

Ingénieur Backend Java : Questions d'Entretien

Java Core

1. Quels sont les quatre principes fondamentaux de la POO en Java ? Réponse : Les quatre principes fondamentaux sont l'encapsulation, l'héritage, le polymorphisme et l'abstraction. L'encapsulation cache l'état interne d'un objet, l'héritage permet l'héritage de classes, le polymorphisme permet le surchargement et la surcharge de méthodes, et l'abstraction fournit un moyen de représenter les caractéristiques essentielles sans inclure les détails de fond.
2. Expliquez le but des génériques en Java et fournissez un exemple. Réponse : Les génériques permettent de paramétrer les types, ce qui permet la réutilisation du code et la sécurité des types. Par exemple, `ArrayList<T>` utilise un paramètre de type `T` pour stocker des éléments de n'importe quel type.
3. Comment crée-t-on un thread en Java, et quel est son cycle de vie ? Réponse : On peut créer un thread en étendant `Thread` ou en implémentant `Runnable`. Le cycle de vie comprend les états Nouveau, Exécutable, En cours d'exécution, Bloqué, En attente, En attente temporisée et Terminé.
4. Décrivez les différentes zones de mémoire gérées par la JVM. Réponse : La JVM gère le tas, la pile, la zone des méthodes, la pile des méthodes natives et le registre du compteur de programme. Le tas stocke les objets, tandis que chaque thread dispose de sa propre pile pour les variables locales et les appels de méthode.
5. Quelle est la différence entre les exceptions vérifiées et non vérifiées en Java ? Réponse : Les exceptions vérifiées doivent être déclarées ou capturées, tandis que les exceptions non vérifiées ne sont pas vérifiées au moment de la compilation. Par exemple, `IOException` pour les vérifiées et `NullPointerException` pour les non vérifiées.
6. Comment met-on en œuvre la sérialisation en Java, et pourquoi est-elle importante ? Réponse : La sérialisation est mise en œuvre en implémentant l'interface `Serializable`. Elle est importante pour sauvegarder et restaurer l'état d'un objet, utile pour le réseau et la persistance.
7. Comparez `ArrayList` et `LinkedList` dans le framework de collections Java. Réponse : `ArrayList` est adapté pour un accès et un parcours rapides, tandis que `LinkedList` est meilleur pour les insertions et les suppressions. `ArrayList` utilise une mémoire contiguë, tandis que `LinkedList` utilise des nœuds avec des pointeurs.
8. Quelles sont les expressions lambda en Java, et comment se rapportent-elles aux interfaces fonctionnelles ? Réponse : Les expressions lambda fournissent un moyen concis de représenter une interface à une seule méthode (interfaces fonctionnelles). Elles sont utilisées pour implémenter des interfaces fonctionnelles comme `Runnable` ou `Comparator`.
9. Expliquez les opérations clés disponibles dans l'API Stream Java. Réponse : L'API Stream inclut des opérations intermédiaires (par exemple, `map`, `filter`) et des opérations terminales (par exemple, `forEach`, `collect`). Elles permettent des opérations de style fonctionnel sur les collections.

10. Comment utilise-t-on la réflexion en Java pour inspecter les classes en temps d'exécution ? Réponse : La réflexion permet d'inspecter les classes, les méthodes et les champs en utilisant `Class.forName()`, `getMethods()` et `getField()`. Elle est utilisée pour le comportement dynamique et les frameworks.
-

