
    private $photo;
}
```

ManyToMany

```
/**
 * @ORM\Table(name="articles")
 * @ORM\Entity
 */
class Article
{
    /**
     * @var ArrayCollection
     *
     * @ORM\ManyToMany(targetEntity="tag")
     * @ORM\JoinTable(name="posts_tags")
     */
    private $tags;
}
```

ManyToMany bidirectionnel

```
class Article
{
    /**
     * @var ArrayCollection
     * @ORM\ManyToMany(targetEntity="tag",
     *  inversedBy="articles")
     * @ORM\JoinTable(name="articles_tags")
     */
    private $tags;
}
```

```
class Tag
{
    /**
     * @var ArrayCollection
     *
     * @ORM\ManyToMany(targetEntity="Article",
     *  mappedBy="tags")
     */
    private $articles;
}
```

EVENTS

LifeCycleCallbacks

Triggers au niveau de Doctrine

```
/**
 * Auteur
 *
 * @ORM\Table(name="auteurs")
 * @ORM\Entity
 * @ORM\HasLifecycleCallbacks()
 */
class Auteur
{
    ...
}
```

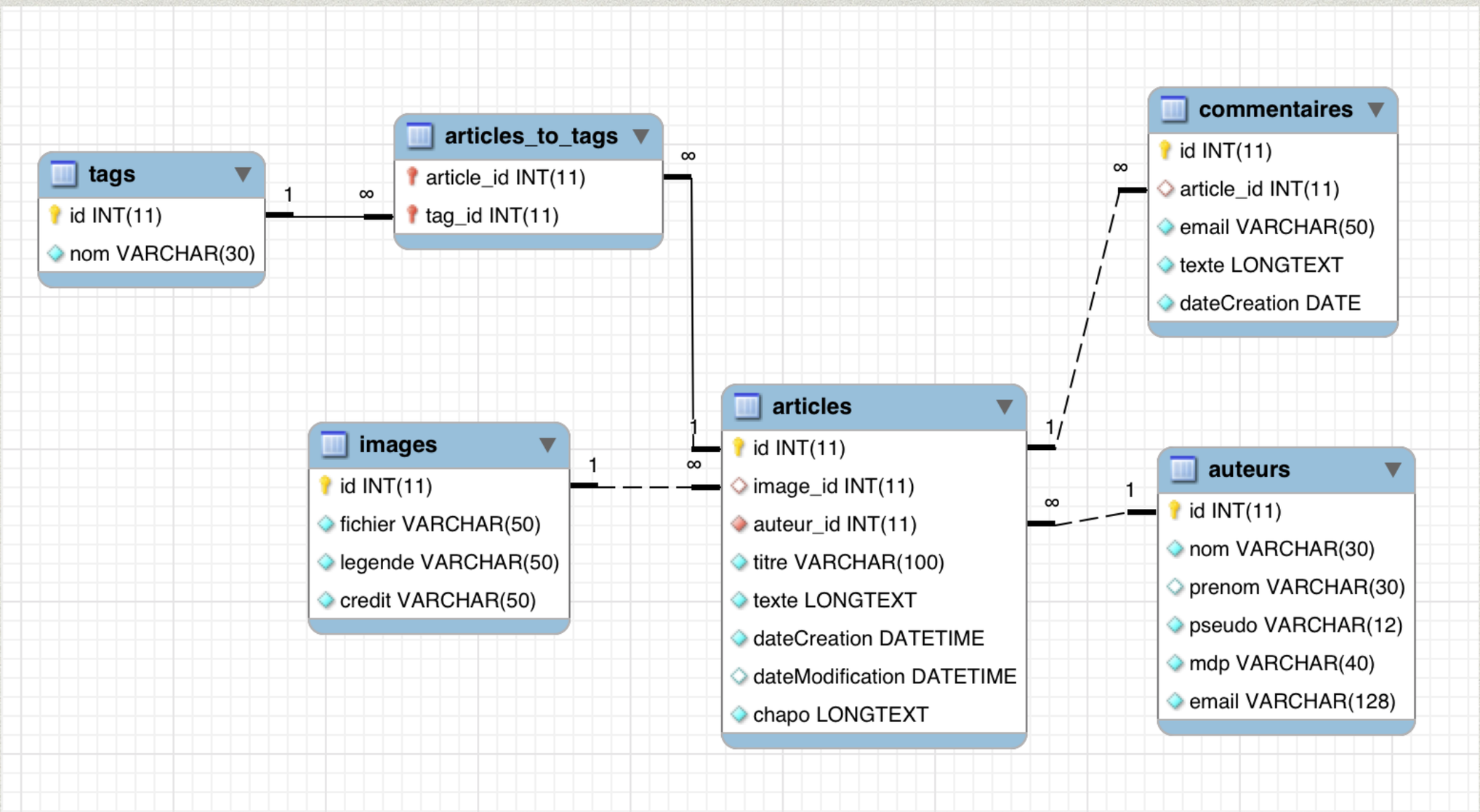
LifeCycleCallbacks

