

Développement

## IoT - Développer des applications connectées avec Java

4 jours (28h00) | ★★★★★ 4,6/5 | IOT-JAVA | Évaluation qualitative de fin de stage | Formation délivrée en présentiel ou distanciel <sup>(1)</sup>

Formations Informatique > IoT - Objets connectés > Développement



### À l'issue de ce stage vous serez capable de :

- Programmer efficacement en Java pour la technologie IoT.

### Niveau requis

Avoir une bonne pratique de Java et des environnements de développement. Connaître les réseaux et les hardwares est souhaité.

### Public concerné

Programmateurs et chefs de projets.

### Cette formation :

- Est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- Bénéficie d'un suivi de son exécution par une feuille de présence émarginée par demi-journée par les stagiaires et le formateur.

#### (1) Modalité et moyens pédagogique :

Formation délivrée en présentiel ou distanciel \* (e-learning, classe virtuelle, présentiel à distance). Le formateur alterne entre méthodes \*\* démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation). La validation des acquis peut se faire via des études de cas, des quiz et/ou une certification.

Les moyens pédagogiques mis en oeuvre (variables suivant les formations) sont : ordinateurs Mac ou PC (sauf pour les cours de l'offre Management), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur ou écran tactile interactif (pour le distanciel). Environnements de formation installés sur les postes de travail ou en ligne. Supports de cours et exercices.

\* Nous consulter pour la faisabilité en distanciel. \*\* Ratio variable selon le cours suivi.

# Programme

## Jour 1 : Présentation de l'écosystème IoT, API Embedded J2SE

### Ecosystème de l'Internet des Objets (IoT)

- IoT vs M2M
- Architecture et concepts de base
- Communication / données, capteurs et gestion de l'énergie
- Formats de données et stockage
- Domaines et types d'applications IoT
- Langages et environnements de développement
- Offre du marché
- Normes
- Sécurité
  - Authentification
  - Confidentialité

### Techonologies et architecture de IoT (vue d'ensemble)

- Infrastructure IoT
  - Composants
  - Capteurs
  - Réseaux sans fil
- Extensions et lien avec l'infrastructure classique et le Cloud
- Systèmes d'exploitations et tendance IoT
  - Raspbian
  - Windows
  - Linux...
- Réseaux physiques de proximité
  - Wi-Fi
  - Bluetooth
  - RFID
  - LoRa
  - LoRaWAN...
- Protocoles de communication / Broker
  - MQTT
  - API REST...

### Programmation Java pour IoT

- Présentation et choix d'IDE
- Librairie IoT for Java : Cloud IoT
- Programmation Java J2SE Embedded
- Présentation de la plateforme de développement

### Exemples de travaux pratiques (à titre indicatif)

- *Installation et configuration de la plateforme de développement : Raspbian, Arduino*
- *Classe template*
- *Les collections*
- *Gestion de fichiers*
- *Gestion des données en bases relationnelles*
- *Interface graphique*
- *Test unitaire*
- *Déploiement d'applications*

## **Jour 2 : General Purpose Input / Output (GPIO)**

### **Programmation du GPIO**

- I/O numériques
- I2C
- Impulsion PWM (Pulse Width Modulation)
- Gestion des interruptions
- Capteurs
  - GPS
  - Ultrasonique
  - Mouvement
  - Inclinaison
  - Vibration...

### **Exemples de travaux pratiques (à titre indicatif)**

- Programmation des I/O numériques
- Contrôle de moteur au moyen de PWM
- Gestion des interruptions
- Capteurs (GPS, ultrasonique, d'inclinaison, de mouvement...)

## **Jour 3 : Communication et multi-thread**

- Communication
- Programmation bluetooth
- Programmation TCP/IP au moyen de sockets
- Programmation MQTT Broker
- Intégration Arduino
  - Programmeur FTDI

### **Exemples de travaux pratiques (à titre indicatif)**

- Configuration bluetooth
- Client / Serveur : TCP
- Client / Serveur diffusion multicast : UDP
- Echanges Raspberry / Arduino
- Capteur ultrasonique
- Capteur d'inclinaison
- Reconnaissance vidéo avec OpenCV

## **Jour 4**

### **Exemples de travaux pratiques (à titre indicatif)**

- Programmation d'un robot "pelleteuse" : Moteurs / PWM, Arduino, MQTT, Raspberry
- Programmation d'un robot qui évite les obstacles : ultrasons
- Programmation d'un détecteur de passage : capteur de mouvements

## **Modalités d'évaluation des acquis**

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation ou une certification (M2i ou éditeur)