



Programme de formation

Formation développeur web/Développement web back -end

RNCP37674BC02

Certification DWWM - CCP 2 - Développer la partie back-end d'une application web et web mobile, délivrée par le Ministère du Travail du Plein Emploi et de l'Insertion et enregistrée le 13/05/2023.

VOTRE FORMATEUR :

BootCode

Formation 100% à distance développement web et web mobile
BootCode est le 1er Bootcamp à distance pour te reconvertir en un développeur opérationnel, recherché et recruté ! Grâce à notre approche pédagogique innovante en immersion, tu seras capable de relever les défis qui t'attendent. Entièrement créée par des développeurs, BootCode te propose un programme de 5 mois, taillé sur mesure pour apprendre non seulement à coder mais à structurer ton code. Chez BootCode, pas de grand discours académique, pas de promesse et de programme irréaliste en formation...

Infos utiles



Formation
Distancielle



Jours
104



Heures
832



Mois
5



Formation
Certifiante



Tarifs
6 300€ *
*Certification comprise



Contact
01 70 96 00 90
Contact@bootcode.co
www.Bootcode.co
117 avenue Victor Hugo, 92100
Boulogne-Billancourt

Financements

OPCO ✓

**MON
COMPTE
FORMATION** ✓

e
pôle emploi ✓

**BNP
PARIBAS** ✓

agefiph ✓

**TRANSITIONS
PRO** Partenaire
d'avenir ✓

Programme de formation

BootCode, 117 Avenue Victor Hugo, 92100 Boulogne-Billancourt. Siret : 91083842400018. NDA : 11922501192

Prérequis

Aucun diplôme n'est nécessaire. Il est toutefois recommandé d'avoir un esprit logique et une appétence pour les métiers de la tech

Public visés

Il s'agit de personnes en reconversion professionnelle ayant déjà touché au monde du digital.

ACCESSIBILITÉS

Handicap moteur ou physique Maladie chronique handicapante

DOMAINE

Conception web - développement web



Méthode

Notre expérience de 10 ans dans l'univers du développement, nous a permis de concevoir une méthode pédagogique et des programmes innovants qui réinventent l'apprentissage du code et enseignent les meilleures pratiques des développeurs professionnels.

Ici, tu apprendra l'art de la "clean architecture" à travers la méthode dite "hexagonale".

Chez Bootcode, tu n'apprends pas seulement à coder mais aussi à structurer ton code afin de te démarquer.

Tu progresses à ton rythme grâce à un coaching et à un suivi permanent. Notre équipe très réactive et attentive t'accompagnera vers la réussite

Objectifs

- Comprendre l'éco-système javascript autour de Typescript.
- Apprendre à construire un serveur en Node.JS
- Apprendre à rédiger et concevoir des tests automatisés.
- Devenez Junior Software Craftsmanship grâce aux techniques de Test Driven Development et Domain Driven Design.
- Gérer une base de donnée.
- Apprendre les meilleures pratiques de développement autour de l'architecture hexagonale.

Programme

Module 1 : Bases et culture des sciences de l'informatique

Chapitre 1 - Les sciences de l'informatique

Objectif : Comprendre l'environnement technique et acquérir une culture de base sur les sujets qui seront vus tout au long de la formation.

- Présentation du monde du développement web, les particularités, les types de développeur et qu'est-ce qu'un développeur en 2022 ?
- Qu'est-ce qu'un ordinateur, de quoi est-il composé et comment fonctionne-t-il?
- Présentation et utilisation du shell
- **Évaluation individuelle : Écrire un premier script en shell permettant de créer un dossier avec un fichier texte rempli à l'intérieur.**

Objectif : Apprendre à comment gérer son code à travers git, l'outil de commande, et de gestion et comment organiser son travail ?

- Qu'est-ce qu'un langage informatique ? Pourquoi tant de langages et à quoi servent-ils ? Différence entre un langage interprété et compilé.
- Décomposition d'un site internet et compréhension de l'écosystème web.
- Installation et utilisation basique des commandes git.
- **Évaluation individuelle : Construire son premier repository avec l'exercice fait la semaine 1.**

Programme

Objectif : Mise en place de son environnement de travail.

- Présentation d'un serveur applicatif, comment fonctionne t-il?
Comment les sites internet communiquent entre eux et où se situe le métier de développeur dans tout cela.
Préparation des ordinateurs des élèves pour installer tous les logiciels nécessaires afin de commencer les premiers programmes en Javascript.
-
- **Point global et retour sur les trois premières semaines. Réexplication des notions et exemples montrés par le formateur. Évaluation orale.**

Objectif : Évaluation sur une semaine et entraînement au travail collaboratif

- **Évaluation collective : Par groupe de deux, les élèves devront réaliser une évaluation basée sur des exercices successifs qui leur permettront d'automatiser leurs premières tâches par l'écriture d'un code en *scripting shell*. Le code sera déposé sur un *repository* dont le groupe fera partie.**
- Revenir sur les notions, explications et exemples concrets de cas d'usage.