```
class Auteur
{
    /**
     * Set dateCreation
     * @ORM\PrePersist()
     */
    public function prePersistEvent()
    {
      $now = new \DateTime();
      $this->dateCreation = $now;
      $this->dateModification = $now;
    }

}
```

- PrePersist
- PostPersist
- PreUpdate
- PostUpdate
- PreRemove
- PostRemove
- PreFlush
- PostLoad

APPLICATION

Blog


tags
id INT(11)
nom VARCHAR(30)
articles_to_tags
article_id INT(11)
tag_id INT(11)
commentaires
id INT(11)
article_id INT(11)
email VARCHAR(50)
texte LONGTEXT
dateCreation DATE
images
id INT(11)
fichier VARCHAR(50)
legende VARCHAR(50)
credit VARCHAR(50)
articles
id INT(11)
image_id INT(11)
auteur_id INT(11)
titre VARCHAR(100)
texte LONGTEXT
dateCreation DATETIME
dateModification DATETIME
chapo LONGTEXT
auteurs
id INT(11)
nom VARCHAR(30)
prenom VARCHAR(30)
pseudo VARCHAR(12)
mdp VARCHAR(40)
email VARCHAR(128)
1
∞
1
∞
∞
1
1
∞
∞
1

Installation

dans composer.json

```
require: {
  "doctrine/doctrine-fixtures-bundle": "~2.2.0",
}
```

dans la console à la racine du projet

```
composer.phar update
```

Déclaration

dans AppKernel.php

```
$bundles = array(
  new Doctrine\Bundle\FixturesBundle\DoctrineFixturesBundle(),
);
```

Fixture

dans src/AppBundle/DataFixtures\ORM

```
class LoadAuteurs extends AbstractFixture {
    public function load(ObjectManager $manager)
    {
        // Premier auteur
        $auteur = new Auteur();
        $auteur->setNom('Martin');
        $manager->persist($auteur);

        // Persistence
        $manager->flush();
    }
}
```

Chargement des fixtures

```
app/console doctrine:fixtures:load
```

2. bash

```
mbpdesebastien:sf-news seb$ app/console doctrine:fixtures:load
```

Ordre de chargement

- Implémenter l’interface : OrderedFixtureInterface
- Définir une méthode : getOrder() qui retourne un entier

```
class LoadAuteur extends AbstractFixture
      implements OrderedFixtureInterface
{

    public function load(ObjectManager $manager)
    {
       …
    }

    /**
     * Ordre d'exécution
     * @return integer
     */
    public function getOrder()
    {
        return 1;
    }
}
```

Références

Utiliser une fixture dans une autre

J’ai chargé un auteur, je veux l’utiliser pour charger un article

méthodes setReference et getReference
de l’objet AbstractFixtures

```
class LoadAuteur extends AbstractFixture implements
OrderedFixtureInterface
{

    public function load(ObjectManager $manager)
    {
        $auteur = new Auteur();
        // Ici hydratation et persistence de l’objet

        //Enregistrement de la référence
        $this->addReference('auteur_toto', $auteur);
    }

    public function getOrder()
    {
        return 1;
    }
}
```

```
class loadArticle extends AbstractFixture implements
OrderedFixtureInterface
{

    public function load(ObjectManager $manager)
    {
        $article = new Article();
        //Récupération de la référence à l'auteur
        //définie dans le fixture précédente
        $article->setAuteur(
            $this->getReference('auteur_toto')
        );

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

    public function getOrder()
    {
        return 2;
    }
}
```

Principe

- Un objet constructeur de requête
- Étend les possibilités de la classe Repository

Création d'un QueryBuilder

