

Formation Neo4J pour Développeur

Objectifs de la formation Neo4J pour Développeur

Neo4J est une base NoSQL permettant de gérer des données fortement liées les unes aux autres sous forme de graphe. Cette approche permet d'obtenir de bien meilleures performances lors d'un parcours de graphe qu'avec une base de données relationnelle et des jointures.

Cette formation est axée sur le développement avec Neo4J et non sur l'administration. Cependant il est nécessaire de comprendre les concepts et le fonctionnement général de Neo4J afin d'être préparé à l'apprentissage de la programmation qui présente des différences nettes par rapport au SQL traditionnel de par le format de stockage des données dans cette base NoSQL.

Concrètement, à l'issue de la formation Neo4J pour Développeur, vous serez capable de :

- Comprendre les spécificités d'une base NoSQL par rapport à un SGBDR
- Savoir installer Neo4J et utiliser les outils du développeur
- Comprendre la structure des données et le vocabulaire associé dans Neo4J
- Savoir manipuler des données sous forme de graphe
- Améliorer les performances avec les index
- Accéder aux données de Neo4J depuis des programmes tiers (C#, Java, JavaScript, Python)
- Accéder aux données de Neo4J via l'API HTTP

À qui s'adresse cette formation ?

Public :

Cette formation s'adresse aux Développeurs Java, Python ou JavaScript.

Prérequis :

Afin de profiter de cette formation, il est nécessaire que les participants aient des connaissances des principes des bases de données ainsi que des langages de programmation Java, Python ou JavaScript.

Contenu du cours Neo4J pour Développeur

Introduction à Neo4J

La domination historique des bases de données relationnelles

L'apparition des bases NoSQL. Les différents types de bases NoSQL

Présentation de Neo4J, son positionnement, ses avantages

Exemples d'utilisation de Neo4J

Installation de Neo4J

Plateformes supportées et configuration requise

Installation sous Linux, lancement du serveur

Installation sous Windows, création d'un service, lancement

Configuration, shell et interface web

Travaux Pratiques :

Installer Neo4J sur son poste de développement

La structure des données sous Neo4J

Les nœuds : propriétés et labels

Les relations : propriétés, types et sens

Les propriétés : types et tableaux

Chemins et parcours de données

Schéma. Index et contraintes

Travaux Pratiques :

Exploration du schéma de la base d'exemple

Le langage Cypher

Présentation

La syntaxe : expressions, variables, paramètres, opérateurs, motifs, tableaux...

Les clauses : lectures, écritures, unions, projections...

Les fonctions : mathématiques, agrégations, scalaires, liées aux tableaux et chaînes...

Les index : création, utilisation, suppression, ajout de contraintes

Optimisation de requêtes et plans d'exécution

Importation de données

Travaux Pratiques :

Importation de la base d'exemple et exécution de requêtes via l'interface web

Accéder à Neo4J depuis vos propres programmes

Les drivers fournis par Neo4J

Aperçu des drivers C#, Java, JavaScript, Python

Focus sur le driver Java, JavaScript ou Python

- Connexion au serveur et authentification

- Requêtes en lecture et écriture

- Transactions

Aperçu de l'embarquement de Neo4J au sein d'un programme Java

Travaux Pratiques :

Exécution de requêtes via le driver Java, JavaScript ou Python

L'API HTTP

Racine et authentification

Flux JSON

Service transactionnel

Manipulation directe de nœuds

Manipulation directe de relations

Service historique (transactionnel et non transactionnel)

Recensement des propriétés, labels et types de relations utilisés dans la base

Gestion de l'indexation et des contraintes

Travaux Pratiques :

Exécution de requêtes via l'API HTTP grâce à l'outil cURL