

DOSSIER PROFESSIONNEL (DP)

Nom de naissance > BOIRY
Nom d'usage > BOIRY
Prénom > Xavier
Adresse > 71 rue de Dunkerque
75009 PARIS

Titre professionnel visé

Développeur Web et Web Mobile

MODALITÉ D'ACCÈS :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.

Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

- 1 des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
- 2 du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
- 3 des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
- 4 de l'entretien final (dans le cadre de la session titre).

*[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels
du ministère chargé de l'Emploi]*

Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.

DOSSIER PROFESSIONNEL (DP)



<http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

| | | |
|--|-----------|-----------|
| Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité | p. | 6 |
| > Maquetter une application | p... | 6 |
| > Réaliser une interface utilisateur web statique et adaptable | p... | 10 |
| > Réaliser une interface utilisateur web dynamique | p... | 15 |
| Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité | p. | 18 |
| > Créer une base de données | p... | 18 |
| > Développer les composants d'accès aux données | p... | 21 |
| > Développer la partie back-end d'une application web ou web mobile | p... | 25 |
| Titres, diplômes, CQP, attestations de formation <i>(facultatif)</i> | p. | 30 |
| Déclaration sur l'honneur | p. | 31 |
| Documents illustrant la pratique professionnelle <i>(facultatif)</i> | p. | 32 |
| Annexes <i>(Si le RC le prévoit)</i> | p. | 33 |

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité- type



Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n°1 > *Maquetter une application*

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

A partir d'un cahier des charges sommaire, il nous a été demandé en exercice de réaliser les wireframes d'une application web fictive intitulée « o'four ».

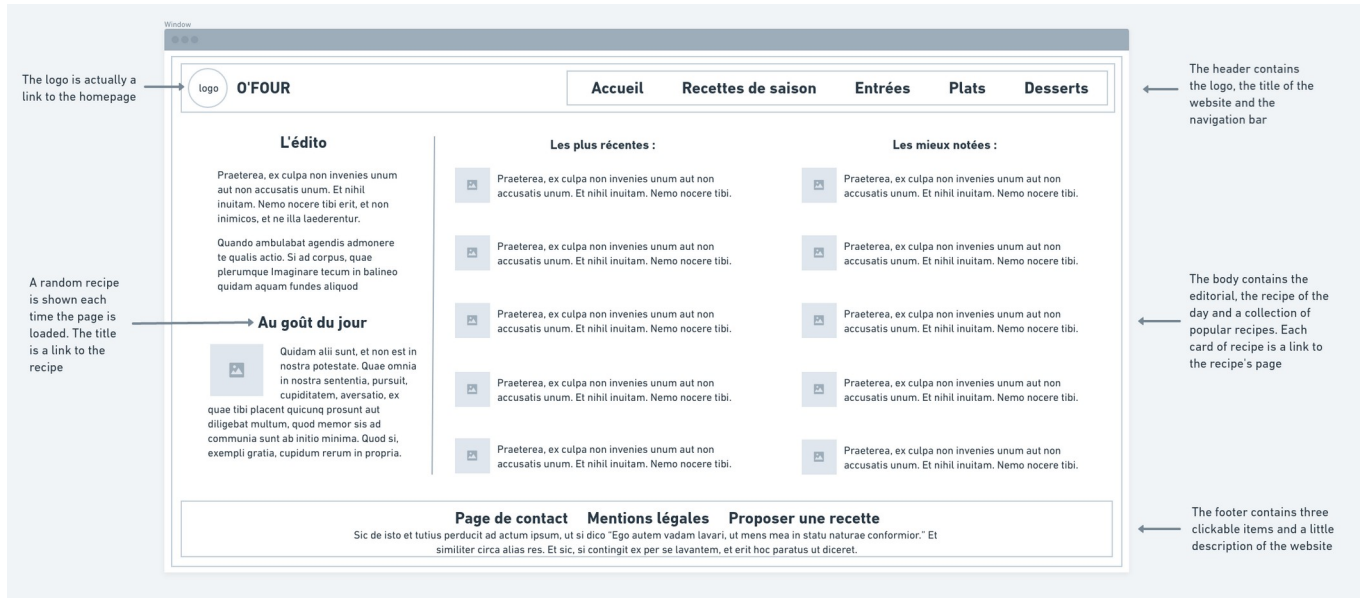
Les wireframes constituent une première étape importante du processus de création d'un site web car elles permettent de visualiser les éléments que contiendront les pages et ainsi de se projeter plus efficacement dans le projet et sa future réalisation.

Ici, il sera particulièrement intéressant de noter les différences entre la version desktop et la version mobile. La mise en évidence de ces différences à travers les wireframes peuvent éventuellement permettre de se poser les bonnes questions en amont du processus de création.