```
// QueryBuilder depuis EntityManager
// (dans un contrôleur)
$em = $this->getDoctrine()->getEntityManager();
$qb = $em->createQueryBuilder();

// QueryBuilder depuis le repository
$repository = $em->getRepository('Bundle:Entity');
// L'argument de la méthode correspond
// à l'alias de l'entité dans le QueryBuilder
$qb = $repository->createQueryBuilder('e');
```

Méthode select

```
// en SQL : SELECT a.* FROM auteurs as a
$qb->select('a');

// en SQL : SELECT a.nom, a.prenom
// FROM auteurs as a
$qb->select(array('a.nom','a.prenom'));
```

Méthode from

```
// en SQL :
//SELECT a.* FROM auteurs as a
$qb->select('a')
    ->from(AppBundle:Auteur, 'a');
```

Méthode where

```
// en SQL :
// SELECT a.* FROM auteurs as a WHERE a.actif=1

$qb->select('a')
    ->from(AppBundle:Auteur, 'a')
    ->where('a.actif=1');
```

Plusieurs conditions

```
// en SQL : SELECT a.* FROM auteurs as a
// WHERE a.actif=1 AND a.langue='fr'
$qb->select('a')
   ->from(AppBundle:Auteur, 'a')
   ->where('a.actif=1')
   ->andWhere("a.langue='fr'");

// en SQL : SELECT a.* FROM auteurs as a
// WHERE a.actif=1 AND a.langue='fr'
$qb->select('a')
   ->from(AppBundle:Auteur, 'a')
   ->where("a.actif=1 AND a.langue='fr'");
```

Méthode orderBy

```
// en SQL : SELECT a.* FROM auteurs as a
// WHERE a.actif=1 AND a.langue='fr'
// ORDER BY a.nom DESC, a.prenom ASC
$qb->select('a')
   ->from(AppBundle:Auteur, 'a')
   ->where("a.actif=1 AND a.langue='fr'")
   ->orderBy('a.nom', 'DESC')
   ->addOrderBy('a.prenom', 'ASC');
```

Jointure

```
// en SQL : SELECT p.*, a.nom FROM posts as p
// INNER JOIN auteurs as a ON p.auteur_id = a.id
// WHERE a.id=5
// ORDER BY p.date_parution DESC
$qb->select(array('p','a.nom'))
    ->from('AppBundle:Post', 'p')
    ->innerJoin('p.auteur', 'a')
    ->where('a.id=5')
    ->orderBy('p.dateParution', 'DESC');
```

Utilise les règles de jointure de l'association entre les entités

Condition de jointure supplémentaire

```
// en SQL : SELECT p.*, a.nom FROM posts as p
// INNER JOIN auteurs as a
// ON p.auteur_id = a.id AND a.actif=1
// WHERE a.id=5 ORDER BY p.date_parution DESC

$qb->select(array('p','a.nom'))
   ->from('AppBundle:Post', 'p')
   ->innerJoin('p.auteur', 'a', 'WITH', 'a.actif=1')
   ->where('a.id=5')
   ->orderBy('p.dateParution', 'DESC');
```

Limites

```
// en SQL (MySQL):
// SELECT p.*, a.nom FROM posts as p
// INNER JOIN auteurs as a ON p.auteur_id = a.id
// WHERE a.id=5 ORDER BY p.date_parution DESC
// LIMIT 5 OFFSET 0;

$qb->select(array('p','a.nom'))
    ->from('AppBundle:Post', 'p')
    ->innerJoin('p.auteur', 'a')
    ->where('a.id=5')
    ->orderBy('p.dateParution', 'DESC')
    ->setFirstResult(0)
    ->setMaxResults(5);
```

Regroupements

```
// en SQL (MySQL):
// SELECT p.*, COUNT(c.id) AS nb_commentaires
// INNER JOIN commentaires as c ON p.id = c.post_id
// GROUP BY p.id
// HAVING nb_commentaires > 5;

$qb->select(array('p','COUNT(c.id) AS nb_commentaires'))
   ->from('AppBundle:Post', 'p')
   ->innerJoin('p.commentaires','c')
   ->groupBy('p.id')
   ->having('nb_commentaires > 5');
```

Exécution de la requête

