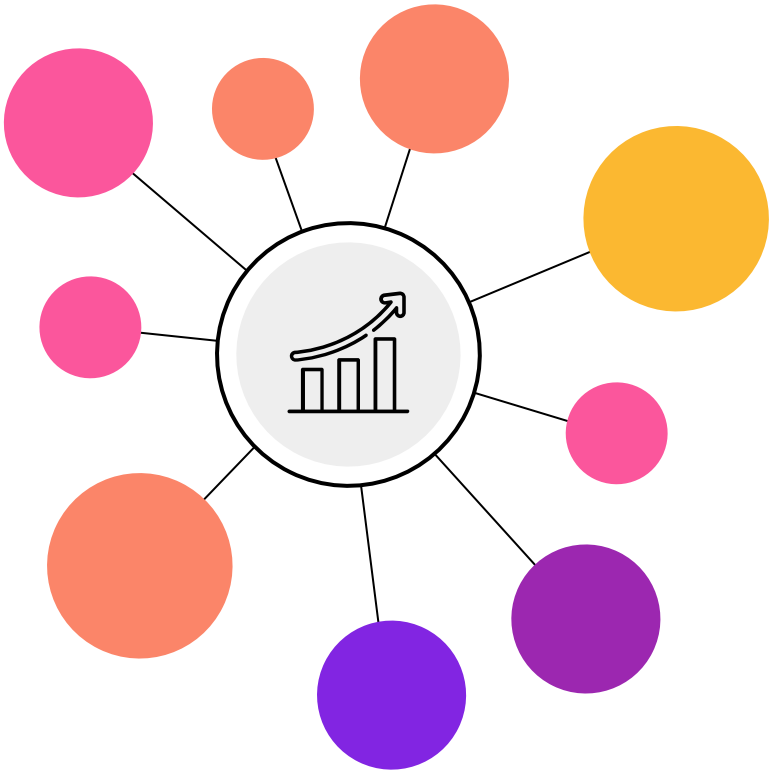
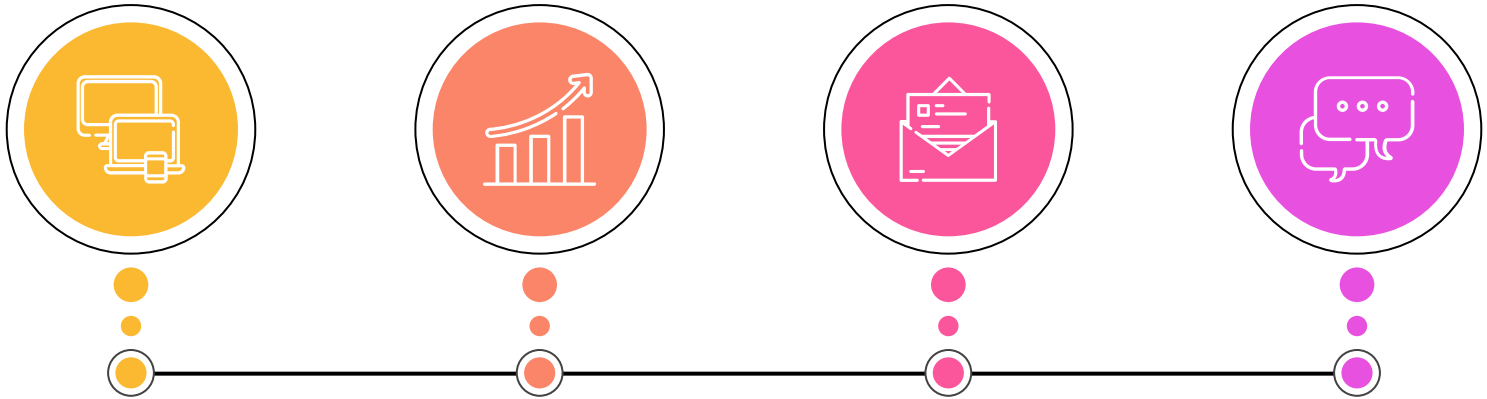


R209 Initiation au développement web

Prérequis



R107
Fondamentaux de la programmation

R109
Introduction aux technologies Web

R207
Sources de données

R208
Analyse et traitement de données structurées

if (lacunes>0) then do_révisions([R107,R109,R207,R208,. . .]);

Contenus

Mise en forme de pages Web

balises HTML avancées ;
- structure d'une page avec son DOM ;
- CSS avancé ou Framework ;
- initiation au dynamisme côté client

Interaction client-serveur

Interrogation d'un SGBD ou d'une API

Sensibilisation à la sécurisation de sites

failles XSS, XSS stockée, injections SQL

Scripts côté serveur.

Projet en mode Saas

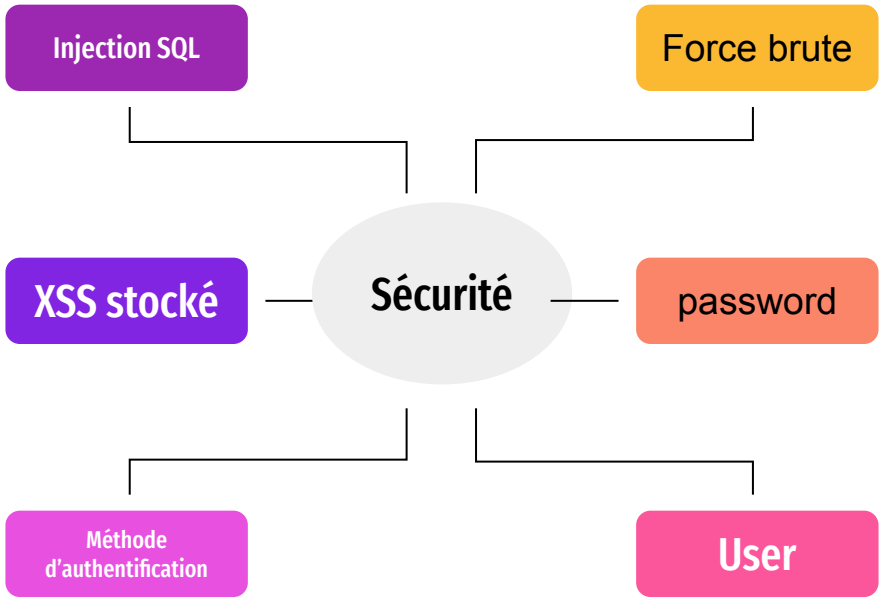
Le **mode Saas** est la mise à disposition d'un logiciel accessible aux utilisateurs via internet. Aucune installation sur les serveurs de l'entreprise cliente n'est requise. Chaque utilisateur dispose d'un compte, avec des niveaux de droit variables, lui permettant d'accéder au logiciel.

Informations sur la sécurité

Quelques failles de sécurité

Groupe de méthodes d'exploitation de faille de sécurité d'une application interagissant avec une base de données. Elle permet d'injecter dans la requête SQL en cours un morceau de requête non prévu par le système et pouvant en compromettre la sécurité.