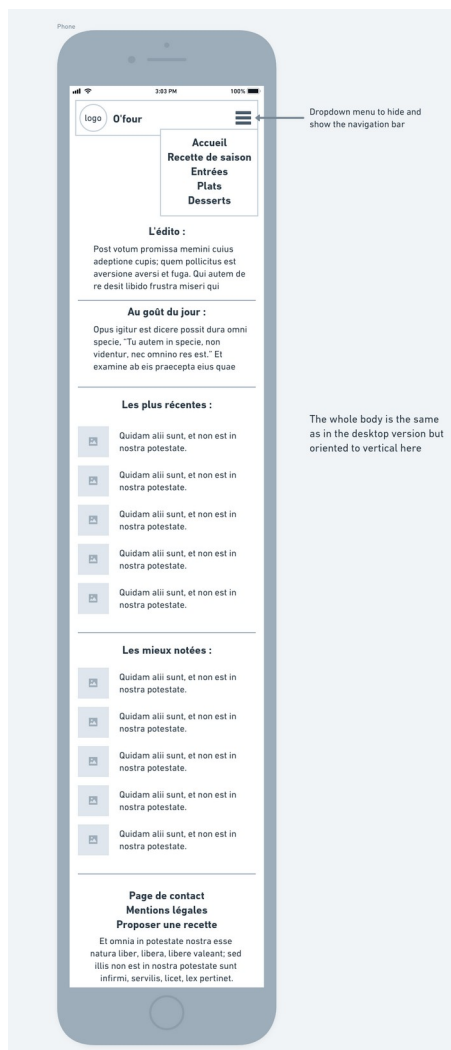
Un des problèmes que nous pouvons noter ici comme exemple sera la place prise par les textes pour chaque recette, éventuellement trop petite en version desktop et trop imposante en version mobile. Plusieurs solutions s'offrent alors à nous : adapter les textes en fonction du support, tronquer le texte en version mobile ou bien encore élaborer des textes équilibrés en imposant un nombre approximatif de caractères.

DOSSIER PROFESSIONNEL (DP)

Version desktop :



Version mobile :



2. Précisez les moyens utilisés :

J'ai utilisé l'outil de création de wireframes proposé par le site Whimsical.com.

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul.

4. Contexte

Nom de l'entreprise, organisme ou association > O'clock

Chantier, atelier, service > Exercice de formation

Période d'exercice > Du : 05/01/2021 au : 05/01/2021

5. Informations complémentaires (facultatif)

Activité- type



Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

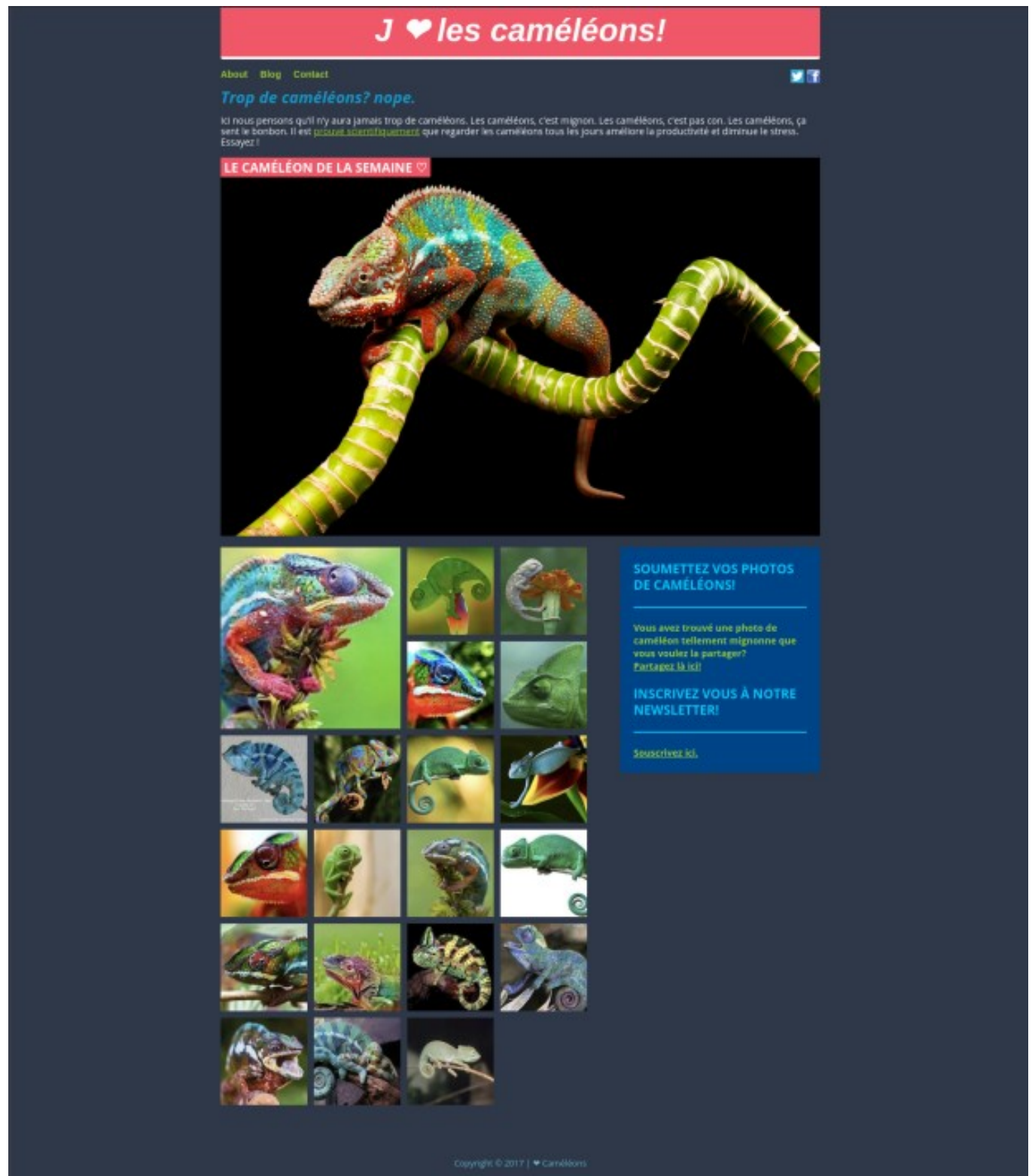
Exemple n° 2 > *Réaliser une interface utilisateur web statique et adaptable*

