

Dossier de compétences

Cyril VASSALLO
Full-Stack Développeur

Développeur web PHP/ JavaScript / MySQL
Concepteur Développeur d'Application

Niveau d'expériences au 11/05/2020 :

- 3 mois en milieu pro
- 2 ans de pratique du code
- 15 mois de pratique PHP, Javascript, MySQL et en intégration HTML5/CSS3
- 8 mois de pratique sur Symfony 4/5
- 5 mois de pratique React ES6, node.js express, webpack

Domaine de Compétences...

Compétences fonctionnelles :

- Conception, développement, intégration, tests, recette, maintenance applicative, rédaction documentation, méthodologie Agile, maquettage, conception BDD, architecture MVC, API RESTful.
- Métiers, secteurs d'activité : Développement web / Informatique / Numérique

Compétences de formation:

1. Maquetter une application
2. Développer une interface utilisateur de type desktop
3. Développer des composants d'accès aux données
4. Développer la partie front-end d'une interface utilisateur web
5. Développer la partie back-end d'une interface utilisateur web
6. Concevoir une base de données
7. Mettre en place une base de données
8. Développer des composants dans le langage d'une base de données
9. Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement
10. Concevoir une application
11. Développer des composants métier
12. Construire une application organisée en couches.
13. Préparer et exécuter les plans de tests d'une application.
14. Déploiement et mise en service d'une application.
15. Documenter, archiver et versionner le code.

Langues : Anglais, Mandarin

Connaissances Techniques

Langages : PHP, JavaScript, MySQL, HTML5/CSS3, SASS, Json, XML

Autres langages: des bases en Java + POO, notion et intérêt pour GraphQL

Bases de données : MySQL, méthode Merise, Uml MPD MCD, script de création SQL, triggers, procédures stockées.

Logiciels : Visual Studio Code, Postman, Photoshop, PhpMyadmin, MySQL Workbench, Eclipse, NetBeans,

Frameworks : Symfony 4/5, React, Api-platform, PHPUnit, Vue.js, Bootstrap 4 encore webpack, node.js(express).

Outils : Trello, Redmine, Jira, Serveur Apache, Git, MS Office

Méthodologie : Agile, cahiers des charges , Gantt

Certifications :

Titre Concepteur Développeur d'Application,
BTS électronique
BAC STI électronique

Systèmes : informatique et composant embarquée, installation, tests, maintenance et support

Réseaux : mise en réseau de périphériques (copieur, machines industrielles, etc...)

Expériences Professionnelles

Emity Du 16/09/2019 au 06/12/2019

Contexte

- Gestion de l’affichage d’informations dynamiques sur écran HD en milieu public ou privé à partir d’une plateforme web en PHP / Javascript. (restaurants, ascenseur, galeries marchandes, totems, résidences de copropriétaires...).

Mission

- Dans une équipe de 5 développeurs, j’ai participé à la conception, le développement, l’amélioration et l’intégration de modules à la plateforme web PHP/Javascript existante.

Projets

- En charge du projet de refonte du module Administration de la plateforme web EManager indispensable pour les grands comptes comme **Europcar**, **Schindler** et **Prosim** pour la gestion des droits des utilisateurs.
 - Recherche et maquettage des besoins et fonctionnalités du module en collaboration avec le directeur technique.
 - Amélioration et utilisation de la base de données existante puis création des composants SQL.
 - Développement front-end en s’appuyant sur des technologies comme JQuery, l’AJAX, Bootstrap 4, Twig 3 , Datatable.
 - Développement back-end en s’appuyant sur une structure MVC personnalisée et en utilisant le langage MySQL pour les accès aux données persistantes.

- Travail en équipe sur le développement, l'amélioration et l'intégration des modules manquants ou existants de la plateforme Web.
 - Intervention sur du code existant et/ou en développement par d'autres développeurs notamment sur les modules :
 - Éditer une affiche (Javascript)
 - Widget Flux RSS (XML)
 - choix des Templates de base pour les nouveaux adhérents de la plateforme web EManager.
 - Conception et mise en place d'une base de données pour le stockage persistant des données issues de l'API Navitia.io
 - Création , reprise et/ou intégration de modules en s'appuyant sur les API's open sources comme OpenWeatherMap, Navitia.io et une base de données MySQL.
 - widget météo (JSON)
 - widget Flux RSS (XML)
 - Création d'une API personnalisée " Info-Transport" revoyant des flux JSON en public
 - Pratique du pair-programming pour assister et donner des clefs de déblocages aux développeurs moins expérimentés.
 - Travail en méthode agile avec l'outil Trello et mise en place du versionning git par branche en ligne de commande Linux.

