**M2i Formation**

2024

**Dossier Professionnel**

**Concepteur Développeur d’Applications**

**Ruben Trottein**

Table des matières

Concepteur Développeur d’Applications

Contenu

[1. Introduction 2](#_Toc183512119)

[2. Description des projets réalisés 3](#_Toc183512120)

[A. Projets CDA 3](#_Toc183512121)

[APIs 3](#_Toc183512122)

[Node.js noSql MongoDB (API / NoSql); 3](#_Toc183512123)

[Applications 4](#_Toc183512124)

[Symfony 4](#_Toc183512125)

[Php 4](#_Toc183512126)

[Java Spring (Web): 4](#_Toc183512127)

[React 4](#_Toc183512128)

[NodeJS / Symfony (BDDs Relationnelles) : 4](#_Toc183512129)

[JavaScript (Full Front-end) 4](#_Toc183512130)

[Java (Console) 5](#_Toc183512131)

[Kotlin (Mobile) 5](#_Toc183512132)

[Ionic (Mobile) 5](#_Toc183512133)

[B. Projets perso 6](#_Toc183512134)

[3. Développement des compétences 7](#_Toc183512135)

[4. Résultats et bilan 8](#_Toc183512136)

[5. Annexes 9](#_Toc183512137)

# Introduction

*Présentez-vous : parcours, expériences, motivations pour devenir CDA.*



# Description des projets réalisés

*Chaque projet doit être décrit en détail. Incluez :*

*Titre et description du projet :*

*Nature du projet (application web, application mobile, etc.).*

*Contexte (professionnel, formation, stage, freelance, etc.).*

*Objectif (résoudre un problème spécifique, répondre à un besoin client).*

*Analyse préalable :*

*Expression des besoins du client ou de l'utilisateur.*

*Cahier des charges ou spécifications fonctionnelles.*

*Conception et organisation :*

*Diagrammes UML (use case, classes, séquence).*

*Modèle conceptuel de données (MCD).*

*Architecture choisie (MVC, microservices, etc.).*

*Outils ou méthodologies utilisés (Agile, Scrum, etc.).*

*Réalisation :*

*Technologies utilisées (langages, frameworks, bases de données).*

*Description des fonctionnalités développées.*

*Extraits de code significatifs (commentés et expliqués).*

*Gestion des versions (Git, GitHub/GitLab).*

*Problèmes rencontrés et solutions :*

*Décrivez les obstacles techniques ou organisationnels et comment vous les avez surmontés.*

*Tests et validation :*

*Types de tests réalisés (unitaires, fonctionnels, utilisateurs).*

*Résultats des tests.*

*Livrables :*

*Présentez ce qui a été livré (code source, documentation, déploiement).*

*Page notion :*

<https://mistertea.notion.site/Index-des-projets-CDA-65e4bd8d7cec45178ff38c3cbcfcc07e>

## Projets CDA

### APIs

#### Node.js noSql MongoDB (API / NoSql);

* API Zoologique d’identification d’Ours : SQL → CRUD;
* Movies / Netflix : SQL → CRUD;
* Foodtrucks : Node.js noSql MongoDB → CRUD ;
* Users : Node.js noSql MongoDB → CRUD ;
* API Utilisateurs Securisée: Node.js noSql MongoDB → CRUD + JWT;

### Applications

#### Symfony

* PartageToo : Site de troc/commerce style LeBonCoin
* Actunews : Blog d’actualités
* ForumNews : Réseau social de réaction / commentaire sur des articles

#### Php

* Surf : Blog en Php procédural, (Sql, Bootstrap)
* Marmitard : Blog culinaire Php procédural orienté Objet

#### Java Spring (Web):

* PayMe : Site de paiements style Lydia (Java, Spring, Thymeleaf, Sql)
* PayMe Hacking BackEnd : Site de paiements avec backdoor pirate (Java, Spring, Thymeleaf, Sql)
* Festival version Web : Site de gestion evenementielle. Ajout de groupes avec leurs scène a un ou plusieurs festivals à venir. Version Thymeleaf / Spring / SQL
* FirstApp : application avec rendu sans templating (Spring uniquement)

#### React

* Pokedex-React : React+ interrogation API

#### NodeJS / Symfony (BDDs Relationnelles) :

* festival (Java + Sql)
* blog (Sql via MCD)
* Actunews ( Sql via Doctrine )

#### JavaScript (Full Front-end)

* Histoire interactive (HTML/CSS)
* Intégrations psd to html : recipe-page, Vitrines (Assurance / Boutique)
  + SunnySide (Défi FrontEnd Mentor)

<https://sunnyside-challenge-chi.vercel.app/>

* + W-Challenge (FrontEnd Mentor)

<https://rubentrottein.github.io/w-challenge/>

* To-Do List Full JS
* Quiz Full JS

#### Java (Console)

* Calculatrice
* Zelda
* Garage
* Festival version console
* Chifumi
* Potager
* Médiathèque
* Appli Notes

#### Kotlin (Mobile)

* Calculatrice
* To-Do list (Array)
* Simon
* UserList (ArrayList)

#### Ionic (Mobile)

* Liste Triable des directives Ionic
* Tableau Front-end de liste d'utilisateurs
* To-do list étendue avec API de sauvegarde
* Application de gestion utilisateurs (Ionic-storage / Local storage)

## Projets perso

# Développement des compétences

*Pour chaque activité-type du titre professionnel, illustrez les compétences développées. Le référentiel est organisé en deux activités principales :*

*Développer des composants d'interface utilisateur.*

*Exemples : création d'une page web avec React ou Symfony, intégration d'APIs.*

*Lien avec vos projets.*

*Développer la persistance des données et des composants back-end.*

*Exemples : développement d'une API REST, gestion des bases de données.*

*Lien avec vos projets.*

*Associez les compétences décrites dans le référentiel officiel avec vos réalisations.*

# Résultats et bilan

*Auto-évaluation :*

*Points forts et axes d'amélioration.*

*Compétences acquises pendant la formation ou le projet.*

*Retour client ou utilisateur :*

*Si possible, incluez des témoignages ou retours des clients.*

*Perspectives :*

*Objectifs professionnels futurs.*

*Domaines dans lesquels vous souhaitez évoluer (web, mobile, IA, DevOps, etc.).*

# Annexes

Documentation technique des projets.

Diagrammes (MCD, UML).

Extraits de cahiers des charges.

Captures d’écran des applications ou démonstrations vidéo (liens vers un dépôt GitHub ou un portfolio en ligne si applicable).