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Afin de répondre aux besoins d'une application responsive, il existe des librairies css telles que Bootstrap qui nous simplifient et nous garantissent le travail. Mais dans le cadre de la formation, il nous a fallu commencer sans avoir recours à de tels outils afin de nous familiariser avec les bonnes pratiques.

Dans cette optique, un exercice a été proposé autour d'un projet de site web statique. Il s'agissait de réaliser l'intégration de la page d'accueil d'une application fictive sur la thématique des caméléons.

Des difficultés de positionnement d'éléments étaient à surmonter et ce fut pour moi l'occasion de m'exercer aux flex-boxes. Concernant l'aspect responsive, ce fut un exercice parfait pour apprendre à utiliser les breakpoints et les media queries.



2. Précisez les moyens utilisés :

Au-delà de la construction de la page et de ses différents éléments en html, l'intérêt de l'exercice se situe surtout dans l'élaboration du css.

Etudions ici deux difficultés rencontrées sur cet exercice :

1. La barre de navigation divisée en deux parties (à gauche avec les liens internes et à droite avec les liens vers les réseaux sociaux).
2. Les photos des caméléons dont la première est deux fois plus large que les suivantes.

Pour le premier problème, j'ai opté pour la propriété inline-block qui permet d'aligner horizontalement les éléments de la barre de navigation et la positionnement absolu pour coller les icônes des réseaux sociaux sur la droite.

```
.main__nav {
  display: inline-block;
  font-family: sans-serif;
  color: yellowgreen;
  font-weight: bolder;
}

.main__nav ul li {
  display: inline-block;
  padding: 15px 15px 15px 0;
}

.main__reseaux {
  position: absolute;
  right: 0;
  padding: 15px 0;
}

.main__reseaux img {
  width: 20px;
}
```

DOSSIER PROFESSIONNEL (DP)

Pour le deuxième problème concernant les images de caméléons en grille, j'ai utilisé la propriété grid des flex-boxes en adaptant la taille de la première image.

```
.photos__wrapper {
  display: flex;
}

.wrapper__galerie {
  width: 600px;
  display: grid;
  grid-template-columns: 130px 130px 130px 130px;
  grid-template-rows: auto;
  grid-gap: 10px;
}

.wrapper__galerie > * {
  width: 130px;
}

.c01 {
  width: 270px;
  grid-column: 1 / 3;
  grid-row: 1 / 3;
}
```

Enfin, afin de traiter la partie responsive du site, j'ai ajouté une media query sur un breakpoint. Ainsi, la grille passe de quatre caméléons par ligne à trois lorsque la taille de l'écran passe en dessous de 600 pixels.

```
@media (max-width: 600px) {
  .photos__wrapper {
    flex-wrap: wrap;
  }

  .wrapper__galerie {
    width: 100%;
    grid-template-columns: 130px 130px 130px;
  }
}
```

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul.