Écosystème Spring

1. Qu'est-ce que le conteneur IoC de Spring, et comment fonctionne-t-il ? Réponse : Le conteneur IoC gère les beans et leurs cycles de vie. Il utilise l'injection de dépendances pour gérer les dépendances, réduisant ainsi le couplage.
2. Expliquez l'auto-configuration de Spring Boot. Réponse : L'auto-configuration configure automatiquement les beans en fonction des dépendances de la classpath, simplifiant la configuration et réduisant le code de configuration.
3. Comment Spring Data JPA simplifie-t-il l'accès aux données ? Réponse : Spring Data JPA fournit des dépôts avec des opérations CRUD et des méthodes de requête, abstraisant les interactions avec la base de données.
4. À quoi sert Spring Security ? Réponse : Spring Security fournit des mécanismes d'authentification et d'autorisation, sécurisant les applications contre les accès non autorisés.
5. Décrivez le rôle de Spring MVC dans les applications web. Réponse : Spring MVC gère les requêtes web, mappant les URL aux contrôleurs et gérant les vues et les modèles pour les réponses web.
6. Qu'est-ce que Spring Cloud et quels sont ses principaux composants ? Réponse : Spring Cloud fournit des outils pour construire des applications natives du cloud, y compris la découverte de services (Eureka), les disjoncteurs (Hystrix) et les passerelles API.
7. Comment Spring AOP améliore-t-il la fonctionnalité de l'application ? Réponse : AOP permet de séparer les préoccupations transversales comme la journalisation et la gestion des transactions du code métier, en utilisant des aspects et des conseils.
8. Qu'est-ce que Spring Boot Actuator, et que fait-il ? Réponse : Actuator fournit des points de terminaison pour la surveillance et la gestion des applications, tels que les vérifications de santé, les métriques et les informations sur l'environnement.
9. Expliquez l'utilisation des profils Spring. Réponse : Les profils permettent différentes configurations pour différents environnements (par exemple, développement, production), permettant des paramètres spécifiques à l'environnement.
10. Comment les démarreurs Spring Boot simplifient-ils la gestion des dépendances ? Réponse : Les démarreurs incluent toutes les dépendances nécessaires pour une fonctionnalité spécifique, réduisant ainsi le besoin de gérer manuellement les dépendances.

Architecture des Microservices

1. Qu'est-ce que la découverte de services, et pourquoi est-elle importante ? Réponse : La découverte de services automatise le processus de localisation des services, essentielle pour les environnements dynamiques et le dimensionnement.
2. Expliquez le rôle d'une passerelle API dans les microservices. Réponse : Une passerelle API agit comme un point d'entrée unique, acheminant les requêtes vers les services appropriés, gérant la sécurité et la traduction de protocole.
3. Qu'est-ce que le modèle de disjoncteur, et comment aide-t-il ? Réponse : Le disjoncteur empêche les pannes en chaîne en interrompant les requêtes vers les services défaillants, leur permettant de se rétablir.
4. Décrivez les principes de conception d'API RESTful. Réponse : Les principes REST incluent l'absence d'état, l'architecture client-serveur, la mise en cache et l'interface uniforme, assurant des API évolutives et maintenables.
5. Qu'est-ce que GraphQL, et en quoi diffère-t-il de REST ? Réponse : GraphQL est un langage de requête pour les API, permettant aux clients de demander exactement ce dont ils ont besoin, réduisant ainsi la surcharge et la sous-charge.
6. Comment gérez-vous le versionnage des API dans les microservices ? Réponse : Le versionnage peut être effectué via les chemins d'URL, les en-têtes ou les paramètres de requête, assurant la compatibilité ascendante et des transitions en douceur.
7. Expliquez le modèle Saga dans les microservices. Réponse : Saga gère les transactions distribuées entre les services, en utilisant une série de transactions locales et de compensations en cas d'échec.
8. Qu'est-ce que les vérifications de santé dans les microservices, et pourquoi sont-elles importantes ? Réponse : Les vérifications de santé vérifient la disponibilité et les performances des services, cruciales pour la surveillance et la gestion des maillages de services.
9. Décrivez le développement contract-first dans les microservices. Réponse : Le développement contract-first définit les API avant la mise en œuvre, assurant la compatibilité et le découplage entre les services.
10. Comment mettez-vous en œuvre la limitation de débit dans les microservices ? Réponse : La limitation de débit peut être mise en œuvre à l'aide de middleware ou d'API comme Spring Cloud Gateway, contrôlant les taux de requête pour prévenir les abus.