```
$query = $qb->getQuery()->getResult();
```

Récupération des résultats

Méthode	Retourne
`getResult()`	un tableau indicé d'entités
`getArrayResult()`	un tableau indicé de tableaux associatifs
`getScalarResult()`	un tableau indicé de valeurs
`getSingleResult()`	une entité, lève une exception NoResultException
`getOneOrNullResult()`	une entité lève une exception NonUniqueResultException
`getSingleScalarResult()`	une seule valeur

Requêtes paramètres

```
$qb->select('a')
    ->from(AppBundle:Auteur, 'a')
    ->where('a.actif= :actif')
    ->andWhere('a.langue= :langue');
```

Passage des paramètres

```
$qb->select('a')
    ->from(AppBundle:Auteur, 'a')
    ->where('a.actif= :actif')
    ->andWhere('a.langue= :langue');

$query = $qb->getQuery();

// Passage des paramètres
$query->setParameter('actif', 1);
$query->setParameter('langue', 'fr');

$result = $query->getArrayResult();
```

Syntaxe alternative

```
$qb->select('a')
    ->from(AppBundle:Auteur, 'a')
    ->where('a.actif= ?1')
    ->andWhere('a.langue= ?2');

$query = $qb->getQuery();

$query->setParameter(1, 1);
$query->setParameter(2, 'fr');
```

Update

```
// Construction de la requête
$qb
  ->update('AppBundle:Auteur', 'a')
  ->set('a.actif', ':actif');

// Passage des paramètres
$query = $qb->getQuery();
$query->setParameter('actif', 1);

// Exécution de la requête
$query->execute();
```

Delete

```
// Construction de la requête
$qb
  ->delete('AppBundle:Auteur', 'a')
  ->where('a.actif = :actif');

// Passage des paramètres
$query = $qb->getQuery();
$query->setParameter('actif', 0);

// Exécution de la requête
$query->execute();
```

Repository personnalisé

- Centralise les requêtes sur une entité
- Une classe héritant de EntityRepository
- Une annotation dans l’entité
- Classement dans un dossier Repository

La classe repository

```
namespace AppBundle\Entity\Repository;

use Doctrine\ORM\EntityRepository;
use Doctrine\ORM\QueryBuilder;

class AuteurRepository extends EntityRepository {

    public function getAuteursWithNumberOfPosts(){
        /*  L'entité est déjà sélectionnée
            puisque nous sommes sur le Repository
            de la classe Auteur
            Nous n'avons donc qu'à préciser l'alias
        */
        $qb = $this->createQueryBuilder('a')
            ->select('a, COUNT(p.id) as nb_posts')
            ->join('a.posts','p')
            ->addGroupBy('a.id');

        return $qb->getQuery();

    }
}
```

Le lien sur l'entité

```
/**
 * Auteur
 *
 * @ORM\Table(name="auteurs")
 * @ORM\Entity(
 * repositoryClass=
 * "AppBundle\Entity\Repository\AuteurRepository"
 * )
 */
class Auteur {
...
}
```

L'utilisation dans un contrôleur

```
public function listeAuteursAction()
    {
        $em = $this->getDoctrine()->getEntityManager();
        $repository = $em->getRepository('AppBundle:Auteur');

        $auteurs = $repository->getAuteursWithNumberOfPosts()
                 ->getResult();

        $this->render('auteur/liste.html.twig',
                      array('auteurs' => $auteurs
        );
    }
```

principe

Un champ de formulaire dont le contenu est alimenté par une entité

```
$builder
    ->add('category',
        'entity',
        array(
            'class'       => 'AppBundle:category',
            'property'    => 'categoryName',
            'empty_value' => 'Choisissez une catégorie'
        )
);
```

Type de contrôle

Options	Multiple = false	Multiple = true
Expanded = false	Categories Développement web Valider Développement web Développement mobile Graphisme Intégration Réseau `<select>` avec une seule option sélectionnable	Développement web Développement mobi Graphisme Categories Intégration `<select>` avec plusieurs options sélectionnables
Expanded = true	Categories Développement web Développement mobile Graphisme Intégration Réseau `<radio>`	Categories Développement web Développement mobile Graphisme Intégration Réseau `<checkbox>`

par défaut expanded et multiple sont false