4. Contexte

Nom de l'entreprise, organisme ou association > O'clock

Chantier, atelier, service > Exercice de formation

Période d'exercice > Du : 21/07/2020 au : 21/07/2020

5. Informations complémentaires (facultatif)

Activité- type



Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

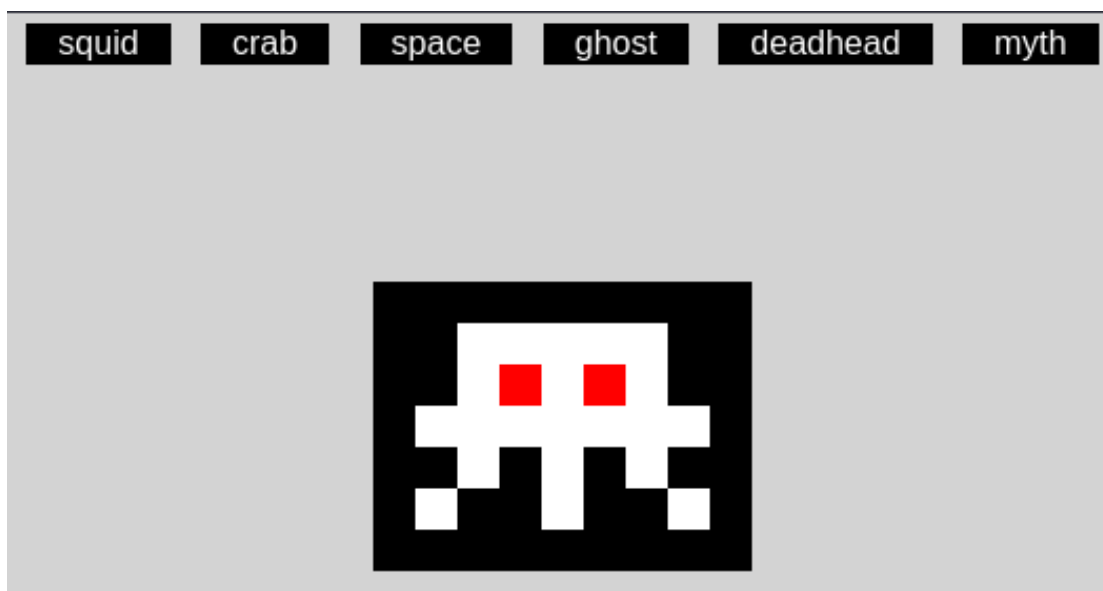
Exemple n° 3 > **Réaliser une interface utilisateur web dynamique**

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans l'optique de nous familiariser avec les possibilités de manipulation dynamique du DOM avec JavaScript, nous avons travaillé sur un exercice faisant appel à la création et à la modification d'éléments par l'intermédiaire d'un script.

Reprenant l'univers du célèbre jeu *Space Invader*, cette petite application propose d'afficher des figures sous forme de sprites, tout comme le jeu d'origine. Des boutons permettent à l'utilisateur d'afficher la figure qu'il souhaite.

Ces sprites sont en réalité composés uniquement de div de couleurs générées lorsque l'utilisateur clique sur un bouton :



Les figures sont représentées sous forme de matrices dans un fichier JS et la grosse partie du travail a été de passer de cette représentation à l’affichage effectif. Pour ce faire il a fallu imbriquer plusieurs boucles.

```
generate: function(invader) {
  // On efface le contenu avant d'en générer un nouveau
  document.getElementById("invader").innerHTML="";

  // En fonction du bouton cliqué, on récupère la valeur
  // de la figure à afficher et on attribue à une variable
  // l'entrée correspondante de l'objet contenant les matrices
  let table;
  switch (invader) {
    case "squid":
      table = map.models.squid;
      break;
    case "crab":
      table = map.models.crab;
      break;
    case "space":
      table = map.models.space;
      break;
    case "ghost":
      table = map.models.ghost;
      break;
    case "deadhead":
      table = map.models.deadhead;
      break;
    case "myth":
      table = map.models.myth;
      break;
  }

  // On parcourt les tableaux afin d'en extraire les informations
  // et dans le même temps on génère les divs
  for (let value of table) {
    let newLine = document.createElement("div");
    newLine.setAttribute("class", "line");
    document.getElementById("invader").appendChild(newLine);
    for (let i=0; i<=value.length; i++) {
      let pixelValue = value.slice(i, i+1);
      for (let decoration in map.types) {
        if (pixelValue == decoration) {
          let newPixel = document.createElement("div");
          newPixel.setAttribute("class", map.types[decoration]);
          document.getElementById("invader").appendChild(newPixel);
        }
      }
    }
  }
}
```


DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Il s'agit d'un travail fait uniquement en html, javascript et css.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul.