Environnement technique : Mac OS, Windows , PHP, Javascript/Jquery(AJAX), Twig, MySQL, Apache (Laragon, Wamp), Chrome, Firefox, Safari

Formation CDA

Du 14/01/2019 au 16/09/2019

Contexte

- Projet « Fil rouge - RPS » « Gestionnaire de questionnaires d'enquête en ligne sur les **Risques PsychoSociaux** »

Mission

- Dans une équipe de 3 développeurs, j'ai participé au maquettage, à la conception et au développement du gestionnaire de questionnaires **RPS**.

Projet

- En charge de concevoir et développer pour la cliente Raymonde Ricourt secrétaire au CHSCT de L'AFPA un outil de sondage permettant de lancer une étude chaque année auprès de collaborateurs sur le risques sociaux psychologique au travail.
 - Application Web avec un front-end Bootstrap, Javascript/Jquery et un back-end avec PHP sur serveur apache2 et mysql-server :
 - Un utilisateur non connecté peut répondre de façon anonyme à un questionnaire
 - Un utilisateur connecté peut être de 2 types : administrateur ou utilisateur
 - Un administrateur connecté peut supprimer un questionnaire, afficher la liste complète des utilisateurs ainsi que modifier/supprimer ces derniers et possède toute les fonctionnalités d'un utilisateur.
 - Un utilisateur connecté peut modifier, éditer et soumettre un nouveau questionnaire en ligne et consulter les statistiques des réponses anonymes.
 - Maquettage:
 - mockup
 - photoshop
 - Modélisation UML :
 - Use case
 - diagramme des classes
 - diagramme des Transitions
 - Merise :

- analyse des données brut
- dictionnaire de données
- règles de gestion métier
- MCD,MPD
- Gestion de Projet:
 - Gantt
 - trello
 - github
- Réalisation du script de création MySQL
- Mise en place de la base de données.
- Architecture back-end custom de type N-tiers:
 - Mise en place d'un design pattern factory pour réaliser la connexion et déconnexion des canaux d'accès avec la BDD MySQL.
 - Réalisation d'un routeur permettant l'instanciation du contrôleur demandé par la requête HTTP et la gestion des rôles utilisateur.
 - Réalisation de la couche service pour les calculs et les échanges avec la base de données.
 - Réalisation de la couche contrôleur réalisant les échanges de données entre les ressources persistantes, les services et le front-end
 - Echange de données client/serveur par le biais de requêtes HTTP (GET et POST) et AJAX avec des réponses au format Content-Type Application/JSON
- Architecture front-end:
 - Bootstrap
 - Javascript/JQuery
 - Datatable.js
 - Charts.js
- Sécurité
 - réécriture des urls par htaccess.
 - mise en place de l'api captcha pour lutter contre les bots.
 - cryptage des mots de passes.
 - validation de compte par mail.
 - utilisation de la POO et des bonnes pratiques de portée des variables et méthodes entre les couches.
 - protection contre les injections SQL et failles XSS grâce aux composants natifs à PHP et du PDO
 - contrôle des champs et validation des données HTTP en Back-end.

- Utilisation du framework PHPUnit pour la réalisation des tests unitaires

Projets Personnels

2020

Contexte : Dans le cadre d'un apprentissage de l'architecture API RESTful. Réalisation d'une API type CRM (client, facture, utilisateurs...) associé à une SPA réalisée React. Le projet illustre mon apprentissage du framework Symfony avec Api-platform pour le back-end et React pour le front-end. La gestion de l'authentification est faite en Stateless avec grâce aux Json Web Tokens générés par le bundle lexik/JWT de symfony 4.

Environnement technique : Linux, Apache2, mysql-server, Symfony 4, React, API-platform, Doctrine, Bootstrap 4, Webpack Encore.