Bases de données et Mise en cache

1. Qu'est-ce que les joints SQL, et quand sont-ils utilisés ? Réponse : Les joints SQL combinent des enregistrements de deux tables ou plus en fonction d'une colonne liée, utilisés pour récupérer des données à travers des tables liées.
2. Expliquez les propriétés ACID dans les transactions de base de données. Réponse : ACID signifie Atomicité, Cohérence, Isolation et Durabilité, assurant un traitement de transaction fiable.
3. Qu'est-ce que Redis, et comment est-il utilisé en mise en cache ? Réponse : Redis est un magasin de clés-valeurs en mémoire utilisé pour la mise en cache, fournissant un accès rapide aux données fréquemment utilisées.
4. Comparez Redis et Memcached pour la mise en cache. Réponse : Redis prend en charge les structures de données et la persistance, tandis que Memcached est plus simple et plus rapide pour la mise en cache de base.
5. Qu'est-ce que le sharding dans les bases de données, et pourquoi est-il utilisé ? Réponse : Le sharding partitionne horizontalement les données sur plusieurs bases de données, utilisé pour la scalabilité et les performances dans les grands systèmes.
6. Comment Hibernate simplifie-t-il les interactions avec la base de données ? Réponse : Hibernate est un framework ORM qui mappe les classes Java aux tables de base de données, simplifiant les opérations CRUD.
7. Expliquez la mise en pool des connexions JDBC. Réponse : La mise en pool des connexions réutilise les connexions de base de données, améliorant les performances en réduisant la surcharge de création de connexions.
8. Qu'est-ce qu'une base de données de séries temporelles, et quand est-elle utilisée ? Réponse : Les bases de données de séries temporelles comme InfluxDB stockent des données horodatées, idéales pour la surveillance, l'IoT et les données de capteurs.
9. Décrivez les niveaux d'isolation des transactions dans les bases de données. Réponse : Les niveaux d'isolation (Non engagé, Engagé, Lecture répétable, Sérialisé) définissent comment les transactions interagissent les unes avec les autres.
10. Comment optimisez-vous les stratégies d'indexation dans les bases de données ? Réponse : Choisissez les index en fonction des motifs de requête, évitez la sur-indexation et utilisez des index composites pour les requêtes multi-colonnes.

Concurrence et Multithreading

1. Qu'est-ce qu'un deadlock en Java, et comment peut-il être évité ? Réponse : Un deadlock se produit lorsque des threads attendent indéfiniment les uns les autres. Il peut être évité en évitant les attentes circulaires et en utilisant des délais d'attente.
 2. Expliquez le framework Executor en Java. Réponse : Le framework Executor gère l'exécution des threads, fournissant des pools de threads et une planification des tâches.
 3. Quelle est la différence entre Callable et Runnable ? Réponse : Callable peut retourner un résultat et lancer des exceptions, tandis que Runnable ne peut pas, rendant Callable plus flexible pour les tâches retournant des résultats.
 4. Décrivez le modèle de mémoire Java. Réponse : Le modèle de mémoire Java définit comment les threads accèdent aux variables, assurant la visibilité et l'ordre des opérations à travers les processeurs.
 5. Qu'est-ce que le mot-clé volatile en Java, et quand doit-il être utilisé ? Réponse : Volatile assure que les modifications apportées à une variable sont visibles par tous les threads, utilisé dans les environnements multithread pour prévenir les problèmes de mise en cache.
 6. Comment empêchez-vous les conditions de course dans les applications multithread ? Réponse : Utilisez la synchronisation, les verrous ou les opérations atomiques pour assurer un accès exclusif aux ressources partagées.
 7. Expliquez le concept de verrou de lecture-écriture. Réponse : Les verrous de lecture-écriture permettent à plusieurs lecteurs ou à un seul écrivain, améliorant la concurrence en permettant un accès partagé.
 8. Qu'est-ce qu'un CountdownLatch, et comment est-il utilisé ? Réponse : CountdownLatch permet à un thread d'attendre qu'un ensemble de threads se termine, utilisé pour coordonner l'exécution des threads.
 9. Décrivez le stripage de verrou en Java. Réponse : Le stripage de verrou divise un verrou en plusieurs parties (bandes), permettant un accès concurrent à différentes parties, réduisant ainsi la contention.
 10. Comment gérez-vous l'interruption des threads en Java ? Réponse : Les threads peuvent vérifier l'état interrompu et lancer `InterruptedException`, permettant une terminaison élégante.
-

Serveurs Web et Équilibrage de Charge

1. À quoi Nginx est-il couramment utilisé ? Réponse : Nginx est utilisé comme serveur web, proxy inverse, équilibreur de charge et cache HTTP, connu pour ses performances élevées et sa scalabilité.