4. Contexte

Nom de l'entreprise, organisme ou association > O'clock

Chantier, atelier, service > Exercice de formation

Période d'exercice > Du : 13/08/2020 au : 13/08/2020

5. Informations complémentaires (facultatif)

Activité- type



Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

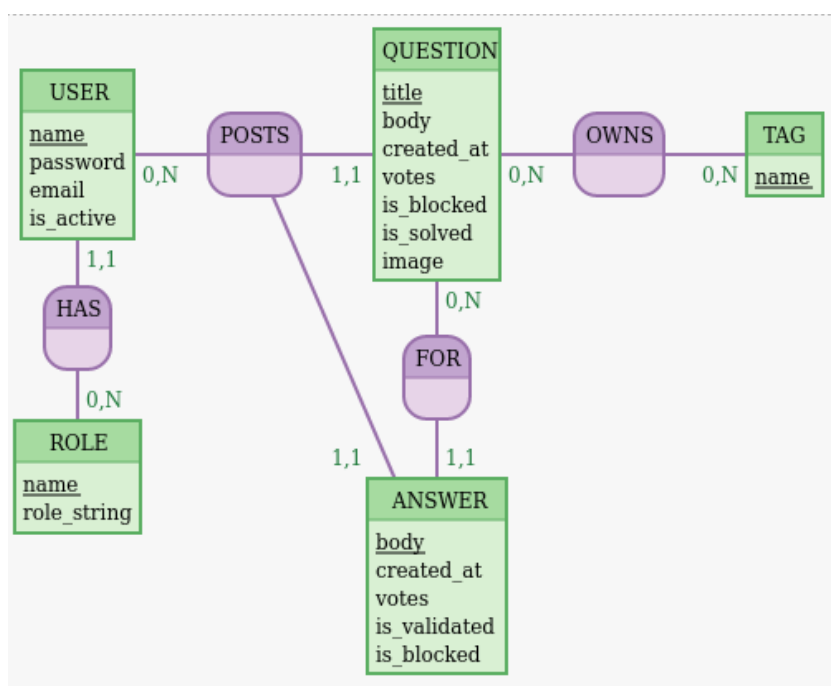
Exemple n° 1 > Créer une base de données

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour un exercice au sein de ma formation j'ai eu à développer une petite application s'inspirant de sites web tels que *Stack Overflow*. Il s'agit effectivement de lister une série de questions posées par des utilisateurs auxquels d'autres peuvent apporter leur réponse. Un système de vote permet ensuite d'apporter plus ou moins de poids aux réponses apportées. Enfin, un tag peut être attribué à une question afin de rendre la navigation plus aisée dans le site.

La première étape de ce projet a été de concevoir et de mettre en place la base de données. Pour ce faire, et suivant la méthode *Merise*, j'ai mis au point un MCD (Modèle Conceptuel de Données) et un MLD (Modèle Logique de Données), ceux-ci aidant à bien appréhender les entités du projet.

Voici le MCD :