```
array(
  'expanded' => true,
  'multiple' => true,
)
```

Requête

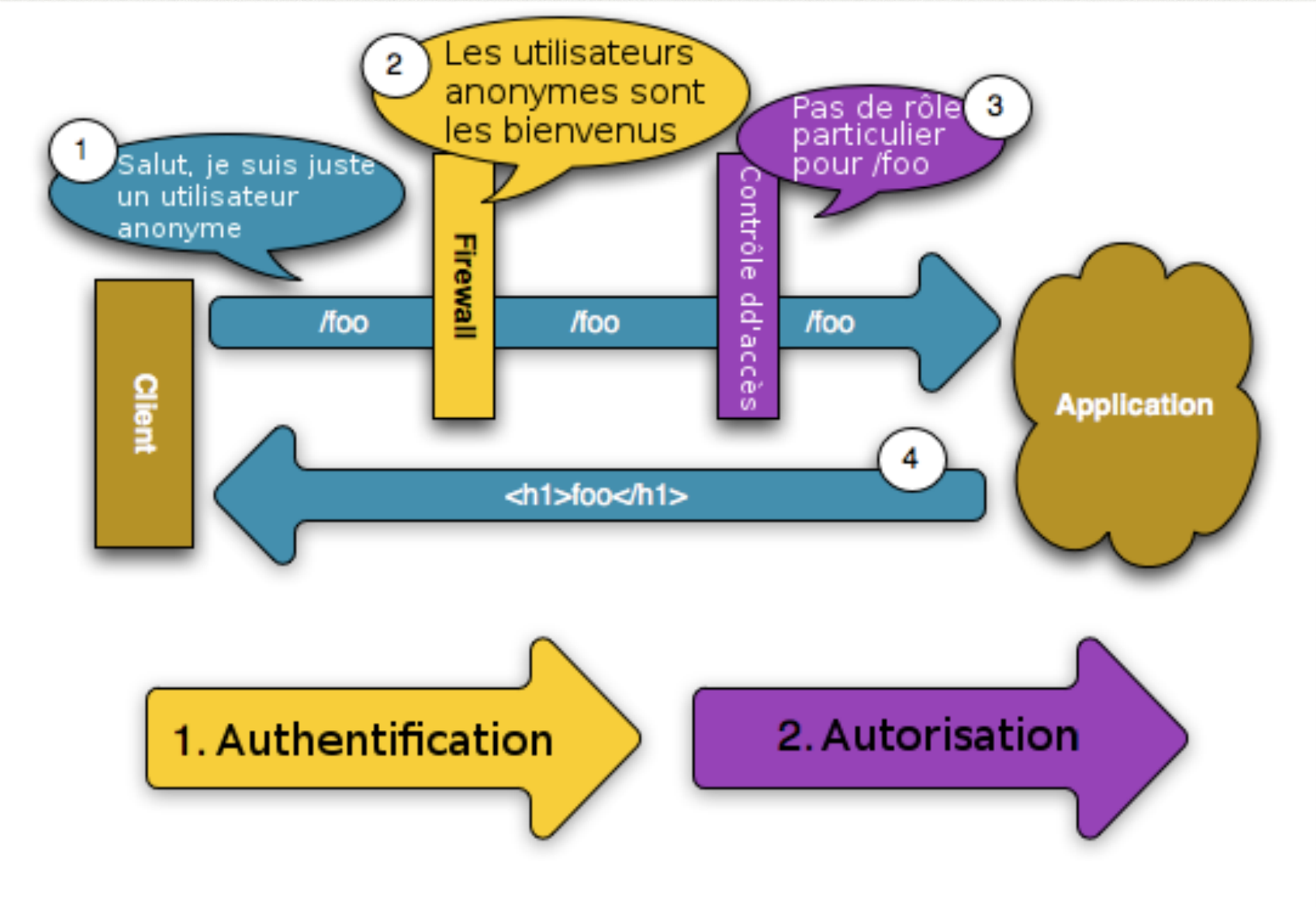
fonction anonyme

injection du repository

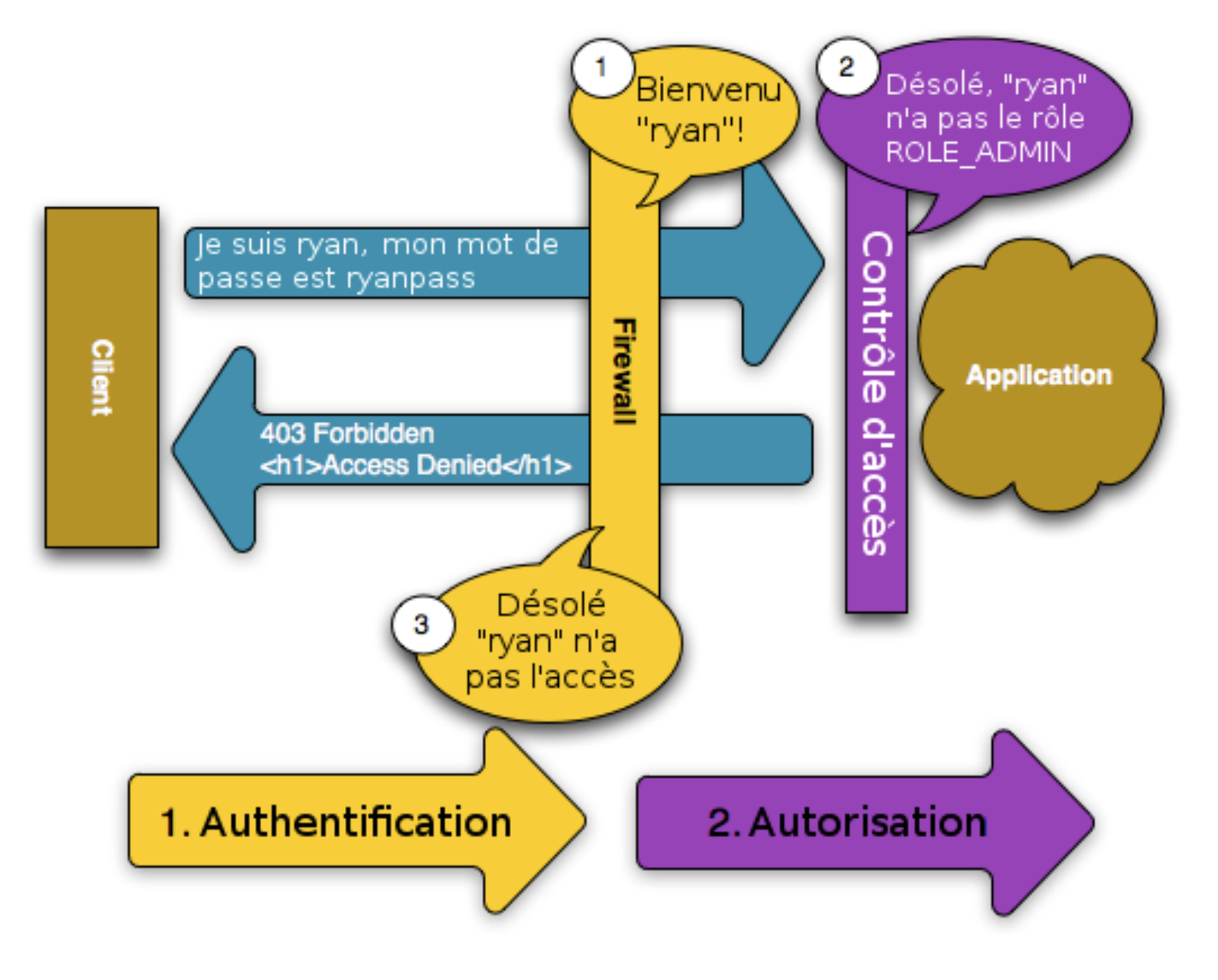
```
array(
  'query_builder' => function(EntityRepository $er){
     return $er->getActiveCategories()->getResult();