Repository git: <https://github.com/zhangsaili/api-course>

--

Contexte : Dans le cadre d'un apprentissage du Framework Symfony 4. Réalisation d'un site web factice de réservation d'appartements. Permettant à des utilisateurs de s'inscrire, de se connecter, de diffuser des annonces consultables par le public anonyme et de réaliser des réservations.

Environnement technique : Linux, Apache2, mysql-server, Symfony 4, Doctrine, Twig, Bootstrap 4

Repository git: <https://github.com/zhangsaili/Symfony-Booking>

2019

Contexte : Réalisation d'un site web portfolio illustrant ma pratique de symfony 4.3 et les technologies comme datatable, Bootstrap 4, Twig, JQuery CSS3 et une base de données MySQL de 10 entités.

Evolution en cours du site (voir repository git) en API RESTful avec React en Front-end et une base de données de 5 ressources.

illustrant la pratique de Symfony 4, React, SASS, le framework Api-Platform et les JWT pour l'authentification.

<https://www.infinity-dev.fr>

Environnement technique : Linux, Apache2, mysql-server, Symfony 4, React, SASS, API-platform, Doctrine, Bootstrap 4, Webpack Encore.

Repository git: <https://github.com/zhangsaili/infinity-dev>

--

Contexte: Conception et développement d'un projet de gestionnaire de questionnaires pour les Risques Psychologiques et Sociaux pour le CHSCT de l'Afpa à l'échelle nationale. Projet PHP, Javascript JQuery, Bootstrap 4. projet en production et réalisé par une équipe de 5 développeurs - Versionning git.

Environnement technique : Linux, Windows, PHP, Javascript/Jquery, HTML5/CSS3, Bootstrap 4, Apache, Chrome, Firefox, Safari

Repository git: <https://github.com/zhangsaili/rps-perso>

—

Contexte : Amélioration d'une structure n-tiers en PHP "**ultra light**" en Orienté objet et respectant les normes de sécurité entre les couches. Il est construit avec Twig en front-end et contient les couches Métiers, Services, Accès aux données pour le back-end. Il se connecte à une base de donnée au choix comme MySQL, PostgreSQL. Il est Prêt à l'emploi et idéal pour travailler sans framework.

Evolution possible avec React en SPA en API-RESTful en flux json

Environnement technique : Linux, Windows , PHP , Javascript/Jquery , MySQL, HTML5/CSS3, Bootstrap 4, Twig Apache, Chrome, Firefox, Safari

Repository git: <https://github.com/zhangsaili/Framework-PHP-Twig3>

2018

Contexte : Réalisation d'une application du jeu le Juste Prix en JAVA et POO de type application jar. Doté d'une interface développée à l'aide de Swing permettant aux utilisateurs de choisir la langue, le nombre de joueurs, le niveau de difficulté.

Environnement technique : Window, JAVA, framework Swing, NetBeans, Eclipse

Repository git:

<https://github.com/zhangsaili/Java-Game-Find-The-Price>

Maîtrise des Framework

React

le 11/05/2020

Mise en application sur : api-course, infinity-dev

A titre d'apprentissage, élaboration d'une mini CRM en SPA avec React interagissant avec une api symfony 4 utilisant api-platform et les JWT permettant de mettre en oeuvre:

- javascript avec la syntaxe ES6
- l'import, export de modules
- transpiler un code ES6 avec babel compatible avec tout navigateur grâce à webpack
- réaliser des full composants avec la gestion des states
- réaliser des composant descriptifs
- réaliser une arborescence de composants interdépendant
- utiliser Axios pour le traitement des requêtes HTTP
- normaliser, dénormaliser, sérialiser et désérialiser
- réaliser des promise ou des callback sur des events personnalisés ou natif
- gérer les tokens utilisateur

Symfony 4.3

le 11/05/2020

Mise en application sur : Symfony-Booking, infinity-dev

A titre d'apprentissage, élaboration de sites webs pour mettre en oeuvre:

- l'utilisation des contrôleurs avec paramétrage et sécurisation des routes.
- l'affichage des pages avec Twig
- le responsive sur tous les supports avec bootstrap 4
- la création et l'automatisation de fausses données pour l'environnement de développement
- la création de formulaires sécurisés et leurs utilisation en back-end
- la soumission et l'enregistrement de données de formulaire

- le contrôle des champs de formulaire et messages d'erreurs
- la récupération et l'enregistrement en base de données avec Doctrine
- la connexion des utilisateurs
- la gestion des Rôles
- la sécurisation et restriction des ressources
- l'édition des pages d'erreur 403 et 404
- le déploiement sur un serveur mutualisé sans ssh en utilisant le .htaccess pour les réécriture d'url.