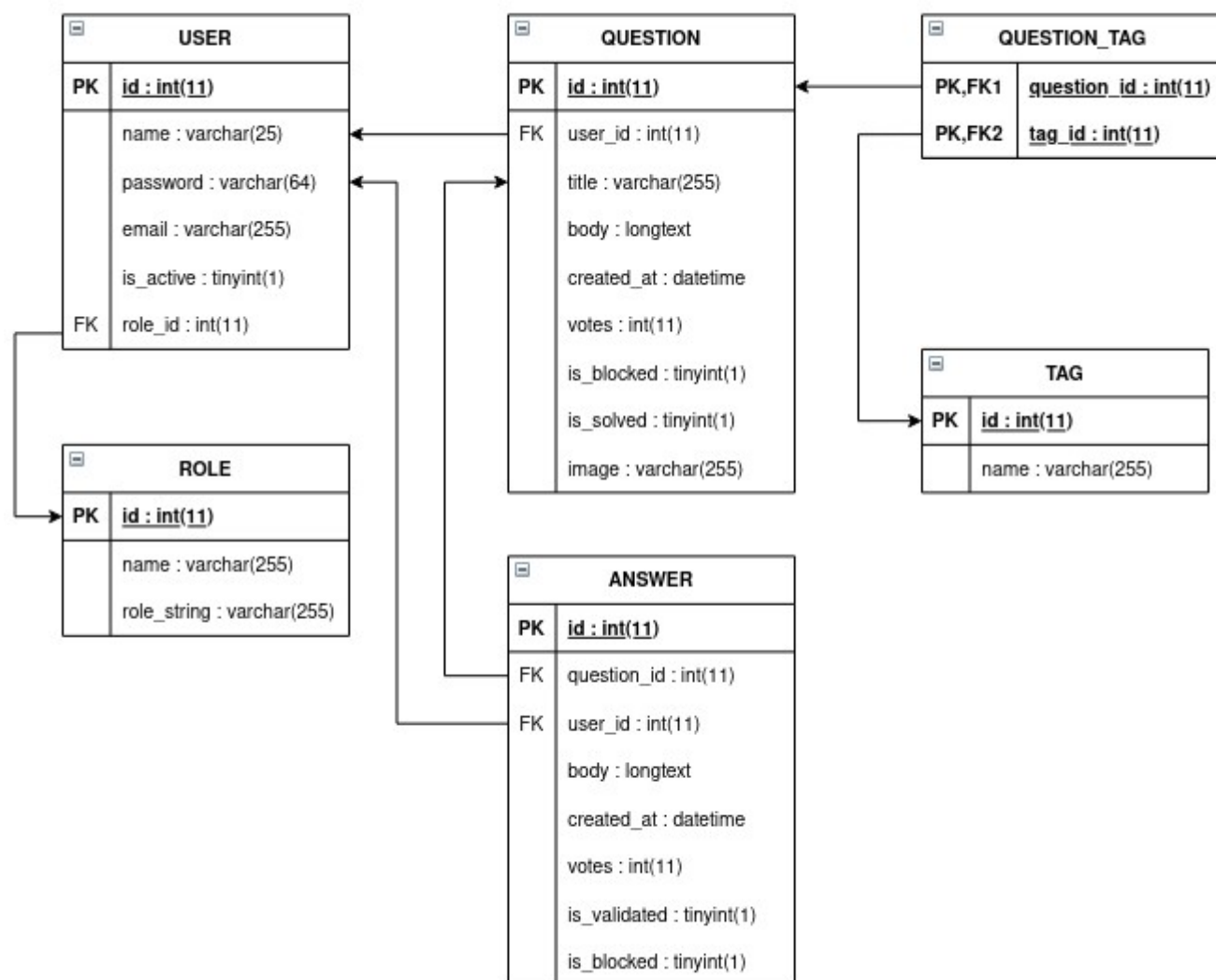


DOSSIER PROFESSIONNEL (DP)

Non seulement le MCD met en lumière les relations entre les différentes tables de la base de données mais il indique également la cardinalité de ces relations. Se poser ces questions et trouver leur réponse en amont de la réalisation de projet permet de ne pas s'y perdre au moment de la conception.

Par exemple ici nous pouvons constater que l'entité des questions est en relation de *OneToMany* avec l'entité des réponses alors que l'inverse est une relation de *OneToOne*. Lors de la conception de nos entités, ces informations nous seront bien utiles.

Ci-dessous, le MLD nous permet de visualiser les relations des entités en fonction de leurs clés étrangères :



DOSSIER PROFESSIONNEL (DP)

Travaillant avec le framework *Symfony* sur ce projet, il ne reste alors plus qu'à utiliser les makers pour mettre en place la base de données. Après avoir renseigné un nom pour notre bdd, il suffit d'appeler *Doctrine* (l'ORM de Symfony) en ligne de commande pour faire la création : **doctrine:database:create**. La commande **make:entity**, quant à elle, permet de lancer un petit wizard qui nous guide afin de créer les entités. Une fois ceci terminée, il nous reste encore à créer le fichier de migration en tapant **make:migration** pour enfin l'appliquer à notre base de données en tapant **doctrine:migrations:migrate**.

2. Précisez les moyens utilisés :

J'utilise le site web mocodo.wingi.net pour réaliser les MCD et le site draw.io pour réaliser les MLD.

La conception de la base de données a été réalisée ici à l'aide de Symfony et de Doctrine.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul.

4. Contexte

Nom de l'entreprise, organisme ou association > O'clock

Chantier, atelier, service > Exercice de formation

Période d'exercice > Du : 11/11/2020 au : 11/11/2020

5. Informations complémentaires (facultatif)

Activité- type



Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 2 > *Développer les composants d'accès aux données*

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour cet exemple, plutôt qu'utiliser Doctrine j'aimerais proposer une application php simple utilisant la classe PDO (PHP Data Objects). Dans un atelier réalisé durant la formation, j'ai en effet eu à communiquer avec la base de données uniquement avec les outils mis à disposition par PHP.

Il s'agissait de mettre en place une petite application à partir d'un MCD et d'une base de données fournie au format SQL. Le thème était Sonic, le célèbre hérisson vidéoludique. Sur la page d'accueil, ce site devait fournir un visuel des différents personnages de la série ainsi qu'une petite description. Chaque personnage disposait de sa page personnelle. En outre étaient présentés les créateurs du jeu auxquels avaient également leur page dédié.

Le challenge consistait donc en premier lieu à la réalisation de ce site suivant le modèle MVC (Modèle Vue Contrôleur) et ce sans framework. Ensuite, et c'est ce qui nous intéresse ici, il s'agissait de faire communiquer l'application avec la base de données.

2. Précisez les moyens utilisés :

J'ai d'abord conçu une classe spécifique qui crée une instance de PDO. Séparer cette partie du reste du code permet de ne pas se répéter et ainsi respecter la recommandation DRY (Don't Repeat Yourself).

Ensuite, j'ai codé une classe Character qui se charge de faire les requêtes SQL et d'appeler PDO afin de récupérer les données de la base.

