

Java バックエンドエンジニア：面接質問集

プロジェクト関連

1. 自己紹介
2. プロジェクト紹介
3. プロジェクトの規模
4. スレッドと並行処理の知識ポイント
5. lock と synchronized の違い
6. MYSQL と MONGODB の違い
7. 使用している SpringCloud のコンポーネントとバージョン

Spring エコシステム

1. SpringBoot のコアとは何か
2. SpringBoot のバージョン
3. 使用されている SpringCloud コンポーネント
4. SpringBoot と SpringCloud の違い
5. Spring の bean 注入方法
6. 複数データソース設定における Bean の命名規則

マイクロサービスアーキテクチャ

1. マイクロサービスモジュールの分割方法
2. 結合において、ビジネス結合と技術結合のどちらを優先するか
3. 使用されているマイクロサービスのパターン
4. API 設計におけるステートレスとステートフルの違い、選択方法
5. クラウド展開におけるデフォルトのノード数

Java 関連

1. 使用中の JDK バージョン
2. Stream の内部実装
3. Stream コードの問題をデバッグする方法
4. Stream のコンポーネントと内部構造

5. スレッド間でメッセージを共有する方法
6. 再入可能ロックとは何か
7. ロックについての理解を説明してください
8. JDK8 と JDK17 の違い

サーバー設定

1. Tomcat のデフォルトスレッド数
2. Tomcat から他の Web サーバーへの移行方法
3. Nginx リバースプロキシの設定

データベースとキャッシュ

1. プロジェクトで Redis 接続を作成する方法、必要な依存関係
2. Jedis を使用して接続を作成する方法
3. Redis のトランザクションコマンド
4. SQL シナリオ: 部門と従業員テーブル間の結合クエリで、部門ごとの最高給与を取得
5. 左結合の利点

プログラミングシナリオ

1. Java8 API を使用して、年齢と名前による Person オブジェクトのソートを実装する
2. 2つの文字列間に共通する文字を出力する

CI/CD

1. Dockerfile の記述経験
2. Jenkins パイプラインにおける Groovy クロージャの機能
3. Nexus ツールの目的
4. DevOps の実践経験

コアコンセプト

1. Java メモリモデル (JMM) の概念と機能
2. HashMap の実装原理と衝突解決

3. プロキシパターンの実装（静的および動的）
4. IoC と AOP の原理
5. Redis がシングルスレッドで高いパフォーマンスを維持する理由
6. Executor スレッドプールのパラメータ設定

インドの面接官が重視するポイント

1. Java の基本概念（例を用いて）
2. ポリモーフィズムの使用法
3. LinkedList のパフォーマンスと違い
4. Map の違い
5. 配列の取り扱いに関する解決策
6. Linux コマンド
7. プロジェクト経験（技術スタック: Python, Angular, Control-M）
8. 英語の能力

英語の流暢さのレベル

1. 自己表現ができない - 基礎が弱く、自己紹介やプロジェクトの説明ができない
2. 基本的/簡単な会話 - 基礎はあるが、準備した自己紹介はできる
3. 日常的なコミュニケーション - 基礎がしっかりしており、英語の会議に参加して内容の大部分を理解できる
4. 実用的 - 基礎が強く、面接官と流暢にコミュニケーションが取れる
5. 実用的かつ技術的な交流 - 優れた英語力を持ち、技術的な議論を障壁なく行える

その他の面接評価ポイント

1. 学習方法
2. 技術コミュニケーション能力
3. クライアントとのコミュニケーション経験
4. プロジェクトの課題と解決策
5. キャリア開発計画
6. 技術的深さ（マルチスレッド、データ検証、フロントエンド開発など）