Programme

Chapitre 2 : Le Javascript, un langage d'expert

Ce chapitre comporte de nombreux exercices et permet aux élèves de commencer à développer leur logique algorithmique.

Objectif : Apprendre les bases du langage Javascript.

- Introduction au Javascript.
- Présentation de Node.JS et de son moteur chrome comme exécuteur de code JS
- Print d'un premier log dans le terminal via le lancement d'un fichier JS grâce à NodeJS.
- Les types primitifs
- Les variables
- Les opérateurs
- Les commentaires
- **Évaluation orale sur les notions**

Objectif : Apprendre à se servir des différents opérateurs et méthodes inhérentes au langage.

- Les fonctions (Décomposé en deux parties)
- Utilisation d'une fonction inhérente à une class (ex : Math.round(number))
- Écriture d'une fonction comportant des paramètres.
- Les tableaux, les boucles et les objets.
- **Évaluation individuelle : autour du célèbre kata FizzBuz" afin d'assimiler les notions des semaines précédentes**

Programme

Module 2 : Le backend, les processus de développement et Typescript

Notre spécialisation repose sur l'apprentissage d'un code typé. C'est pourquoi nous travaillerons principalement sur Typescript qui est un superset de Javascript avec du typage afin que l'élève comprenne mieux son code et puisse voir ses erreurs le plus tôt possible.

Chapitre 1 - Typescript, les typages et les interfaces (1 semaine)

Objectif : Apprendre les différents types primitifs d'un langage informatique.

- Introduction à Typescript : Setup et compilation via ts-node
- Différence entre type et interface
- Apprentissage des pattern identity type et conditionnal type afin de bien comprendre le point de rupture entre un type et une interface.
- Décrire des premières données : Compréhension d'une demande client sous forme de user story et réalisation d'une interface décrivant des propriétés attendues par la spécificité.
- **Évaluation individuelle : Refactorisation d'une base de code Javascript en Typescript avec ajout de types, d'interfaces et d'enum.**

Programme

Chapitre 2 - Ma première API

Objectif : Apprendre les bases d'une api REST par le biais de Node.JS et d'un framework populaire express.

Nous nous intéressons de plus près à Node.JS, de quoi est-il composé lors de l'initialisation du projet et ce qu'est une Api REST.

- Introduction au package.json et compréhension de l'éco-système npm.
- Installation de express et réalisation d'un premier endpoint.
- Installation de Insomnia et appel de notre premier endpoint.
- **Évaluation individuelle : Mise en place d'une base de données dites "in-memory" et réalisation de plusieurs endpoint de type GET - POST - PUT - DELETE et PATCH.**

Objectif : Apprendre les bases de la Programmation Orientée Objet qui sera vu tout au long du cursus.

- Explication et exemple d'une "class" et de ce à quoi cela va nous servir.
- Principe de l'héritage et de l'encapsulation.
- **Exercice évaluation : Reprise de l'api créé initialement, le but sera de créer une class *InMemoryUserRepository* qui gère la suppression, la lecture et la mise à jour de donnée dit "in-memory" et prise en main des imports / exports de librairies, code.**

Programme

Chapitre 3 - Authentification et Autorisation

Objectif : Apprendre à authentifier et à autoriser un client à accéder à notre api.

- Introduction au différent mécanisme de cryptographie.
- Utilisation de la librairie *Bcrypt* dans le cadre d'un envoi de mot de passe au serveur afin que les mots de passe soient stockés de manière sécurisée.
- Introduction à l'utilisation d'un *Json Web Token* lors d'un *login*.
- Réalisation d'un *middleware* d'authentification afin de protéger ses routes contre des requêtes non authentifiées.
- **Évaluation individuelle : Remaniement de l'api existante et apprentissage des notions de "Gateway" pour réaliser les vérifications de token ou d'encryption de mot de passe.**

Chapitre 4 : Testing

Objectif : Apprendre à écrire des tests d'API avec le framework jest.

