

Prepare Interview

제가 한 인터뷰 경험들이 도움이 됐으면 좋겠네요.

인터뷰 한 회사 리스트들

- 카카오 엔터프라이즈
- 당근마켓
- 카카오 커머스
- 우아한 형제들
- 라인 게임즈
- 클래스 101
- 스타트업 - 빅피쳐 인터랙티브
- Ground X
- 네이버 웹툰

인터뷰에서 직전에 준비해야 할 마음가짐

- 천천히 말하는게 좋다. 그래야 듣는 사람이 좀 더 집중을 잘하고 잘 듣는다.
- 모르겠으면 모르겠다고 말하는게 좋다. 개소리하다가 공격 들어옴.
- 임기응변을 어떻게 할지 미리 생각해두면 마음이 편안해 질 수 있다.
- 나같은 경우는 질문에 제대로 대답 하지 못하면 자신감이 급떨어지는데 그런 모습을 보여주는것 자체가 별로 좋게 생각하는 것 같지 않아서 자신감 있는 모습 그대로 보여주려고 노력하는게 좋은 것 같다.

전화 인터뷰

- 주로 하는 질문은 기술 스택에 관한 질문들, CS 에 관한 질문들이 많음.
- 내 생각인데 질문 리스트가 몇백개 있는듯. 그 중에 골라서 질문하는 것 같음.
- Java 에 관한 질문들
 - Java 8, Java 11 에 들어온 기능들
 - Java 에서 String 은 어떻게 관리하는지?
 - Java Wrapper 클래스를 사용하는 이유와 JVM 내에 어디에 저장되는지?
 - Java 에서 Concurrency 를 해결하는 방법들에 대해서.
 - JVM 의 Execution Engine 에 대해서 물어봄.
 - JVM 내의 메모리 구조에 대해서 물어봄.
 - GC 알고리즘이 무엇이고 어떻게 코딩하면 GC 가 많이 생길 것 같은지
 - equals() 와 hashCode() 차이?
 - HashMap 과 Hashtable 차이?
 - ConcurrentHashMap vs Hashtable 차이? (Java Collection 에 대해 마스터 해야함.)

- Annotation 의 Retention 에 대해서 알고있는가?
- Lombok 의 에노테이션은 Retention 이 무엇인가?
- Spring 에 관한 질문들
 - DI (Dependency Injection) 이란? 이걸 하면 뭐가 좋은지?
 - Spring 에서 IoC Container 가 하는 일은?
 - AOP 를 사용할 수 있는 방법은?
 - Spring AOP 와 AspectJ AOP 의 차이점은?
 - Spring AOP 의 self-invocation() 문제와 이를 해결하는 방법은?
 - Spring 에서 Bean Scope 은 무엇이 있는지?
 - Spring 에서 Bean Lifecycle 에 관련된 이벤트 함수를 줄 수 있는 방법은?
 - @Autowired 말고 의존성 주입을 하는 방법은?
 - Spring 에서 Transaction 이란?
 - Transaction 을 어떻게 사용했는지 예시 보여주셈
 - Spring 과 Spring Boot 의 차이점은?
 - JPA N + 1 문제를 해결하는 방법은?
 - JPA Entity 의 상태는 무엇이 있는지?
 - JPA 양방향 관계와 단방향 관계를 각각 언제 사용하는지?
- CS 에 관한 질문들
 - 주로 Network, Database, OS 와 연관된 질문들을 하게됨.
 - Design Pattern
 - 주로 사용하는 디자인 패턴이나 알고리즘은?
 - 왜 이 디자인 패턴을 주로 사용했고 어디에 사용하는지?
 - Network
 - 웹 요청을 날리면 응답이 나갈때 까지의 과정이 어떻게 되는가? (대충 이런식으로 말하면 됨. Spring Boot 기준으로 DNS -> TCP -> WAS -> Filter -> DispatcherServlet -> Interceptor -> Handler -> Interceptor -> ViewResolver -> Filter)
 - HTTP, HTTPS, HTTP2, HTTP3 까지 알아두면 좋음
 - HTTP Method 종류도 물어봄. Header 까지 알고 있으면 좋을듯.
 - Database
 - 데이터베이스 질문하면 Index 질문 무조건 함.
 - Index 왜 씀?
 - Index scan 보다 Table scan 이 빠른 상황과 그 이유는?