2. Expliquez la différence entre un équilibreur de charge et un proxy inverse. Réponse : Un équilibreur de charge distribue le trafic sur plusieurs serveurs, tandis qu'un proxy inverse transfère les requêtes aux serveurs backend, fournissant souvent un cache et une sécurité.
 3. Qu'est-ce que HAProxy, et pourquoi est-il utilisé ? Réponse : HAProxy est un équilibreur de charge et serveur proxy à haute disponibilité, utilisé pour gérer et distribuer les connexions réseau.
 4. Comment configurez-vous SSL/TLS sur un serveur web ? Réponse : SSL/TLS est configuré en obtenant des certificats et en configurant des écouteurs HTTPS, chiffrant les données en transit.
 5. Qu'est-ce que la mise en cache côté serveur, et comment est-elle mise en œuvre ? Réponse : La mise en cache côté serveur stocke les données fréquemment accédées en mémoire, mise en œuvre à l'aide d'outils comme Varnish ou Redis pour améliorer les performances.
 6. Expliquez l'importance de la journalisation dans les serveurs web. Réponse : La journalisation aide à surveiller l'activité du serveur, à résoudre les problèmes et à auditer la sécurité, en utilisant des outils comme ELK Stack pour l'analyse.
 7. Quelles sont les meilleures pratiques pour sécuriser les serveurs web ? Réponse : Les meilleures pratiques incluent l'utilisation d'en-têtes de sécurité, la mise à jour du logiciel et la configuration des pare-feu pour protéger contre les menaces.
 8. Comment gérez-vous la persistance des sessions dans l'équilibrage de charge ? Réponse : La persistance des sessions peut être réalisée à l'aide de sessions collantes ou de réplication de sessions, assurant que les sessions utilisateur restent cohérentes.
 9. Qu'est-ce que le déchargement SSL, et pourquoi est-il bénéfique ? Réponse : Le déchargement SSL déchiffre le trafic SSL/TLS au niveau de l'équilibreur de charge, réduisant la charge du serveur et améliorant les performances.
 10. Décrivez le processus de mise à l'échelle horizontale des serveurs web. Réponse : La mise à l'échelle horizontale consiste à ajouter plus de serveurs pour gérer une charge accrue, gérée par des équilibreurs de charge et des groupes de mise à l'échelle automatique.
-

CI/CD et DevOps

1. Qu'est-ce que GitOps, et en quoi diffère-t-il de la CI/CD traditionnelle ? Réponse : GitOps traite l'infrastructure comme du code, utilisant des dépôts Git pour gérer les configurations et les déploiements, mettant l'accent sur les définitions déclaratives.
2. Expliquez la stratégie de déploiement Blue/Green. Réponse : Le déploiement Blue/Green implique de faire fonctionner deux environnements identiques, en basculant le trafic vers le nouvel environnement après un déploiement réussi.

3. Qu'est-ce qu'un pipeline Jenkins, et comment est-il configuré ? Réponse : Un pipeline Jenkins est une série d'étapes pour construire, tester et déployer un logiciel, défini dans un Jenkinsfile en utilisant une syntaxe déclarative ou scriptée.
4. Comment mettez-vous en œuvre l'intégration continue dans un pipeline CI/CD ? Réponse : L'intégration continue automatise la construction et le test du code lors des commits, assurant que le code est toujours dans un état déployable.
5. Quel est le rôle de Docker dans la CI/CD ? Réponse : Les conteneurs Docker fournissent des environnements cohérents pour la construction, le test et le déploiement des applications, assurant la parité à travers les étapes.
6. Expliquez le concept d'Infrastructure as Code (IaC). Réponse : IaC gère l'infrastructure à l'aide de code, permettant le contrôle de version, l'automatisation et la cohérence dans les configurations d'environnement.
7. Quels sont les avantages de l'utilisation de Kubernetes dans la CI/CD ? Réponse : Kubernetes orchestre les applications conteneurisées, fournissant une scalabilité, une auto-guérison et des capacités de déploiement déclaratif.
8. Comment gérez-vous l'analyse de sécurité dans un pipeline CI/CD ? Réponse : Les outils d'analyse de sécurité comme SonarQube ou OWASP Dependency Check s'intègrent dans les pipelines pour détecter les vulnérabilités tôt.
9. Décrivez le processus de retour en arrière d'un déploiement échoué. Réponse : Les retours en arrière peuvent être automatisés à l'aide de contrôle de version ou d'outils CI/CD, revenant à une version stable connue en cas d'échec.
10. Quelle est l'importance de la gestion des environnements dans DevOps ? Réponse : La gestion des environnements assure la cohérence à travers le développement, les tests et la production, réduisant les problèmes spécifiques à l'environnement.

