

- *Maîtriser le cœur de la technologie Node.JS*
- *Optimiser les performances de leurs applications*
- *Améliorer la qualité des applications*

3**Prix : 1?650 € € / HT****OUTILS PÉDAGOGIQUES****MODALITÉS D'ÉVALUATION****MODALITÉS DE FINANCEMENT****MODALITÉS ET DÉLAIS D'ACCÈS****OBJECTIFS PÉDAGOGIQUES****ACCESSIBILITÉ****LES POINTS FORTS DE LA FORMATION****PRÉ-REQUIS**

- Maîtrise de JavaScript et 1ères expériences avec Node.JS

MODALITÉS ET DÉLAIS D'ACCÈS**ATTESTATION OBTENUE****EFFECTIF DE LA FORMATION****CERTIFICATION****MODALITÉ PÉDAGOGIQUE**

Cours dispensé en mode présentiel avec une alternance d'apports théoriques et méthodologiques, et de mises en situations pratiques

PROCHAINES SESSIONS

Nous consulter.

Nous consulter.

Nous consulter.

PROGRAMMES DE NODE.JS AVANCÉ**Rappels sur Node.JS**

- Installation et utilisation du REPL : présentation nvm&co
- JavaScript côté serveur : démonstrations et premiers travaux pratiques
- La « single-threadedeventloop »
- API non bloquante : intérêts
- Aller plus loin avec les objets JavaScript?: Object.create,Object.defineProperty
- La programmation fonctionnelle?: map, reduce, currying (illustration avec lodash)

La programmation asynchrone

- Dompter le paradigme asynchrone?
- Les différentes API : callback et bibliothèques associées, fibers, promesses
- Les avantages et pièges à éviter

- Gérer la soupe de callbacks avec Async

Les modules NODE.JS

- Description
- Fonctionnement
- Structure

Écrire un serveur avec Node.JS

- Présentation
- Démonstration

Les bibliothèques d'accès aux bases de données

- Présentation
- NoSQL: MongoDB et Redis
- Modélisation avec NoSQL
- Différentes utilisations de Redis
- Implémentation du modèle

Communication en temps réel

- Définition et problématiques
- Les technologies à disposition
- Intégration des WebSockets HTML5 avec Socket.IO

Communication inter-process en temps réel

- Le pub/sub avec Redis
- AMQP dans Node JS
- Présentation RabbitMQ et ZeroMQ
- Event-loop distribuée

Les tests avec NodeJS

- Tests unitaires avec Mocha (atelier)
- Tests fonctionnels avec les headless browsers
- Intégration avec npm

Travaux pratiques

- Exercices sur les server-sideevents, websockets avec socket.io

Gestion de la performance avec Javascript et Node.JS

- Écrire du JavaScript performant pour V8 : les bonnes pratiques
- Gestion de la mémoire : la pile et la mémoire totale, comment les gérer
- Anticiper et trouver les fuites mémoire
- Effectuer des calculs lourds : pool de workers, amqp...
- Les Cluster
- Utiliser tous les processeurs de sa machine
- Cluster et données partagées
- La solution haute performance Redis
- Bonus spécial troll
- Comparaison avec l'équivalent Apache/PHP

Qualité

- Déboguer son application : utilisation du débogueur v8
- Profiling : état des lieux, outils
- Bonnes pratiques : les meilleurs outils de test, contrôle de qualité ducode, conventions...
- Discussion libre

Conclusion et conseils

- Synthèse des bonnes pratiques de développement NodeJS
- Veille : comment faire le tri dans les milliers de modules disponibles ?
- Le futur avec ES6 : let, const, arrows, classes, destructuration, proxy, observation, generators, comprehensivelists, collections, promesses...
- Le futur présent : fonctionnalités ES6 utilisables dans Node