En illustration, vous trouverez ci-jointes les classes mentionnées ci-dessus.

```

<?php

namespace Sonic\Utils;

use Exception;
use PDO;

class Database
{
    private $dbh;
    private static $_instance;

    private function __construct()
    {
        $config = parse_ini_file(__DIR__ . '/../config.ini');

        try {
            $this->dbh = new PDO(
                'mysql:host=' . $config['DB_HOST'] . ';dbname=' . $config['DB_NAME'] . ';charset=utf8',
                $config['DB_USERNAME'],
                $config['DB_PASSWORD'],
                [PDO::ATTR_ERRMODE => PDO::ERRMODE_WARNING]
            );
        } catch (Exception $exception) {
            echo 'Erreur de connexion...<br>';
            echo $exception->getMessage(). '<br>';
            echo '<pre>';
            echo $exception->getTraceAsString();
            echo '</pre>';
            exit;
        }
    }

    public static function getPDO()
    {
        if (empty(self::$_instance)) {
            self::$_instance = new Database();
        }
        return self::$_instance->dbh;
    }
}

```

```
<?php

namespace Sonic\Models;

use Sonic\Utils\Database;
use PDO;

class Character
{
    private $id;
    private $name;
    private $description;
    private $picture;
    private $type_name;

    /**
     * La méthode find() permet d'obtenir les informations dans la BDD pour un personnage unique
     * Elle prend en paramètre l'ID du personnage et retourne l'objet Character
     */
    public function find($id)
    {
        $sql = '
            SELECT *
            FROM `character`
            WHERE id = ' . $id
        ;
        $pdo = Database::getPDO();
        $pdoStatement = $pdo->query($sql);
        $result = $pdoStatement->fetchObject(Character::class);

        return $result;
    }
    ...
}
```

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul.

4. Contexte

Nom de l'entreprise, organisme ou association > O'clock

Chantier, atelier, service > Exercice de formation

Période d'exercice > Du : 09/09/2020 au : 09/09/2020

5. Informations complémentaires (facultatif)

Activité- type



Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 3 > *Développer la partie back-end d'une application web ou web mobile*

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Reprenons l'exemple de l'application de questions / réponses vue précédemment. Une fois la base de données modélisée et créée, il ne restait plus qu'à développer les fonctionnalités dignes d'un petit site web. C'est ce que je vous propose de voir à présent.

L'application, développée à l'aide du framework Symfony, devait être capable en premier lieu de gérer l'inscription ou la connexion d'un utilisateur. Pour ce faire, deux pages de formulaires codées en twig s'occupent de la partie front. Côté back, il a fallu développer deux classes intégrant les contraintes de validation :

```
class RegisterType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('username', null, [
                'constraints' => new NotBlank,
            ])
            ->add('password', PasswordType::class, [
                'constraints' => new NotBlank,
            ])
            ->add('email', EmailType::class, [
                'constraints' => [
                    new Email,
                    new NotBlank,
                ],
            ])
        ;
    }
    ...
}
```

```
class LoginType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('_username')
            ->add('_password', PasswordType::class)
        ;
    }
    ...
}
```

Une fois cette partie formulaire faite, je me suis attaqué aux contrôleurs qui transmettent les informations entrées par l'utilisateur. Penchons-nous plus précisément sur le code gérant l'inscription. Le contrôleur transmet donc les informations et à la condition que celles-ci soient valides, on effectue les traitements suivants :

- On encode le mot de passe (ici il s'agira d'un encodage bcrypt comme indiqué dans le fichier security.yaml). En effet, par souci de sécurité, aucun mot de passe ne doit être entré en dur dans une base de données.
- On assigne au nouvel utilisateur le rôle par défaut (ROLE_USER) qui permet d'avoir accès au minimum de l'application.
- On peut alors persister les informations en base de données grâce à l'entity manager de Doctrine.
- On affiche un message flash pour indiquer à l'utilisateur que l'opération a réussi.
- Pour finir, on redirige vers la page de connexion afin que l'utilisateur entre ses nouveaux identifiants.

```
class UserController extends AbstractController
{
    /**
     * @Route("/user/register", name="user_register")
     */
    public function register(Request $request, UserPasswordEncoderInterface $encoder, RoleRepository $roleRepository)
    {
        $user = new User();
        $form = $this->createForm(RegisterType::class, $user);
        $form->handleRequest($request);
    }
}
```

DOSSIER PROFESSIONNEL (DP)

```
if ($form->isSubmitted() && $form->isValid()) {
    $user->setPassword($encoder->encodePassword($user, $user->getPassword()));
    $role = $roleRepository->findOneByRoleString('ROLE_USER');
    $user->setRole($role);

    $entityManager = $this->getDoctrine()->getManager();
    $entityManager->persist($user);
    $entityManager->flush();

    $this->addFlash('success', 'Vous êtes enregistré. Vous pouvez maintenant vous connecter.');
```

return \$this->redirectToRoute('login');

```
}

return $this->render('user/register.html.twig', [
    'form' => $form->createView(),
]);
}

...
```