Le cross-site scripting est un type de faille de sécurité des sites web permettant d'injecter du contenu dans une page, provoquant ainsi des actions sur les navigateurs web visitant la page



L'attaque par force brute est une méthode utilisée en cryptanalyse pour trouver un mot de passe ou une clé. Il s'agit de tester, une à une, toutes les combinaisons possibles. Cette méthode est en général considérée comme la plus simple concevable

Faiblesse des passwords

Stratégie de sécurité

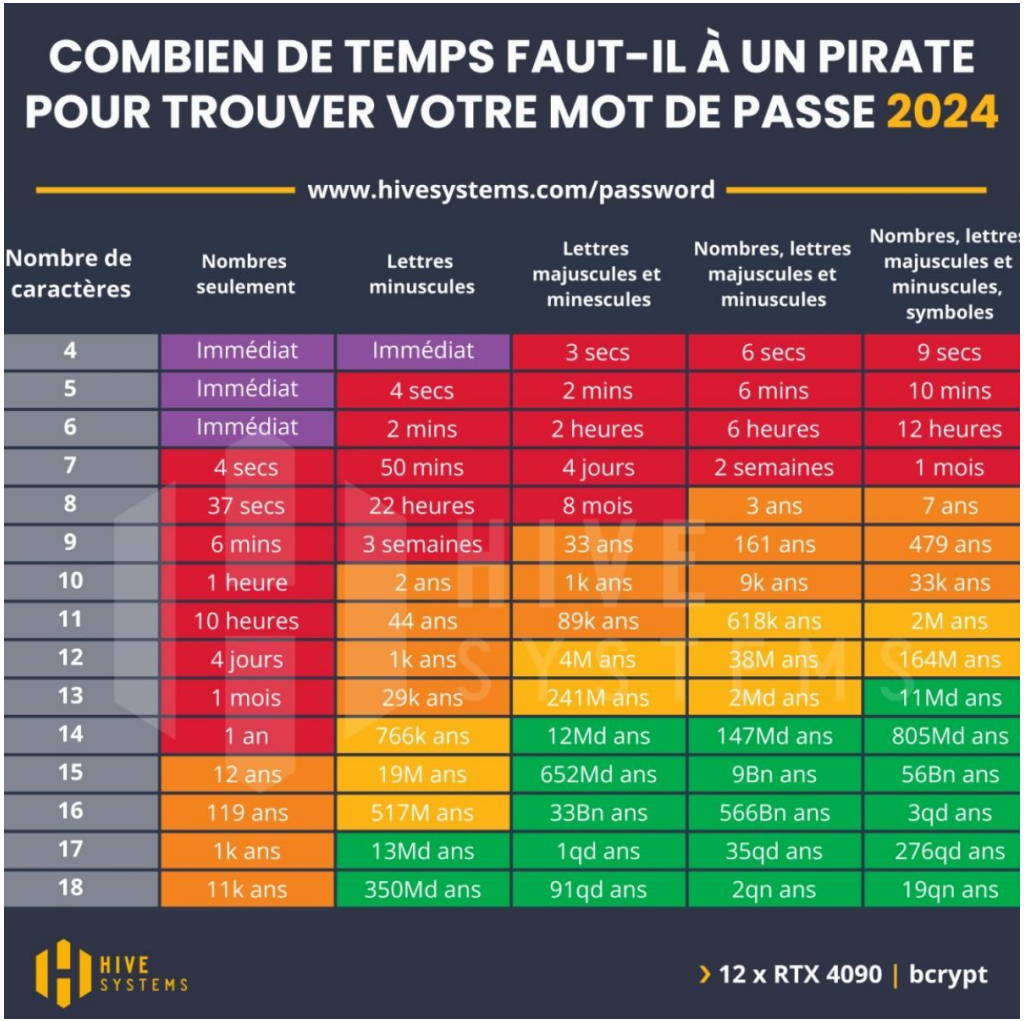
L'utilisateur de l'application

Depuis plus d’une décennie, les compétitions sportives internationales majeures font régulièrement l’objet d’attaques informatiques. Ces évènements, en raison de l’intérêt qu’ils suscitent chez des centaines de millions de personnes et des flux financiers importants qu’ils génèrent, constituent une opportunité d’agir pour des attaquants informatiques aux motivations diverses. En effet, ces derniers peuvent chercher à perpétrer des actes malveillants à des fins lucratives, de déstabilisation (notamment liée au contexte géopolitique) ou encore d’espionnage. Les Jeux olympiques et paralympiques 2024, organisés en France, pourraient être les cibles d’attaques informatiques de ce type.

L’organisation et le déroulement d’un évènement sportif de cette ampleur nécessitent en effet la mise en œuvre de nombreux systèmes d’information, que ceux-ci appartiennent au pays hôte, aux organisateurs ou à leurs prestataires, sous-traitants, sponsors ou toute autre entité participant d’une façon ou d’une autre à cet évènement. Ces attaques sont susceptibles de perturber de manière substantielle les compétitions sportives d’ampleur.

Texte de l’ANSSI

Agence nationale de sécurité des systèmes d’information.
Article de avril 2024



Nombre de caractères	Nombres seulement	Lettres minuscules	Lettres majuscules et minescules	Nombres, lettres majuscules et minuscules	Nombres, lettres majuscules et minuscules, symboles
4	Immédiat	Immédiat	3 secs	6 secs	9 secs
5	Immédiat	4 secs	2 mins	6 mins	10 mins
6	Immédiat	2 mins	2 heures	6 heures	12 heures
7	4 secs	50 mins	4 jours	2 semaines	1 mois
8	37 secs	22 heures	8 mois	3 ans	7 ans
9	6 mins	3 semaines	33 ans	161 ans	479 ans
10	1 heure	2 ans	1k ans	9k ans	33k ans
11	10 heures	44 ans	89k ans	618k ans	2M ans
12	4 jours	1k ans	4M ans	38M ans	164M ans
13	1 mois	29k ans	241M ans	2Md ans	11Md ans
14	1 an	766k ans	12Md ans	147Md ans	805Md ans
15	12 ans	19M ans	652Md ans	9Bn ans	56Bn ans
16	119 ans	517M ans	33Bn ans	566Bn ans	3qd ans
17	1k ans	13Md ans	1qd ans	35qd ans	276qd ans
18	11k ans	350Md ans	91qd ans	2qn ans	19qn ans

Chiffres 2021

Chiffres 2021

SecNumacadémie.gouv.fr
Formez-vous à la sécurité du numérique

Bienvenue sur le MOOC de l'ANSSI.

Vous y trouverez l'ensemble des informations pour vous **initier à la cybersécurité**, approfondir vos connaissances, et ainsi **agir efficacement sur la protection de vos outils numériques**. Ce dispositif est accessible gratuitement. Le suivi intégral de ce dispositif vous fera bénéficier d'une attestation de réussite.

Accéder au MOOC de l'ANSSI

Pour information

https://secnumacademie.gouv.fr/

Le RGPD

Le sigle RGPD signifie « **Règlement Général sur la Protection des Données** » (en anglais « General Data Protection Regulation » ou GDPR).
Le RGPD encadre le traitement des données personnelles sur le territoire de l'Union européenne.

[david.lacan@aresformation.com] Detection d'une attaque sur l'IP 51.222.75.2

noreply@soyoustart.com
À moi

SAS OVH - https://www.soyoustart.com
2 rue Kellermann
BP 80157
59100 Roubaix

Madame, Monsieur,

Nous venons de détecter une attaque sur l'adresse IP 51.222.75.212.

Afin de protéger votre infrastructure, nous avons aspiré votre
traffic sur notre infrastructure de mitigation.

Toute l'attaque sera ainsi filtrée par notre infrastructure, et seul
le traffic légitime arrivera jusqu'à vos serveurs.

A la fin de l'attaque, votre infrastructure sera immédiatement retirée de la mitigation.