Modèles de Conception et Meilleures Pratiques

1. Qu'est-ce que le modèle Singleton, et quand doit-il être utilisé ? Réponse : Singleton assure qu'une classe n'a qu'une seule instance, utile pour gérer des ressources partagées comme les bases de données ou les paramètres de configuration.
2. Expliquez le modèle Factory et ses avantages. Réponse : Le modèle Factory fournit une interface pour créer des objets sans spécifier leurs classes, favorisant un faible couplage.
3. Qu'est-ce que le modèle Strategy, et comment favorise-t-il la flexibilité ? Réponse : Le modèle Strategy permet de sélectionner un algorithme au moment de l'exécution, permettant des changements de comportement flexibles sans modifier le code.

4. Décrivez les principes SOLID et leur signification. Réponse : Les principes SOLID (Responsabilité unique, Ouvert/Fermé, Substitution de Liskov, Ségrégation de l'interface, Inversion de dépendance) guident la conception pour un code maintenable et évolutif.
 5. Comment l'injection de dépendances améliore-t-elle la qualité du code ? Réponse : L'injection de dépendances réduit le couplage en externalisant la création d'objets, rendant le code plus modulaire et testable.
 6. Qu'est-ce que l'événement sourcing, et en quoi diffère-t-il du stockage de données traditionnel ? Réponse : L'événement sourcing stocke une séquence d'événements qui décrivent les changements d'état, permettant la reconstruction de l'état et les audits de traçabilité.
 7. Expliquez le modèle d'architecture CQRS. Réponse : CQRS sépare les commandes (opérations d'écriture) et les requêtes (opérations de lecture), optimisant les préoccupations d'écriture et de lecture séparément.
 8. Quelles sont les meilleures pratiques pour la refactorisation du code ? Réponse : Les meilleures pratiques incluent des changements petits et incrémentiels, le maintien des tests et l'utilisation d'outils pour les refactorisations automatisées.
 9. Comment assurez-vous les pratiques de code propre ? Réponse : Les pratiques de code propre incluent des noms significatifs, le respect des normes et l'écriture de code auto-documenté.
 10. Quelle est l'importance du TDD (Développement Piloté par les Tests) ? Réponse : Le TDD consiste à écrire des tests avant le code, assurant que le code répond aux exigences et améliorant la maintenabilité par des tests continus.
-

Sécurité

1. Qu'est-ce que OAuth2, et comment est-il utilisé pour l'autorisation ? Réponse : OAuth2 est un cadre d'autorisation permettant aux applications tierces d'accéder aux ressources sans partager les informations d'identification.
2. Expliquez les JWT (JSON Web Tokens) et leur rôle dans la sécurité. Réponse : JWT fournit un moyen compact et auto-contenu de transmettre de manière sécurisée des informations entre les parties, utilisé pour l'authentification et l'échange d'informations.
3. Qu'est-ce que le RBAC, et comment simplifie-t-il le contrôle d'accès ? Réponse : Le contrôle d'accès basé sur les rôles attribue des permissions aux rôles, simplifiant la gestion de l'accès des utilisateurs en attribuant des rôles aux utilisateurs.
4. Comment empêchez-vous les attaques par injection SQL ? Réponse : Utilisez des instructions préparées et des requêtes paramétrées pour séparer le code et les données, empêchant l'exécution de SQL malveillant.

5. Qu'est-ce que le XSS (Cross-Site Scripting), et comment peut-il être empêché ? Réponse : Le XSS permet aux attaquants d'injecter des scripts dans les pages web ; il peut être empêché en nettoyant les entrées et les sorties et en utilisant des en-têtes de sécurité.
6. Expliquez l'importance du chiffrement dans la sécurité des données. Réponse : Le chiffrement protège la confidentialité des données en les convertissant en un format illisible, assurant que seules les parties autorisées peuvent y accéder.
7. Quelles sont les meilleures pratiques pour la programmation sécurisée en Java ? Réponse : Les pratiques incluent la validation des entrées, l'utilisation de bibliothèques sécurisées et le respect des directives de sécurité comme OWASP.
8. Comment mettez-vous en œuvre des journaux d'audit dans les applications ? Réponse : Les journaux d'audit enregistrent les actions des utilisateurs et les événements système, fournissant une visibilité et une responsabilité pour la sécurité et la conformité.
9. Qu'est-ce que l'authentification à deux facteurs, et pourquoi est-elle importante ? Réponse : L'authentification à deux facteurs ajoute une couche de sécurité supplémentaire en nécessitant deux formes de vérification, réduisant les risques d'accès non autorisé.
10. Décrivez le rôle d'un pare-feu d'application web (WAF). Réponse : Un WAF protège les applications web contre les attaques comme l'injection SQL et le XSS en filtrant et en surveillant le trafic HTTP.