- Introduction à la librairie *jest* et au fichier *tsconfig.json* afin d'augmenter les capacités de Typescript.
- Écriture de notre premier test. L'idée ici sera de tester une simple fonction afin de montrer comment le moteur de test fonctionne.
- Mise en place de tests d'intégration à l'aide de *supertest* afin d'automatiser les tests de l'api.
- Présentation de *user story* et introduction au *Test Driven Development*. L'objectif est de s'habituer à réfléchir à l'intention de son code avant de coder via le test.
- **Évaluation individuelle : Réalisation des tests d'intégration sur l'api créée initialement.**

Programme

Chapitre 5 : MongoDB - Première utilisation d'un système de stockage externe

Objectif : Apprendre à se servir d'un système de stockage de données externes.

- Présentation de *mongoose* et installation de *mongodb* sur les machines des élèves.
- Réalisation d'une interface *UserRepository* afin de démontrer la puissance de l'interchangeabilité des dépendances.
- Réalisation d'un *repository MongoDB* qui implémente les mêmes méthodes que la *class InMemoryUserRepository*.
- **Exercice évaluation: Changements de la couche de données afin de s'assurer que le changement de dépendances n'a pas causé d'anomalies par le biais des tests écrits précédemment.**

Chapitre 6 : Le Domain Driven Design (DDD)

Objectif : Apprendre une nouvelle méthodologie

- Ce chapitre donnera toutes les clefs nécessaires à l'insertion professionnelle et à la poursuite en autonomie de l'élève à l'issue de la formation dans le métier de développeur.
- Il apprendra à structurer son code de manière qualitative et professionnelle.
- Cela permettra de donner à l'élève, un cadre de travail et une méthodologie qu'il pourra utiliser dans n'importe quelle situation.
- **Évaluation orale sur les notions.**

Programme

Objectif : Apprendre à challenger le métier et avoir une parfaite compréhension du besoin client.

- Présentation du *pattern DDD*.
- Refactorisation de la *codebase* existante afin de sensibiliser les élèves à l'organisation d'un code organisé en couches via un dossier "*core*", comportant du code absous de tout *framework* et "*adapters*" qui lui intégrera les parties applicatives.
- Réalisation d'un premier *Aggregate Root User* qui comportera une méthode *static create* et une *ValueObject Email* qui fera la validation d'email via une *Regex*.
- **Exercice évaluation : Utilisation de l'agrégat au sein de l'api créée initialement afin de faire les validations nécessaires et ainsi sécuriser le serveur et garantir l'intégrité des données.**

Objectif : Apprendre à découper une fonctionnalité au travers d'un agrégat.

- Découvertes de ce qu'est une spécificité fonctionnelle et description d'un *Aggrégat* et ses logiques métiers.
- Initiation au *usecase* et aux techniques de composition d'une *class* via les interfaces.
- Réalisation de tests unitaires sur les *usecase* en utilisant une implémentation *InMemory* pour s'assurer que les *specs clientes* sont bien respectées.
- **Évaluation individuelle: Refactoriser les *api* et décomposer son code via des interfaces et *usecases* (cas métiers).**

Programme

Module 3 : Projet commun : Réalisation d'un serveur applicatif

Objectifs : ce module mettra en avant les points forts du travail collaboratif et prendra en compte les avancés de chaque élève individuellement.

C'est une vraie mise en situation professionnelle qui sera faite ici, telle qu'elle a été expérimentée par les formateurs lors de leurs missions professionnelles.

Le formateur décidera de la configuration des équipes. Ce sera l'occasion d'échanger et de profiter au maximum des connaissances apprises et acquises durant les mois précédents.

Tout au long du projet, les élèves seront challengés par des tâches annexes qui leur donneront les clefs et leur permettront de réaliser le projet commun.

Etape 1 :

- Présentation du projet global : Réalisation d'un serveur applicatif pour convertir du texte en format audio.
- Retours sur les difficultés globale de la classe et répartition de différents workshops technique tout au long du projet.
- Cours sur le Domain Driven Design et comment découper une grande application en petits sous-métiers.
- Présentation du monorepo avec le framework nx et comment tirer partie de Typescript dans un projet multidimensionnel.
- **Point de situation individuel entre l'élève et le formateur pour comprendre ses blocages et voir des pistes de progression. En fonction des éléments de réponse, l'élève se verra attribuer un rôle précis dans le projet commun afin qu'il tire au maximum les avantages de la formation et qu'il soit en mesure de parfaire sa connaissance du monde du développement.**

Programme

Etape 2 :

- Découpage de la classe en différentes "*feature team*" ou équipe spécialisée sur un corps de fonctionnalités.
- Equipe "Onboarding" (Authentification et inscription), équipe "TTS" (*Text To Speech*) et équipe "paiement".
- Présentation de l'outil *Jira* et premier ticket à développer.
- Découpage des tickets entre les différentes équipes. Chaque équipe devra en concertation se dispatcher les tâches entre chaque membre.
- Une fois les tâches distribuées. Le professeur veillera à la bonne communication des membres et appuiera les équipes en support sur les différents premiers sujets à développer.
- **Evaluation orale sur les différents tickets et taches distribuées**