Pour plus d'informations sur l'infrastructure de mitigation OVH : http://www.soyoustart.com/fr/anti-ddos/

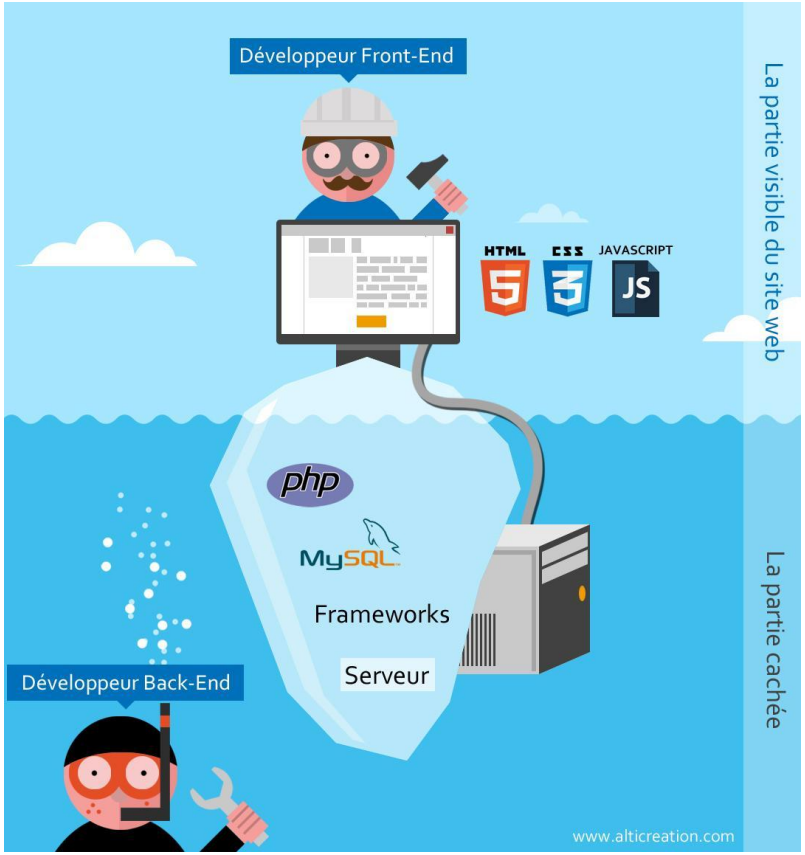
Cordialement,
L'équipe So you Start

Vous avez une question ?
Contactez notre support technique et commercial par email ou au 09 72 100 111.
N'hésitez pas à rejoindre notre communauté sur le forum : https://community.ovh.com/

So you Start
SAS OVH - https://www.ovh.com/
2 rue Kellermann
BP 80157
59100 Roubaix

exemple attaque DDOS

Le développement Web



Le développement Web

Le boulversement de l’IA

Dans le code

- **Automatisation de la programmation**
- **Développement assisté par IA**
- **Amélioration des tests et de la détection des bugs**
- **Optimisation des performances**
- **Gestion et analyse des données**
- **Développement de nouvelles compétences**

Dans le travail

- **Automatisation des tâches répétitives**
- **Automatisation des tâches répétitives**
- **Automatisation des tâches répétitives**
- **collaboration homme-Machine**

    - **=> Transformation des compétences et des emplois**


PROJET

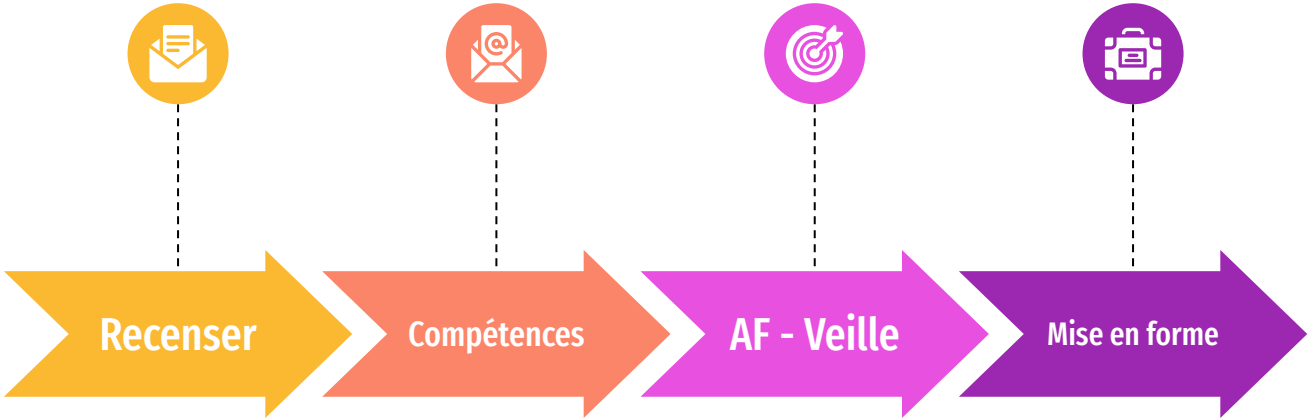
Difficultés

- Transversalité des savoirs
- Besoins des acquis
- Monter en compétences
- Mise en place de l’environnement de travail qui peut être chronophage

Notre projet : portfolio

Créer une plateforme permettant d’inscrire vos expériences pro et perso afin d’afficher votre portfolio mis à jour

User story



Formations & expériences

Lier des compétences

Décision

Visuel final

User story

Mettre en place l'env	Créer le modèle	Développer	Mettre en prod
			
technique	BDD	Codage + test	Démarrage

Modèle du projet

User
id
nom
prénom
email
naissance
role
tel
description
objectif
permis (oui/non)
vehicule (oui/non)
linkedin
youtube
insta
tiktok

Formation
id
nomEtablissement
villeEtablissement
distanciel (oui/non)
dateD
dateF
nomformation
option
#niveau
#type

Type
id
nom

Competences
id
nom

Niveau
id
nom

AFVeille
id
nom
acquisition
veille continue (oui/non)
#type

Experiences
id
dateD
dateF (null)
nomEtablissement
villeEtablissement
nomPoste
mission
responsable (oui/non)
#type

Rappel notion Base de Données

1) **clé primaire**
   a) La clé primaire est un ou plusieurs attributs d'une table qui permet d'identifier de manière unique chaque enregistrement (ou ligne) de cette table. Elle assure l'unicité et l'intégrité des données au sein de la table.

2) **clé étrangère**
   a) Une clé étrangère (Foreign Key) est un ou plusieurs attributs d'une table qui établit et impose un lien entre les données de deux tables. Elle est utilisée pour maintenir l'intégrité référentielle entre les tables en s'assurant que les valeurs de la clé étrangère dans une table correspondent aux valeurs de la clé primaire dans une autre table.

Rappel notion Base de Données : les relations

OneToOne (Un-à-Un)

Une relation OneToOne signifie qu'une entité est associée à une seule instance d'une autre entité et vice versa. C'est souvent utilisé pour représenter des relations exclusives entre deux entités.

OneToMany (Un-à-Plusieurs)

Une relation OneToMany signifie qu'une entité est associée à plusieurs instances d'une autre entité. C'est souvent utilisé pour représenter des collections ou des listes d'éléments.

Une catégorie peut contenir plusieurs produits/Chaque produit appartient à une seule catégorie.

ManyToOne (Plusieurs-à-Un)

Une relation ManyToOne signifie que plusieurs instances d'une entité peuvent être associées à une seule instance d'une autre entité. Cette relation est souvent la contrepartie de la relation OneToMany.

Un client peut passer plusieurs commandes /Chaque commande est associée à un seul client.

