

DOSSIER PROFESSIONNEL (DP)

Nom de naissance

▶ Agnan

Nom d'usage

▶ Agnan

Prénom

▶ Pierre-Alain

Adresse

▶ 2 rue de la prospérité (escalier c) La Varenne Saint-Hilaire 94210

Titre professionnel visé

Développeur web et mobile

MODALITE D'ACCES :

- ☒ 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.



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

Sommaire

Exemples de pratique professionnelle

Développer la partie back d'une application web	p.	5
▶ Intitulé de l'exemple n° 1 Mise en place d'un serveur web.....p.	p.	5-8
Développer la partie front d'une application web	p.	9
▶ Intitulé de l'exemple n° 1 Conception d'un portfolio sous la technologie reactJsp.	p.	9-10
▶ Intitulé de l'exemple n° 2 Création d'une application todo list dynamiquep.	p.	11-12
Titres, diplômes, CQP, attestations de formation	p.	13
Déclaration sur l'honneur		14

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1 Développer la partie back d'une application web

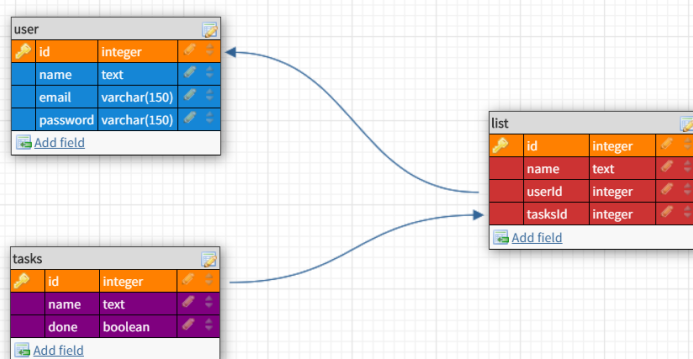
Exemple n°1 ► Mise en place d'un serveur web

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

V. REAC : Créer une base de données

Express, mcd, jeux de test

Avant de procéder à la création même de la base de données, j'ai conçu sous **db designer** le model conceptuel de données d'application. J'ai donc constitué trois tables : **user**, **list** et **tasks**.



Express Js m'a permis de procéder à la configuration du serveur **back-end** de mon application, tels que :

- La création de **schema via mongoose**.
- La mise en place **de router** contenant l'ensemble des requêtes.
- L'utilisation de package (exemple : cryptage des mots de passe, génération de token).
-

J'ai donc dans un premier temps installé **le node module express** avec la commande « **npm install express** ».

Enfin, j'ai effectué un grand nombre de jeux de test de données afin d'évaluer plusieurs cas de figures de requêtes.

VI. REAC : Développer les composants d'accès aux données

Pour le développement des composants d'accès aux données à l'aide de **nodeJs** et **expressJs**. J'ai procédé à la conception de mes requêtes en mettant en place des routes. Celles-ci fonctionnent de la manière suivante :

- ▶ Une route « **register** » qui aura pour objectif de récupérer des données inscrit dans le body, cette route dispose également d'une fonctionnalité de hachage du mot de passe de l'utilisateur pour plus de sécurité grâce au package **bcrypt** »
- ▶ La route « **login** » permet de comparer les données reçues et donc à l'utilisateur de s'identifier. Cela génère ensuite un token pour assurer plus de sécurité à la connexion de l'utilisateur. En comparant le mot de passe haché avec celui existant, cela permet à l'utilisateur de se connecter.
- ▶ Les routes **addList**, **updateUser** et **deleteList** permettent aux utilisateurs de gérer la création de leurs listes.
- ▶ Les routes **adTasks**, **updateList** et **deleteTasks** permettent aux utilisateurs de gérer la création des **tasks** qui seront liées aux listes précédemment créées par l'utilisateur.
- ▶ Enfin les routes **ListId** et **TasksId** permettent de récupérer l'ensemble des listes et des tasks qui ont été créées.