Côté sécurité, l'accès aux options de modération est limité en fonction du rôle et grâce au fichier security.yaml :

```
access_control:
- { path: ^/admin/(question|answer)/toggle, roles: ROLE_MODERATOR }
- { path: ^/admin/(tag|user), roles: ROLE_MODERATOR }
- { path: ^/answer/validate, roles: ROLE_USER }
- { path: ^/question/add, roles: ROLE_USER }
- { path: ^/question/\d+, roles: ROLE_USER, methods: ["POST"] }
- { path: ^/user/(profile|edit), roles: ROLE_USER }
```

```
role_hierarchy:
    ROLE_ADMIN: ROLE_MODERATOR
    ROLE_MODERATOR: ROLE_USER
```

Pour la suite de l'exercice il restait encore à compléter le CRUD (Create Read Update Delete) pour chaque entité. Il est toujours difficile suivre à la lettre la conception d'un CRUD et on préférera même parfois le BREAD (Browse Read Edit Add Delete). La raison de tel ou tel choix se fera en fonction du contexte. Ici je prendrai l'exemple de l'entité tag car elle reflète bien la nécessité de se rapprocher d'une organisation en BREAD. En effet, la page d'accueil recense tous les tags (Browse) et on peut également afficher un tag (Read) pour l'éditer (Edit). Il est en outre possible d'en ajouter (Add) et d'en effacer (Delete).

DOSSIER PROFESSIONNEL (DP)

```
<?php

namespace App\Controller;

use App\Entity\Tag;
use App\Form\TagType;
use App\Repository\TagRepository;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

/**
 * @Route("/admin/tag")
 */
class TagController extends AbstractController
{
    /**
     * @Route("/", name="tag_index", methods="GET")
     */
    public function index(TagRepository $tagRepository): Response
    {
        return $this->render('tag/index.html.twig', ['tags' => $tagRepository->findAll()]);
    }

    /**
     * @Route("/new", name="tag_new", methods="GET|POST")
     */
    public function new(Request $request): Response
    {
        $tag = new Tag();
        $form = $this->createForm(TagType::class, $tag);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            $em = $this->getDoctrine()->getManager();
            $em->persist($tag);
            $em->flush();

            $this->addFlash('success', 'Tag ajouté.');
```

```
            return $this->redirectToRoute('tag_index');
        }

        return $this->render('tag/new.html.twig', [
            'tag' => $tag,
            'form' => $form->createView(),
        ]);
    }

    /**
     * @Route("/{id}", name="tag_show", methods="GET")
     */
    public function show(Tag $tag): Response
    {
        return $this->render('tag/show.html.twig', ['tag' => $tag]);
    }

    /**
     * @Route("/{id}/edit", name="tag_edit", methods="GET|POST")
     */
    public function edit(Request $request, Tag $tag): Response
    {
        $form = $this->createForm(TagType::class, $tag);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            $this->getDoctrine()->getManager()->flush();

            $this->addFlash('success', 'Tag modifié.');
```

```
            return $this->redirectToRoute('tag_edit', ['id' => $tag->getId()]);
        }

        return $this->render('tag/edit.html.twig', [
            'tag' => $tag,
            'form' => $form->createView(),
        ]);
    }

    /**
     * @Route("/{id}", name="tag_delete", methods="DELETE")
     */
    public function delete(Request $request, Tag $tag): Response
    {
        if ($this->isCsrfTokenValid('delete', $tag->getId(), $request->request->get('_token'))) {
            $em = $this->getDoctrine()->getManager();
            $em->remove($tag);
            $em->flush();

            $this->addFlash('success', 'Tag supprimé.');
```

```
        }

        return $this->redirectToRoute('tag_index');
    }
}
```

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Cette application est entièrement réalisée à l'aide de Symfony, de son ORM Doctrine et de son langage de templates twig.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul.

4. Contexte

Nom de l'entreprise, organisme ou association > O'clock

Chantier, atelier, service > Exercice de formation

Période d'exercice > Du : 11/11/2020 au : 25/11/2020

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

| Intitulé | Autorité ou organisme | Date |
|-------------------------------|-----------------------|------------|
| Développeur web et web mobile | O'clock | 21/12/2020 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

DOSSIER PROFESSIONNEL (DP)

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] **Xavier Boiry**,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont
exacts et que je suis l'auteur(e) des réalisations jointes.

Fait à **Paris** le **26/01/2021**
pour faire valoir ce que de droit.

Signature :



Documents illustrant la pratique professionnelle

(facultatif)

| Intitulé |
|----------|
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)