ManyToMany (Plusieurs-à-Plusieurs)

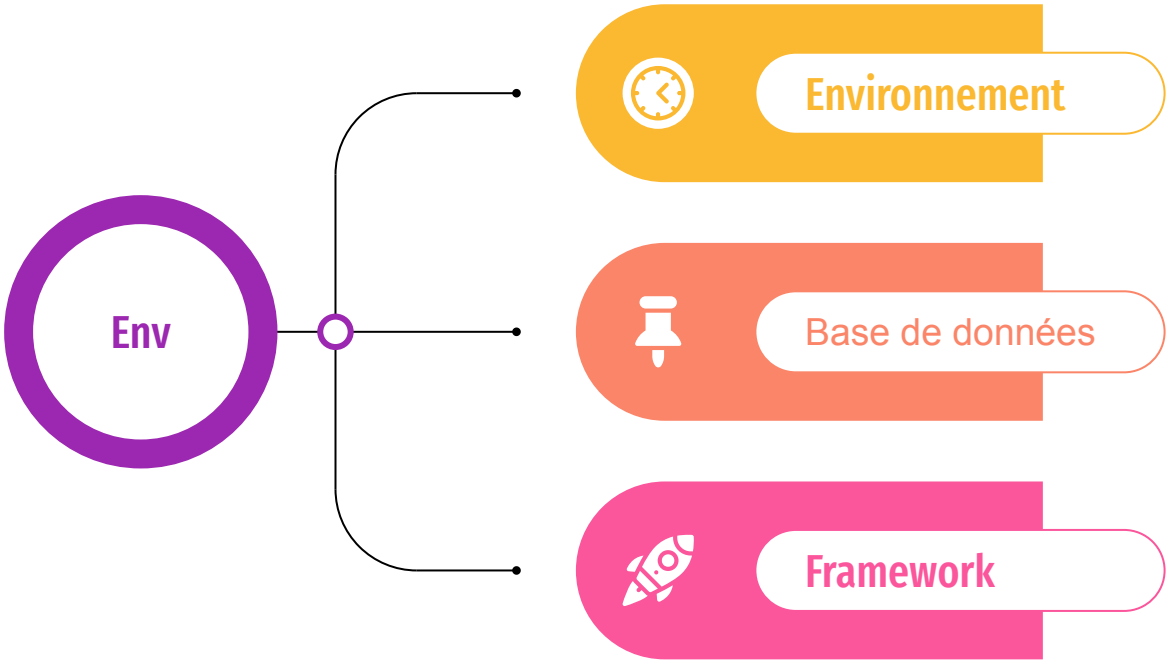
Une relation ManyToMany signifie que plusieurs instances d'une entité peuvent être associées à plusieurs instances d'une autre entité. C'est souvent utilisé pour représenter des relations bidirectionnelles complexes où plusieurs entités peuvent être liées de manière non exclusive.

Rappel notion Base de Données : les relations

- Une formation a un type : ManyToOne
- Une formation a un niveau : ManyToOne
- Une experience a un type : ManyToOne
- une AF-Veille a un type : ManyTone
- **Plusieurs formations peuvent avoir plusieurs compétences : ManyToMany**
- **Plusieurs expériences peuvent avoir plusieurs compétences : ManyToMany**
- **Plusieurs Af-Veille peuvent avoir plusieurs compétences : ManyToMany**

Méthode de travail

Environnement Technique

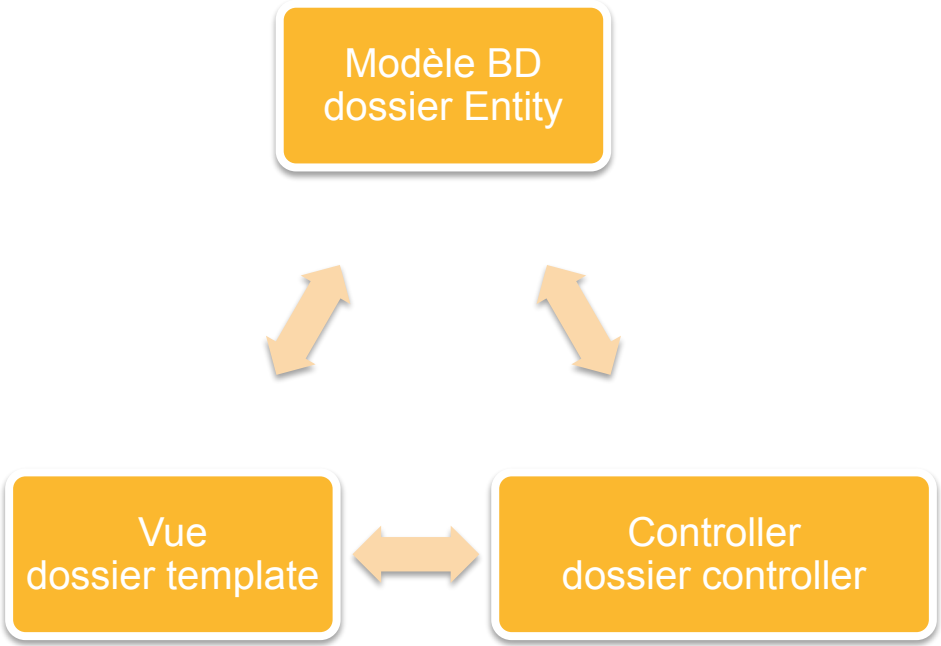


Apache, php, composer, git

Maria db – mysql +
phpmyadmin (visualisation)