VII. Développer la partie back-end d'une application web ou web mobile

Dans un second temps, j'ai créé des variables contenant l'ensemble des dépendances (**bodyParser**, **mongoose**, **bcrypt**, **jwt**) dont l'utilisation sera nécessaire pour la conception du back de l'application :

- ▶ **body-parser** : body-parser est un middleware permettant de traiter et gérer les données postées dans le corps des requêtes (le body). Dans cet exemple il me permet notamment de traiter les données reçues en format **json**.
- ▶ **mongoose** : cette dépendance facilite le traitement des bases des données no sql de type document comme la création des modèles facilitant la vérification des données qu'on souhaite recevoir.
- ▶ **Bcrypt** : bcrypt est un package permettant de procéder au hachage des mots de passe créés par l'utilisateur. Dans le cas de la todo list, lorsque l'utilisateur s'enregistre leurs de mot de passe est automatiquement haché. Pour l'identification le mot de passe précédemment créé est comparé avec celui qu'entre l'utilisateur dans l'étape de la connexion.
- ▶ **Jwt** : Pour assurer une meilleure sécurité lors de l'authentification de l'utilisateur sur l'application. Le **jsonwebtoken (jwt)** effectue un hash de l'id de l'utilisateur afin de créer un token, ce token servira de signature de connexion.

DOSSIER PROFESSIONNEL (DP)

En ce qui concerne le développement de la partie back-end de l'application todo list, la conception de schéma par **mongoose** permet de concevoir une structure indiquant les données attendues par le serveur. Pour cela, j'ai construit 3 schémas : un schéma User, List et tasks.

Ces schémas fonctionnent de la manière suivante :

- ▶ **User** : Tout comme je l'avais imaginé au sein du mcd, le schéma user contient les champs **name**, **email**, **password** qui sont de **type string et requis** notamment pour l'inscription et la connexion de l'utilisateur. Le champs email dispose d'une propriété **unique** afin d'éviter la création de doublon d'adresse email à la l'inscription. Enfin, j'ai assigné à ce schéma une priorité **ref** qui correspond à la **foreign key** de la table list, car chaque user a la possibilité de créer plusieurs liste dans l'application.
- ▶ **List** : La table list contient un champ name de type **string** et d'une **foreign key** Tasks comme une list peut disposer de plusieurs tâches.
- ▶ **Tasks** : La table **tasks** contient un champ name de type **string**, d'un champ **done** de type **booléen** afin de voir si une tâche est finalisé ou non. Elle dispose également d'un champ date et enfin d'une **foreign key** List.

2. Précisez les moyens utilisés :

- ▶ J'ai utilisé les technologies **NodeJs**, **npm** et **ExpressJs** afin de procéder à la mise de toute la structure du serveur back de la **todoList**.
- ▶ J'ai effectué l'ensemble de ces tâches grâce au logiciel d'édition de code **Visual studio code**.
- ▶ **Postman** me permettait de tester mon api grâce aux requêtes que j'avais conçu, les tests réguliers sur ce logiciel m'a permis d'affiner mes routes.
- ▶ La gestion de la base de données no Sql s'effectuait via **Robo 3T**.
- ▶ Grâce à **dbdesigner** j'ai pu concevoir mon modèle conceptuel de données, reflétant la structure et la logique de ma base de données.

3. Avec qui avez-vous travaillé ?

Ce projet fut réalisé seul en autonomie.

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ► *Simplon*

Chantier, atelier, service ► *Projet back todoList*

Période d'exercice ► Du : *10/02/2020* au : *15/02/2020*

Activité-type 2 Développer la partie front d'une application web

Exemple n°1 ► Conception d'un portfolio sous la technologie reactJs

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

I. REAC : Maquetter une application

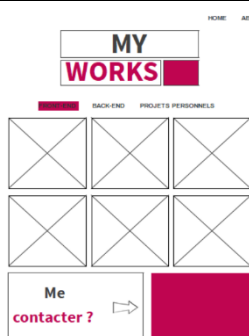
Dans le cadre de la conception d'un portfolio, ayant opté pour un format one page j'ai élaboré le maquetage de chacune des parties constituant mon portfolio (Home, about, my works, contact) :



Home



About



Works



Contacts

II. REAC : Réaliser une interface utilisateur web statique et adaptable

Concevant le projet portfolio sous la technologie reactJs, j'ai procédé à la création de l'application avec la commande **npx create-react app**. Ensuite avec l'utilisation des packages **Route, Switch et Router**, j'ai précisé au système que le composant home correspond à la racine de l'application.