Etape 3:

- *Sprint Planning* avec toute la classe et affectation des tâches de la semaine.
- *Workshop* technique avec l'ensemble de la classe pour savoir comment comprendre le besoin client et comment mettre en pratique les connaissances acquises précédemment. Le professeur fera office de *Project Manager* et détaillera le besoin à l'élève qui sera mis en situation professionnelle.
- Exercice sur la description du besoin. L'élève devra mettre en place un plan de développement et réfléchir à comment concevoir la fonctionnalité par écrit qu'il devra remettre au formateur.
- **Évaluation individuelle : Oral de 30 minutes pour chaque élève qui détaillera son plan pour développer la fonctionnalité. Chaque élève sera challengé par le professeur qui remettra en question leurs travaux.**

Programme

Etape 4 :

- *Sprint Planning* pour démarrer la semaine et affectation des nouvelles tâches. Pour la simulation du projet. Les reliquats ne seront pas reconduits les semaines suivantes, afin que l'élève puisse acquérir et utiliser le maximum d'outils et de connaissances.
- Réorganisation des équipes en deux parties. L'une sera rattachée au pôle "paiement" et l'autre sera rattachée au pôle "voix". L'objectif pour la première sera d'intégrer l'api Stripe et la seconde sera d'intégrer *Google Text-To-Speech* afin de transformer du contenu écrit en contenu audio.
- Atelier de brainstorming autour des différentes problématiques du projet et découpage en mini-tâche.
- **Évaluation collective : Chaque équipe, au maximum de 4 personnes, devra présenter son plan de développement, présenter les fonctionnalités de leurs api respectives et résumer leurs plans de tests en oral de 1 heure. Préparation de l'évaluation sur 3 jours.**

Etape 5 :

Objectif : Apprendre à travailler en méthode pair-programming afin de favoriser l'entraide et la bienveillance.

- Intégration des api respectives. Chaque élève ayant eu sa tâche attribuée devra tout au long de la semaine, la réaliser en *pair-programming* (groupe de deux personnes).
- **Evaluation collective : Le formateur veillera, tout au long de la semaine à travers des réunions journalières avec chaque duo, qu'il n'y a pas de blocage de la part des élèves et s'assurera de la mise en place des fonctionnalités demandées.**

Programme

Etape 6 :

Objectif : Maîtriser sa communication écrite et orale face à un recruteur tech ou un client et apprendre à se démarquer grâce aux techniques de programmation apprises durant le cursus et présentation de la certification.

- Atelier d'entretiens client et recruteur tech.
- Rédaction CV de chaque élève, organisation de son compte LinkedIn et de son compte Malt.
- Atelier de création du statut auto-entrepreneur et explication sur les différentes modalités de création de la structure juridique.
- **Évaluation individuelle finale**

L'élève devra présenter le projet fini. Parler des compétences acquises tout au long du projet, de ses difficultés et de comment il a tiré parti des compétences acquises au sein de la formation en vue du passage de la certification.

Modalités d'accès

- 1er prise de contact
- Mail récapitulatif (site internet, programme de formation, mais aussi les différents financements en fonction des situations)
- Entretien d'évaluation 10/15min (quelle que soit la formation)
- **Quiz culture générale pour l'offre back-end**
Réponse immédiate. Si la réponse est positive, intégration
- d'une des sessions de formation.
- Si autofinancement : Intégration rapide.
- Si organismes de financement : délai un peu plus long.

Délais d'accès

- Prise de contact rapide pour des questions de places disponibles et de logistiques.
- Autofinancement : intégration rapide.
- Organismes de financement : délai de 4 mois minimum à respecter avant le début de la formation.

Modalités d'évaluation

- Evaluation toutes les semaines. En fin de semaine.
- Évaluation orale et / ou travaux à rendre sur GitHub.
- Une leçon apprise = 1 petit projet à rendre.
- Fin de formation : projet final à rendre qui servira pour la certification.

Prochaines sessions

- **Formation développeur back-end.**

Du 08/09/2025 au 30/01/2026



Blocs de compétences

On notera qu'il est tout à fait possible de valider un ou des blocs de compétences au préalable en vue du passage de la certification.

En effet, si le candidat a déjà passé et obtenu une des certifications, inutile de la repasser par la suite.

Ce qui est acquis est acquis !