Symfony / bootstrap
PHP

Modèle MVC


Modèle BD
dossier Entity
Vue
dossier template
Controller
dossier controller

Démarrage projet symfony

Symfony new cv --full

Création du projet

Liens avec la BD

- Avoir un user qui a les droits sur une bd cv
- - modifier le fichier .env

Symfony console make:user

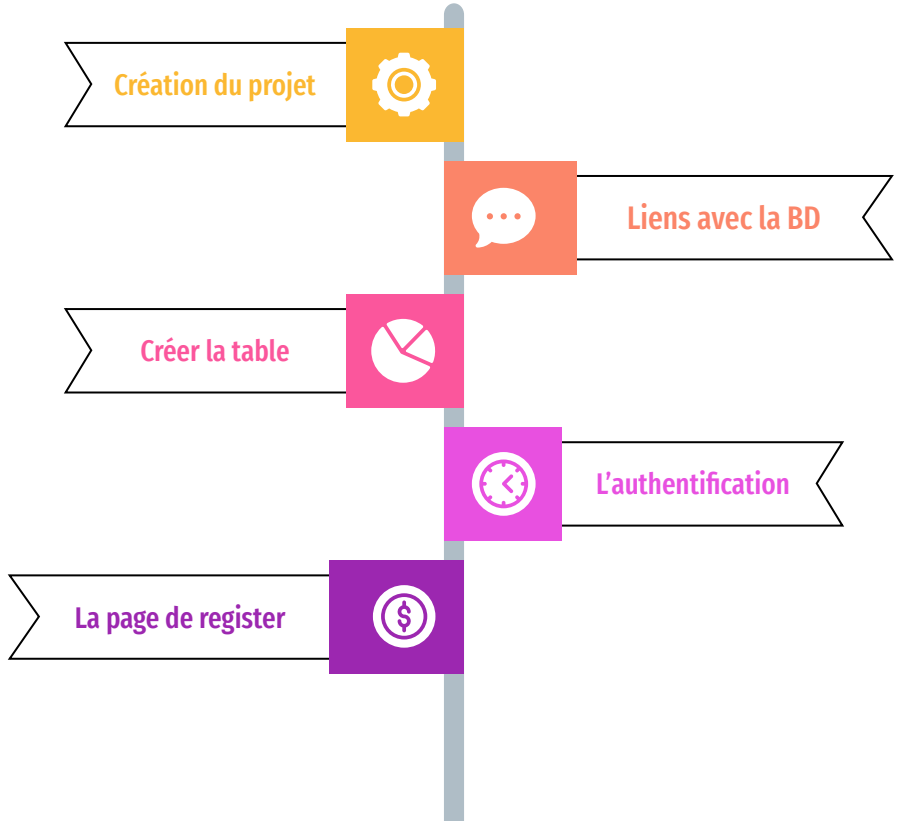
Créer la table

L'authentification

Symfony console make:auth

Symfony console make:registration-form

La page de register

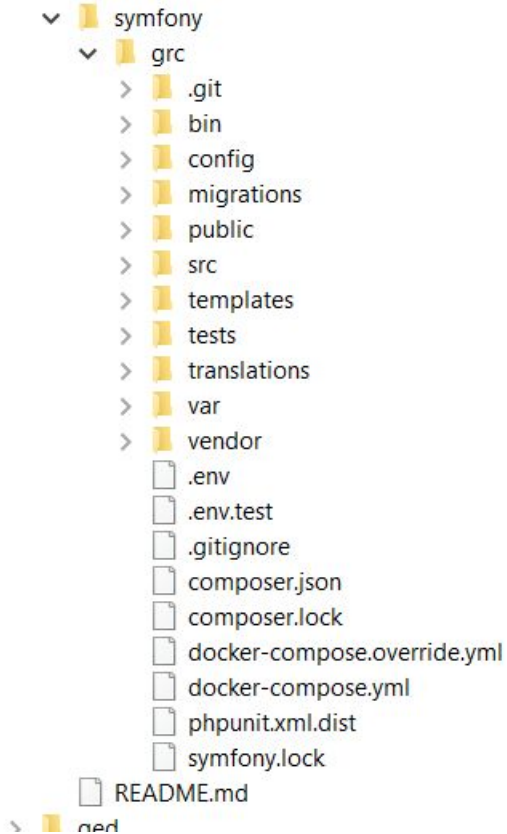


Arborescence

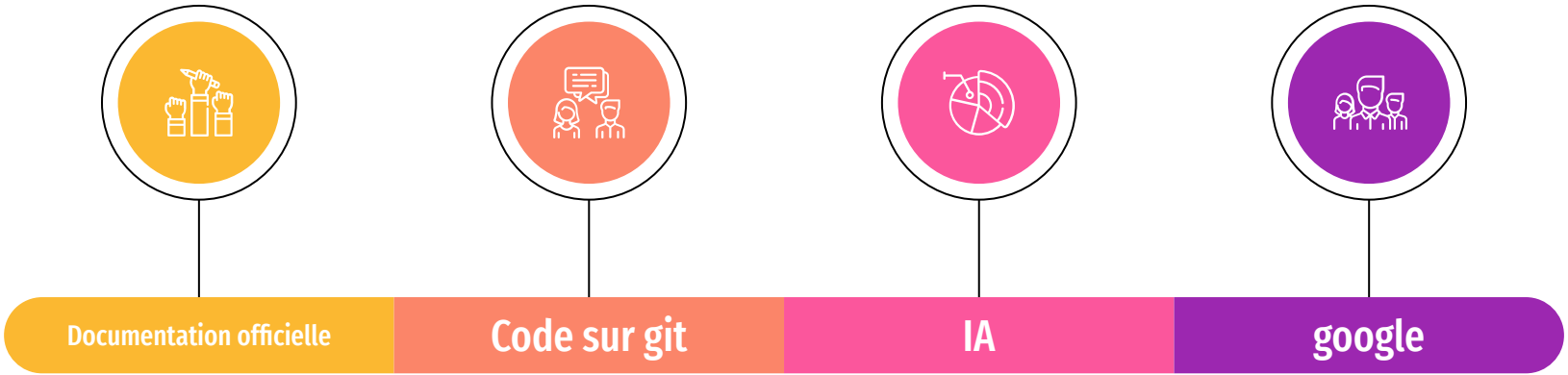
src / :
- Controller tous les contrôleurs
- Entity toutes les tables sous la forme Nom.php
- Repository un fichier par table pour créer des requêtes

templates/ :
- Liste des dossiers (un par controller) -> fichier twig (front)
- Fichier base.html.twig qui va contenir notre template et le menu

.env :
- Fichier qui permet d'identifier le user et la base de données



Support pour ce module



Symfony, bootstrap ...	Code sur github	IA	...

Pour la bonne réussite de ce module

Les prérequis

Avoir les bases
Ne pas prendre du retard

Comprendre le fonctionnement de symfony

Routage, interrogation bd

Env de développement stable

Env portable
Facilement reprogrammable
sauvegarde

Modèle – base de données

Créer le modèle rapidement, correctement et définitivement

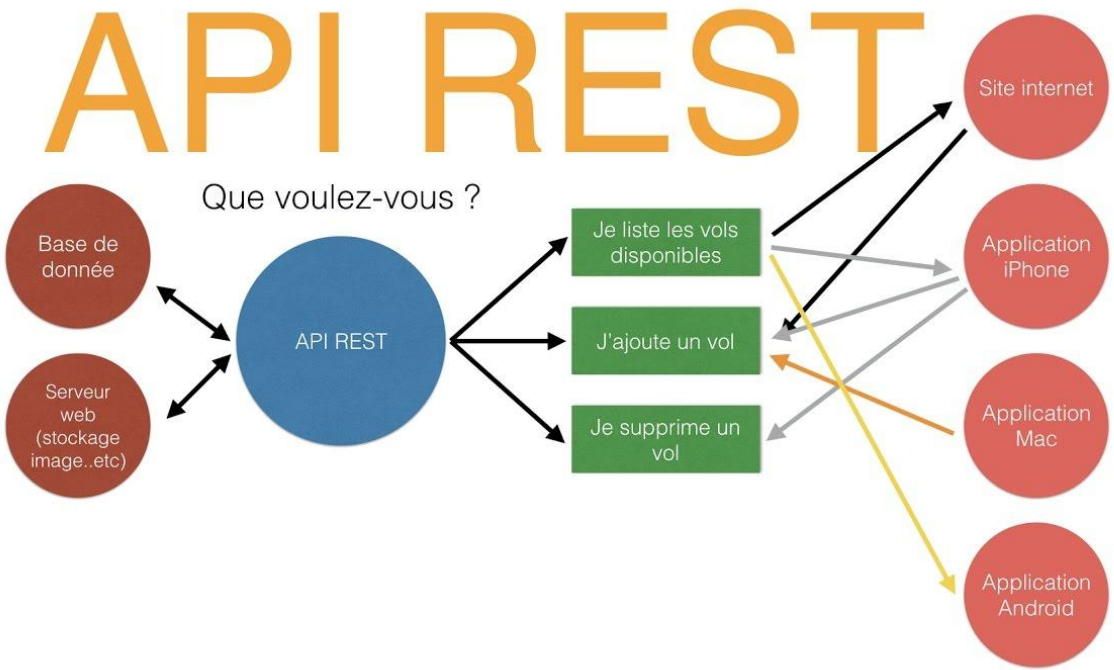
utiliser les outils

bootstrap, bootswatch

Création de l’API

Une **API (Application Programming Interface)** est un ensemble de règles et de protocoles qui permet à différentes applications logicielle de communiquer entre elles. Elle définit les méthodes et les formats de données que les applications peuvent utiliser pour échanger des informations et accéder aux fonctionnalités d'un système logiciel.