Ainsi, j'ai divisé mon application en quatre composants principaux correspondant aux quatre parties principales du portfolio. Ensuite, via des composants class j'ai réalisé le rendu de chacune de ces parties dans le **render**. L'ensemble des ces composants se situe dans un dossier « components » que j'ai situé au niveau du src.

Ayant opté pour la méthode des **grid-templates**, j'ai adapté la mesure de ces grilles au niveau du media-queries de chaque device (dekstop, tablette et mobile) afin que mon application soit responsive.

2. Précisez les moyens utilisés :

- ▶ Pour réalisation des maquettes j'ai utilisé le logiciel **justimind** pour sa facilité de prise en main et des fonctionnalités très complètes offerte par l'application.
- ▶ L'ensemble de ces tâches grâce au logiciel d'édition de code **Visual studio code**.
- ▶ Pour la réalisation de ce projet, celui-ci fut conçu avec le Framework **React Js**.

3. Avec qui avez-vous travaillé ?

Ces projets ont été réalisés en autonomie.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *Simplon*

Chantier, atelier, service ▶ Conception d'un portfolio sous la technologie reactJs

Période d'exercice ▶ Du : *17/02/2020* au : *20/02/2020*

Activité-type 2 Développer la partie front d'une application web

Exemple n° 1 ► Création d'une application todo list dynamique

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

III. REAC : Développer une interface utilisateur web dynamique

Pour la mise en place de cette application monopage, l'utilisation du framework **React.js** m'a semblé une nouvelle fois pertinent pour la réalisation de ce projet. Afin de procéder à la création de l'application j'ai utilisé la commande **npx create-react app**.

J'ai donc dans un premier temps construit un **composant « Home »**, ce composant contiendra la quasi-totalité de l'ensemble de mes méthodes liées à cette application. Grâce à l'utilisation des packages **Route, Switch et Router**, j'ai donc logiquement précisé avec le paramètre **exact path** que le composant Home correspond à la racine de l'application.

Ensuite j'ai conçu cinq composants utiles à la mise en place de l'application :

- **Le composant inputForm** a pour rôle la réception des données entrées dans le formulaire de la todo list, pour cela j'ai conçu les props **valeurs, handleSubmit et handleChange**. La props « valeurs » pour la réception des données entrées dans l'input du formulaire. « handleSubmit » est appelé dans l'événement **onSubmit** afin de recevoir les données du formulaire au moment du click du bouton submit.
- **Le composant list** a pour objectif l'affichage des données entrées par l'utilisateur, en bouclant sur ce composant React Js reproduit la structure de celui-ci. Chaque paragraphe dispose donc de props **taskId, taskValue, taskDone, dataDone**. Afin que chacune des tâches disposent d'un id (taskId), d'une valeur qui puisse être visible (taskValue) et disposant d'un attribut data pour le changement d'une tâche réalisée ou en cours (dataDone).
- **Le composant title** permet d'afficher le nom de l'application.
- **Le composant filtre** consiste à filtrer les tâches de l'application todo list.
- **Le composant home** est le composant parent de l'ensemble des composants précédemment cités, les méthodes sont donc situées à ce niveau afin d'effectuer la logique des éléments enfants.

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

- ▶ L'ensemble de ces tâches grâce au logiciel d'édition de code **Visual studio code**.
- ▶ Pour la réalisation de ce projet, celui-ci fut conçu avec le Framework **React Js**.
- ▶ Je me suis appuyé sur la documentation officielle de React pour chacune des étapes de la conception de l'application.
- ▶ J'ai consulté régulièrement le site **stackoverflow** pour la résolution des problèmes rencontrés au cours du développement.

3. Avec qui avez-vous travaillé ?

Ce projet fut réalisé seul en autonomie.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *Simplon*

Chantier, atelier, service ▶ Conception d'une todo list sous React Js

Période d'exercice ▶ Du : *17/12/2019* au : *21/12/2019*

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

Intitulé	Autorité ou organisme	Date
Certification free formers	facebook	23/09/2019
Certification Opquast	Opquast	20/11/2019
Certification agile	Simplon	26/02/2020

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] Agnan Pierre-Alain ,
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 Saint-maur-des fossés le 16/03/2020

pour faire valoir ce que de droit.

Signature : Agnan Pierer-Alain