  },
)
```

méthode du repository de l’entité

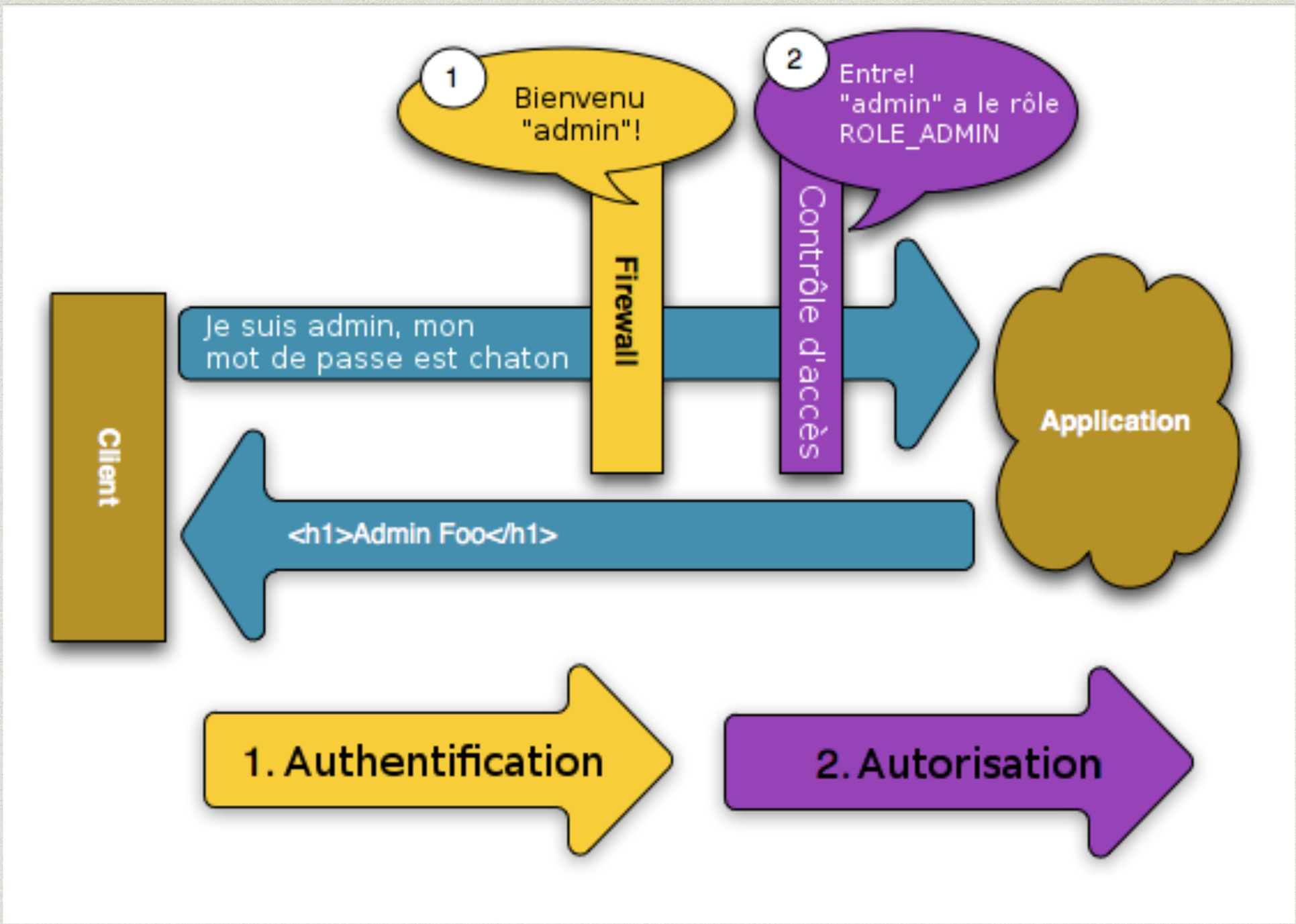
Anonyme


1
Salut, je suis juste un utilisateur anonyme
2
Les utilisateurs anonymes sont les bienvenus
3
Pas de rôle particulier pour /foo
Client
Firewall
Contrôle d'accès
Application
/foo
/foo
/foo
4
<h1>foo</h1>
1. Authentification
2. Autorisation

Authentifié non autorisé



Authentifié et autorisé



Composants

- Firewall (authentifie)
  - Provider (source de données)
  - Encoder (encryption du mot de passe)
- Access control (protège l’accès)

Configuration

app/config/security.yml