Créer une API



La sérialisation

Normalisation / Sérialisation

- La normalisation permet de transformer des objets en tableau associatif
- La sérialisation permet de transformer des tableaux associatifs en chaîne de caractère (XML, json, csv, …)

Que souhaitons nous

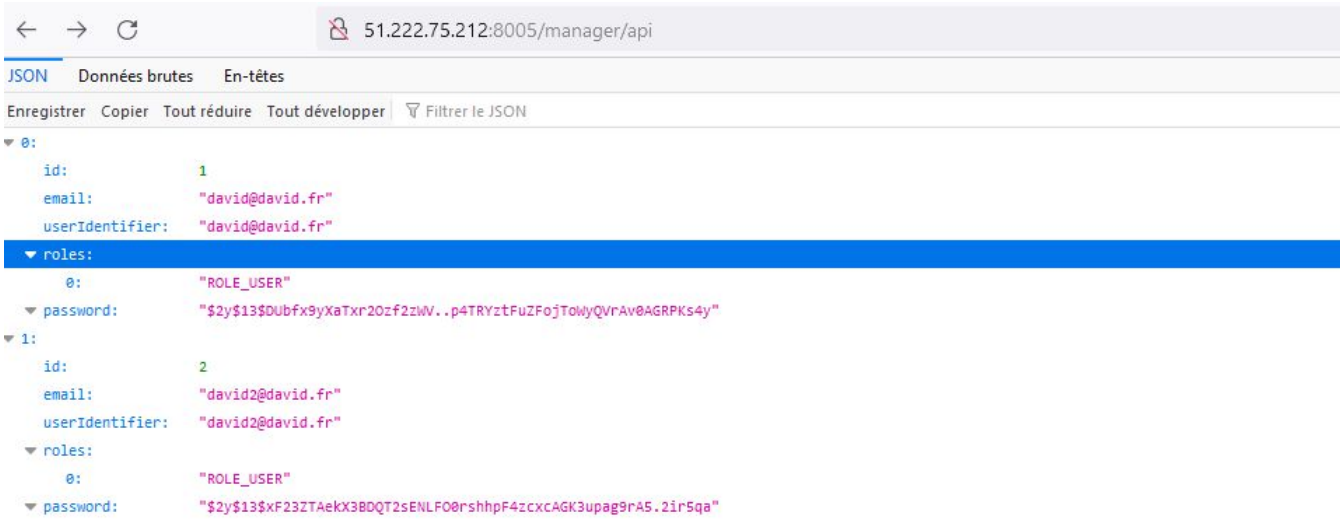
Nous souhaitons que lorsqu'un navigateur ou un code quelconque se rend à une adresse précise, la plateforme va chercher des informations à la base de données qui lui renvoie des données interprétables et compréhensibles facilement.

Obtenir un format json pour remplacer nos objets php qui arrivent sous forme de tableaux associatifs.

Idée générale :
Lorsque l'on reçoit une requête HTTP , on répond par du texte. (json, csv, xml, …)

Exemple : liste des users de mon application

L’adresse : http://51.222.75.212:8005/manager/api renvoie la liste des user de mon application



Exemple : liste des users de mon application

Code de la route /manager/api

```php
<?php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;
use App\Entity\User;
use App\Repository\UserRepository;
use Symfony\Component\Serializer\SerializerInterface;
use Symfony\Component\Serializer\Serializer;


class ManagerController extends AbstractController
{
```

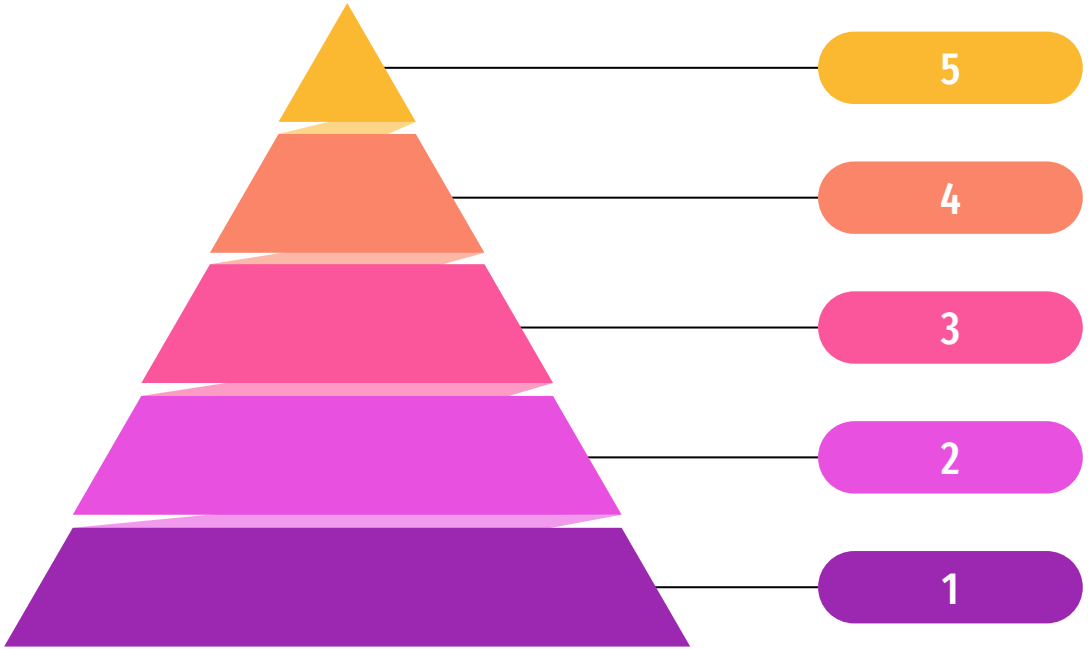
```php
    /**
     * @Route("/manager/api", name="app_manager_api")
     * @Method({"GET", "POST"})
     */
    public function api(UserRepository $userRepo): Response
    {

        $recupUserObject = $userRepo->findAll();
        $response = $this->json($recupUserObject,200,[]);
        return $response;


    }
```

Exemple : liste des users de mon application

Code de la route /manager/api

```php
/**
 * @Route("/manager/api", name="app_manager_api")
 * @Method({"GET", "POST"})
 */
public function api(UserRepository $userRepo): Response
{

    $recupUserObject = $userRepo->findAll();
    $response = $this->json($recupUserObject,200,[]);
    return $response;

}
```

Feuille de route



Mise en place de l'API

Création des fonctions, des routes

Mise en place du visuel fichier et de la nav bar base.html.twig

Création du modèle en ligne de commande

Mise en place environnement de dev

Symfony, lien bd, securité, test

La technique : environnement

Schéma des outils

Votre machine

- un répertoire de travail
- un ide

côté client

- un navigateur

côté serveur

- un serveur web
- env php
- composer
- git
- symfony
- serveur maria db
- phpmyadmin

VM
serveur distant

git
documentation
IA

Liste des commandes

Liste des commandes de départ

Symfony new grc –full
Modification du fichier.env
Symfony console doctrine:database:create
Symfony console make:user
Symfony console make:auth
Symfony console make:registration-form
Symfony console serve –d
Test d’accès via un navigateur

Liste des commandes

Symfony console make:controller
Symfony console make:entity
Symfony console make:migration
Symfony console doctrine:migrations:migrate

Le fichier .env

Liste des commandes de départ

Le fichier .env se situe à la racine
Il permet d'appeler les modules : mail, api et notamment base de données.

1) vous devez inscrire la ligne :
   DATABASE_URL="mysql://identifiant:mdp@127.0.0.1:3306/grc"

2) puis en ligne de commande, à la racine de votre projet vous devez taper :
   Symfony console doctrine:database:create

La technique : code php

Objet

1) En PHP, un objet est une instance d'une classe.
2) Les objets sont au cœur de la programmation orientée objet (POO) et permettent de regrouper des données et des comportements associés en un seul concept cohérent.
3) Cela facilite la gestion et l'organisation du code, le rend plus modulaire, réutilisable et maintenable.