En détails:

CLI:

- Lancer un server local
- Créer un Contrôleur
- Créer une base de données
- Faire les migrations de base de données
- Créer et mettre à jour une entité, des propriétés et des relations
- Créer une classe formulaire
- Création d'une classe fixtures
- Installer des dépendances avec composer
- Installer des dépendances avec npm
- Versionner avec git
- lancer les testes unitaires avec PHPUnit

Config:

- .env - base de données et switch environnement dev/prod
- security.yaml - initialiser un encoder et des providers et des firewall
- route.yaml - ajouter des routes
- service.yaml - ajouter un service
- twig.yaml - initialiser un thème général bootstrap 4
- webpack.config.js - configurer installer react avec encore webpack
- configurer api-platform et sécuriser l'accès à l'api
- configurer le bundle lexik/jwt

Controller:

- Appelle des méthodes des contrôleurs de Symfony avec la classe **AbstractController**
- Injection de dépendances
- Routes paramétrées avec le composant **Route**
- Rendu de la vues et passage de donnée Twig avec la méthode **render()**
- Création et rendu de formulaires avec les méthodes **createForm()** et **createView()**
- Sécurisation des routes avec le composant **Security** et **IsGranted**
- Gérer l'authentification avec le composant **AuthenticationUtils**
- Obtenir des données de requête avec le composant **Request**

- Gérer les erreurs de formulaire avec le composant **FormError**
- Crypter les passwords avec le composant **UserPasswordEncoderInterface**
- Gérer les messages front-end avec la méthode **addFlash()**
- Mettre à jour une entité avec le composant **EntityManagerInterface**
- Faire persister les données avec les méthodes **persiste()** et **flush()**

Repository:

- Ajouter des requêtes personnalisées avec la méthode **createQueryBuilder()** du composant **ServiceEntityRepository** de Doctrine

Entity:

- Créer une entité ORM avec le composant **Mapping**
- Créer une classe modèle sans relation avec la base de données.
- Eviter les doublons avec le composant **UniqueEntity**
- Exécuter des méthodes d'entité personnalisées avant et après l'insertion des données avec le composant **ORM\HasLifecycleCallbacks** (genre de trigger d'ORM)
- Paramétrage des rôles avec l'implémentation du composant **UserInterface** sur une entité User
- Mise en place des contraintes d'insertion avec le composant **Constraints**
- utilisation du composant **ApiResponse** de api-platform
- utilisation du composant **Groups** pour le filtrage des données à découvrir

Form:

- Ajout de champ avec le composant **FormBuilderInterface**
- Typage des champs grâce aux classes **TextType**, **EmailType**, **PasswordType** etc ..)
- Ajout d'options

Twig:

- Construire le template de base
- Utilisation des **blocks** dans les templates enfants
- Factorisation de template en utilisant **include**
- Inclusion de route paramétrées avec la méthode **path()** de Twig
- Utilisation conditions et itérations dans les templates Twig
- Utilisation des balises formulaire + thème personnalisé avec **widget**, **row**, **label** et **alert**
- Utilisation des filtres | Twig
- Utilisation des variables dans les templates Twig
- Utilisation des variables globale dans l'objet Twig app

Fixtures:

- Création de fausses données avec itération et enregistrement en base de données de développement
- Utilisation de la dépendance **faker**
- Utilisation des api's comme **lorempixel**, **placeholder.it** et **randomuser.me**

Formations

2020 : 196h– Formation Javascript-Full-stack, React – DIGINAMIC
Montpellier
certification opquast obtenu

2019 : Bac + 3 – Concepteur Développeur d'Application – AFPA
St-Jean-de-védas
Diplôme obtenu

2005 : Bac + 2 – BTS ELECTRONIQUE – Lycée Dhuoda
Nîmes
Diplôme obtenu

2003 : Bac – STI ELECTRONIQUE – Lycée Dhuoda
Nîmes
Diplôme obtenu