```
security:

  encoders:

  role_hierarchy:

  providers:

  firewalls:

  access_control:
```

Encoder

Méthode de hachage des mots de passes

```
encoders:
    Symfony\Component\Security\Core\User\User: plaintext
```

Providers

Source de données des utilisateurs

```
providers:
    in_memory: #nom arbitraire
      memory:
        users:
          user:  { password: userpass, roles: [ 'ROLE_USER' ] }
          admin: { password: adminpass, roles: [ 'ROLE_ADMIN' ] }
```

Role hierarchy

Gestion des rôles en cascades

```
security:
    role_hierarchy:
        ROLE_ADMIN:       ROLE_USER
        ROLE_SUPER_ADMIN: [ROLE_USER, ROLE_ADMIN, ROLE_ALLOWED_TO_SWITCH]
```

Firewall

Gestion de l'authentification

```
firewalls:

    #1er firewall
    dev:
      pattern: ^/(_(profiler|wdt)|css|images|js)/
      security: false

    #2e firewall
    main:
      #URL protégée par le firewall
      pattern:   ^/
      anonymous: true
```

Access Control

Gestion de l'autorisation

```
security:
    access_control:
        - { path: ^/admin, roles: ADMIN}
```

Le Firewall

```
firewalls:
    dev:
      pattern: ^/(_(profiler|wdt)|css|images|js)/
      security: false

    login_firewall:
      anonymous: true
      pattern: ^/login$

    main:
     pattern:   ^/
     anonymous: false
     provider: in_memory
     form_login:
        login_path: login_route
        check_path: login_check
     logout:
        path: logout_route
        target: /
```

Les firewalls sont testés de façon séquentielle,
dès qu’une correspondance est trouvée,
la recherche s’arrête et le firewall détecté s’applique.

Comme pour les routes, il faut donc définir
les firewalls les plus spécifiques en premier

Le routage

```yaml
# app/config/routing.yml

# ...

login_route:
    path: /login
    defaults:
        _controller: MonBundle:Default:login

# Ces deux dernières routes n'ont pas de contrôleur
# c'est Symfony qui se charge de gérer ces actions
login_check:
    path: /login_check

logout_route:
    path: /logout
```

Le formulaire

```
{% if error %}
  <div class="alert alert-danger">{{ error.message }}</div>
{% endif %}

<form action="{{ path('login_check') }}" method="post">
  <label for="username">Login :</label>
  <input type="text" id="username" name="_username"
  value="{{ lastUserName }}" />

  <label for="password">Mot de passe :</label>
  <input type="password" id="password" name="_password" />
  <br />
  <button type="submit">Connexion</button>
</form>
```

Le contrôleur

```
public function loginAction(Request $request){

    //Si l'utilisateur est identifié alors redirection
    if ($this->get('security.authorization_checker')
        ->isGranted('IS_AUTHENTICATED_REMEMBERED')
    ) {
        return $this->redirectToRoute('accueil');
    }

    //Sinon récupération des erreurs et affichage du formulaire de login
    $authenticationUtils = $this->get('security.authentication_utils');

    return $this->render(
        'MonBundle:Default:login.html.twig',
        [
            'lastUserName' => $authenticationUtils->getLastUserName(),
            'error' => $authenticationUtils->getLastAuthenticationError()
        ]
    );

}
```

Les pseudo rôles

IS_AUTHENTICATED_ANONYMOUSLY	Tous les utilisateurs même anonymes possèdent ce pseudo rôle
IS_AUTHENTICATED_REMEMBERED	Tous les utilisateurs authentifiés hors anonymes donc
IS_AUTHENTICATED_FULLY	Les utilisateurs authentifiés par un formulaire. Exclu les utilisateurs automatiquement authentifiés grâce à un cookie

Récupérer l'utilisateur

dans un contrôleur

```
$security = $container->get('security.token_storage');
$user = $security->getToken()->getUser();

//Ou ceci si le contrôleur étends la classe Controller
$user = $this->getUser();
```

dans Twig

```
{% if app.user is defined %}
  Hello {{ app.user.username }}
  - {{ app.user.email }}
{% endif %}
```

LE CONTRÔLE DES ACCÈS

Le contrôle d’accès

```
security:
    # ...

    access_control:
        - { path: ^/admin, roles: ROLE_ADMIN }
        - { path: ^/, roles: ROLE_USER }
```

Dans un contrôleur

```
public function indexAction()
{
    $user = $this->getUser()
    $this->denyAccessUnlessGranted(
            'ROLE_ADMIN',
            $user,
            'Accès interdit'
    );
}
```

Dans un contrôleur avec des annotations

```
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Security;

/**
 * @Security("has_role('ROLE_ADMIN')")
 */
public function indexAction()
{
    // ...
}
```

Dans un gabarit Twig

```
{% if is_granted('ROLE_ADMIN') %}
    <a href="...">Déconnexion</a>
{% endif %}
```