Objet

1) **Classe** : Une classe est un modèle ou un plan à partir duquel des objets sont créés. Elle définit les propriétés (attributs) et les méthodes (comportements) que les objets de cette classe auront.
2) **Objet** : Un objet est une instance concrète d'une classe. Lorsque vous créez un objet, vous allouez de la mémoire pour une instance de la classe et vous pouvez accéder aux propriétés et aux méthodes définies dans cette classe.
3) **Propriétés** : Les propriétés sont des variables qui appartiennent à une classe. Elles définissent les caractéristiques de la classe.
4) **Méthodes** : Les méthodes sont des fonctions qui appartiennent à une classe. Elles définissent les comportements que les objets de la classe peuvent réaliser.

get et set

Les getters et les setters sont des méthodes utilisées en programmation orientée objet pour accéder et modifier les valeurs des propriétés privées ou protégées d'une classe.
L'utilisation de getters et setters permet de contrôler et encapsuler l'accès aux données, ajoutant ainsi des niveaux de sécurité et de validation.

get et set

```php
<?php
class Person
{
    private $name;
    private $age;

    // Getter pour la propriété $name
    public function getName()
    {
        return $this->name;
    }

    // Setter pour la propriété $name
    public function setName($name)
    {
        // Vous pouvez ajouter des validations ici
        if (is_string($name) && !empty($name)) {
            $this->name = $name;
        } else {
            throw new Exception("Invalid name");
        }
    }

    // Getter pour la propriété $age
    public function getAge()
    {
        return $this->age;
    }

    // Setter pour la propriété $age
    public function setAge($age)
    {
        // Vous pouvez ajouter des validations ici
        if (is_int($age) && $age > 0) {
            $this->age = $age;
        } else {
            throw new Exception("Invalid age");
        }
    }
}

// Utilisation de la classe Person
$person = new Person();
$person->setName("John Doe");
$person->setAge(30);

echo "Name: " . $person->getName() . "<br>";
echo "Age: " . $person->getAge();
?>
```

Le fichier base.html.twig - Exemple

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>{% block title %}Welcome!{% endblock %}</title>
        {% block stylesheets %}{% endblock %}
    </head>
    <body>
        <nav><!-- Contenu de ma navbar --></nav>

        {% block body %}{% endblock %}

        <footer><!-- Contenu de mon footer --></footer>

        {% block javascripts %}{% endblock %}
    </body>
</html>
```

```
{# Template général de la page #}
{% extends 'base.html.twig' %}

{# Titre de la page #}
{% block title %}Mon titre{% endblock %}

{# Ici vous mettez vos codes HTML et balises
Twig au besoin #}
{% block body %}
    <h1>Mon super titre H1</h1>
    <p>Un paragraphe contenant pleins de mots à
lire</p>
{% endblock %}
```

Les fichiers twig

```php
/**
 * @Route("/home", name="home")
 */
public function index()
{
    return $this->render('home/index.html.twig', [
        'firstname' => 'John',
    ]);
}
```

controller

Vue

```twig
{# Template générale de la page #}
{% extends 'base.html.twig' %}

{# Titre de la page #}
{% block title %}Mon titre{% endblock %}

{# Ici vous mettez vos codes HTML et balises Twig au besoin #}
{% block body %}
    <p>Bonjour {{ firstname }}</p>
{% endblock %}
```

Le fichier base.html.twig

apport du fichier base.html.twig

Le fichier de vue va hériter du fichier de base et remplir les parties du squelette grâce aux blocs.

Le fichier "*base.html.twig*" va récupérer le contenu du bloc `title` et le placer en lieu et place du sien et faire la même chose pour le bloc `body`. Cela vous donnera un fichier HTML complet et propre. Pour ce qui est des blocs `stylesheets` et `javascripts`, il ne remplira rien, car ces blocs n'existent pas dans le fichier "*index.html.twig*" et il fera de même avec `title` et `body` s'ils sont manquants.

Exemple échange controller avec vue insertion en base de données

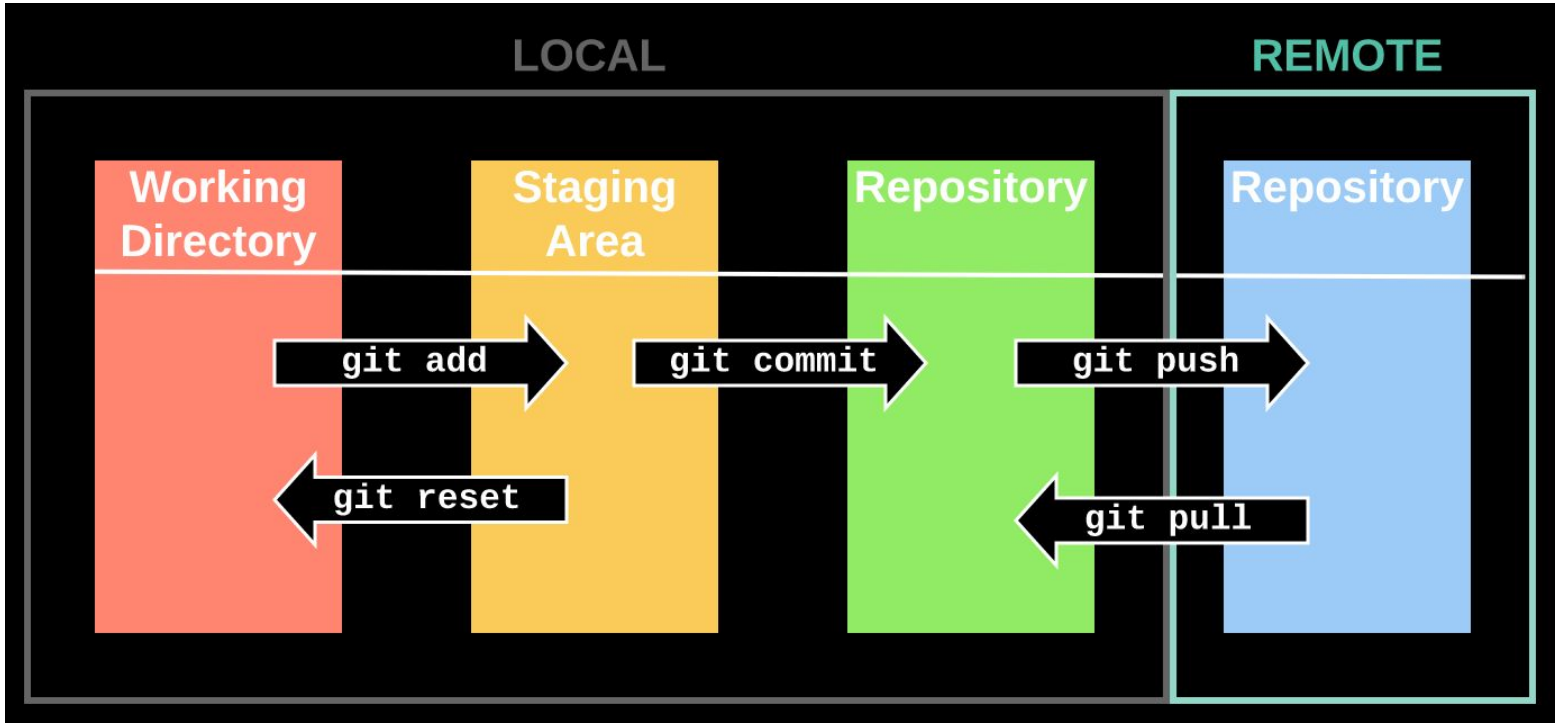
```php
namespace App\Entity;
use App\Repository\VilleRepository;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;
#[ORM\Entity(repositoryClass: VilleRepository::class)]
class Ville
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column(type: 'integer')]
    private $id;

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

    #[ORM\OneToMany(mappedBy: 'ville', targetEntity: Diplome::class)]
    private $diplomes;

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

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

    public function getNom(): ?string
    {
        return $this->nom;
    }

    public function setNom(string $nom): self
    {
        $this->nom = $nom;

        return $this;
```

```php
<?php

namespace App\Controller;

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

use Symfony\Component\HttpFoundation\Request;
use App\Entity\Ville;
use Doctrine\ORM\EntityManagerInterface;

class FormulaireController extends AbstractController
{
    #[Route('/insertVille', name: 'app_insert_ville')]
    public function index(): Response
    {
        return $this->render('formulaire/insertVille.html.twig', [
            'controller_name' => 'nouvelle ville',
        ]);
    }
    #[Route('/ville', name: 'app_ville')]
    public function ville(Request $request, EntityManagerInterface $manager): Response
    {
        $ville = new Ville();
        $ville->setNom($request->request->get("ville"));
        $manager->persist($ville);
        $manager->flush();
        return $this->render('formulaire/insertVille.html.twig', [
            'controller_name' => 'nouvelle ville',
        ]);
    }
```

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

{% block title %}Hello FormulaireController!{% endblock %}

{% block body %}
<style>
    .example-wrapper { margin: 1em auto; max-width: 800px; width: 95%; font: 18px/1.5 sans-serif; }
    .example-wrapper code { background: #F5F5F5; padding: 2px 6px; }
</style>

<div class="example-wrapper">
    <h1>{{ controller_name }}! ✅</h1>

    <form action="/ville" method="post">
        <fieldset>
            <legend>Formulaire</legend>
            <div class="form-group row">
                <label for="staticVille" class="col-sm-2 col-form-label">Nom</label>
                <div class="col-sm-10">
                    <input type="text" class="form-control-plaintext" id="ville"
                    name = "ville" >
                </div>
            </div>
            <button type="submit" class="btn btn-primary">Submit</button>
        </fieldset>
    </form>

</div>
{% endblock %}
```

Mémo Git

Mémo GIT


LOCAL
REMOTE
Working Directory
Staging Area
Repository
Repository
git add
git commit
git push
git reset
git pull

Mémo GIT

une fois votre repository créé et lié. Pour envoyer votre travail, vous devez effectuer **à la racine de votre répertoire** les commandes suivantes dans l'ordre :

1) git add .
2) git commit -m "message"
3) git push

une demande d'authentification vous sera demandée.

PHP

1) fichier .php
2) bloc d’instruction encadré par {....}
3) une instruction termine par ;
4) attention à l’indentation
5) besoin de commentaires

Gestion des erreurs Symfony

Symfony Exception | Symfony Docs | Symfony Support

ResourceNotFoundException > NotFoundHttpException | HTTP 404 Not Found

No route found for "GET /zeroiueuirhgiuoerh"

Exceptions 2 | Logs 1 | Stack Traces 2

Symfony\Component\HttpKernel\Exception\
NotFoundHttpException

in D:\Documents\demos\symfony-upload-multiple\vendor\symfony\http-kernel\EventListener**RouterListener.php** (line 136)

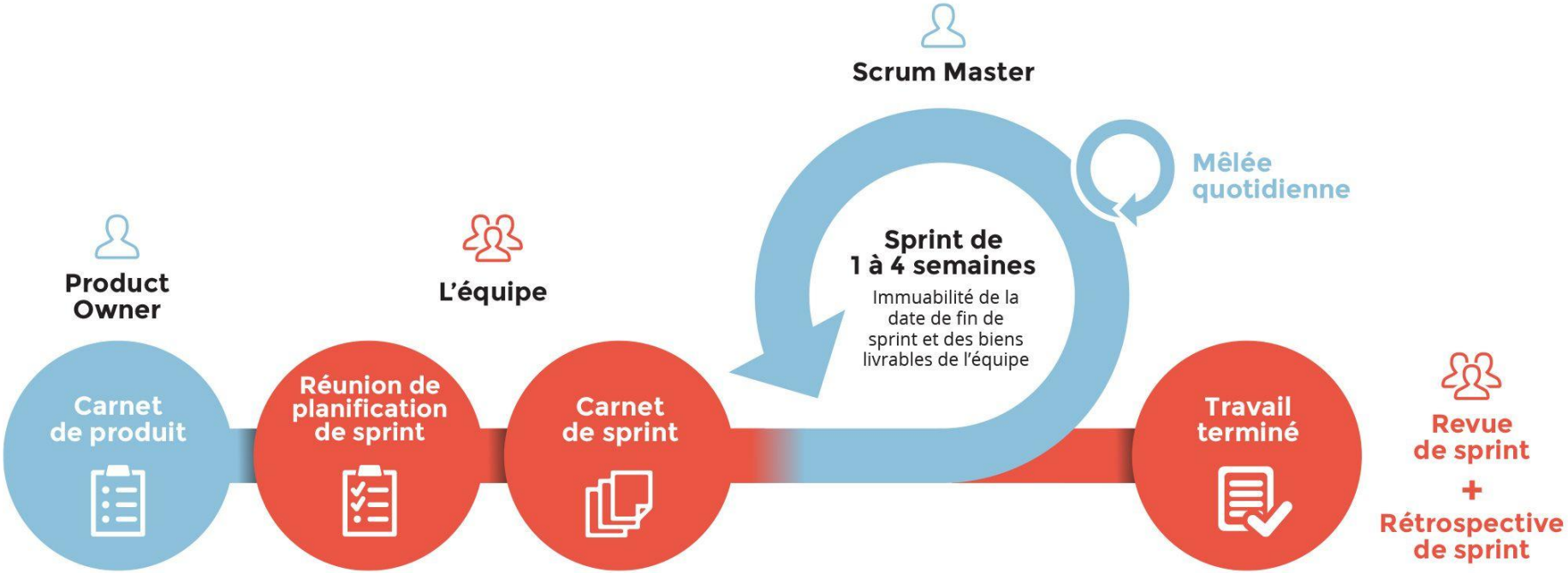
```
131.
132.             if ($referer = $request->headers->get('referer')) {
133.                 $message .= sprintf(' (from "%s")', $referer);
134.             }
135.
136.             throw new NotFoundHttpException($message, $e);
137.         } catch (MethodNotAllowedException $e) {
138.             $message = sprintf('No route found for "%s %s": Method Not Allowed (Allow: %s)', $request->getMethod(), $request->getPat
139.
140.             throw new MethodNotAllowedHttpException($e->getAllowedMethods(), $message, $e);
141.         }
```

RouterListener -> **onKernelRequest**(*object*(RequestEvent), 'kernel.request', *object*(TraceableEventDispatcher))
in D:\Documents\demos\symfony-upload-multiple\vendor\symfony\event-dispatcher\Debug**WrappedListener.php** (line 117)

WrappedListener -> **__invoke**(*object*(RequestEvent), 'kernel.request', *object*(TraceableEventDispatcher))

404 | 552 ms | 6.0 MB | 1 | 4 in 0.67 ms | n/a | 6 ms | Server | 5.0.7

Exemple de management de projet


Scrum Master
Mêlée quotidienne
Product Owner
L'équipe
Sprint de 1 à 4 semaines
Immuabilité de la date de fin de sprint et des biens livrables de l'équipe
Carnet de produit
Réunion de planification de sprint
Carnet de sprint
Travail terminé
Revue de sprint
+
Rétrospective de sprint

Préparatifs

1) Environnement technique avec php >8
2) bootswatch (choix du thème)
3) BD avec user et password
4) IDE paramétré (twig, php, …)
5) config > package > twig.yaml

```
form_themes: ['bootstrap_5_layout.html.twig']
```

Merci de votre attention