- Clustering Index vs Nonclustering Index
 - B-Tree Index vs HashTable Index
 - 어떤 칼럼을 인덱스로 둘것인지?
- 실행 계획 질문도 함.
 - 실행 계획 어떻게 하니?
 - 실행 계획 보면 뭐가 나오니?
 - 실행 계획으로 쿼리 튜닝 어떻게 하니?
- Primary Key 와 Unique Key 의 차이는?
- Cross Join, Outer Join, Inner Join 에 대해 물어보는 경우도 있었음.
- 정규화 질문도 자주 함.
 - 제 1 정규화 ~ 보이스 콜드 정규화까지 알고있으면 됨.
- 트랜잭션 Isolation 질문도 자주 함.
 - Read Committed, Repeatable Read, Serialization 을 이해하고 동시성 문제들 (Dirty Read, Dirty Write, Lost Update, Write Skew, Phantom 문제에 대해 이해하고 있으면 좋을 듯.)
- Operating System
 - Process 와 Thread 차이 질문 가끔 함
 - Context Switching 에 대해서
 - System Call 의 동작 원리는?
 - OS 에서 Memory 관리는 어떻게 하는지?
- etc
 - Blocking 과 NonBlocking, sync 와 async 의 차이는?
 - Parallel 과 Concurrency 의 차이는?

직무 인터뷰

- 직무 인터뷰는 회사마다 조금씩 다르다. 이 차이를 알고가자.
- 기획자님이 면접에 들어오는 경우가 가끔 있어서 지원 요건에 있는 업무를 파악하고 내가 어떤 비즈니스 문제를 해결할 수 있을지 생각해둬야 한다.
- 직무 인터뷰는 인터뷰 당시에 준비하기 보다는 평소에 어떻게 개발을 해왔는지가 더 중요한거 같다. 평소에 꼼꼼하게 그리고 대충 개발하지 않았더라면 충분히 잘 대답할 수 있는 듯.
- 이력서 기반의 면접이 주로 있음.

- 대충 서비스를 만들었는지 얼마나 고민해서 서비스를 만들었는지의 질문이 주로 들어옴. 이걸 평소에 잘하면 준비 안해도됨.
 - 내 경험으로는 회사 들어가면 개발 일지를 매일 매일 작성해두면 도움이 많이 될 듯. 자신의 고민이 다 들어가있을거니까. 얼마큼 고민하면서 서비스를 만드느냐가 중요한 거 같음.
- 문제를 주고 문제 해결을 물어보는 질문
 - 빠빠로 데이 이벤트를 할려고 한다. 이때 트래픽이 폭주 할 수 있을 거 같은데 절대 장애가 나면 안된다. 어떻게 해결할 것인가?
 - 거래내역을 데이터베이스에 저장하고 싶은데 이를 동시성 문제 없이 안전하게 저장 할려면 어떻게 해야할까?
 - 사용자들을 Redis 캐시에 저장하고 싶은데 어떻게 캐싱하면 좋을까?
 - 알정이 촉박한데 코드 리뷰와 테스트 코드를 짤 시간이 부족하다. 어떻게 해결할 것인가?
- 기술 스택과 관련된 질문
 - 기술 원리에 대한 질문보다 기술을 정확하게 이해하고 사용하는 지에 대한 질문을 주로 함. 그래서 한번만 사용한 기술 스택 적었으면 털릴 수 있음. 면접관들은 이거에 대해 익숙하다고 생각하고 질문 들어옴.
 - 주로 사용한 기술이 Redis 인 경우
 - Redis 캐싱 전략은 어떻게 되는가?
 - 캐시에 어떤 데이터를 넣어놨는가?
 - 왜 Memcached 가 아닌 Redis 를 선택했는가?
 - 주로 사용한 기술이 Kafka 인 경우
 - 카프카를 도입한 이유가 무엇인가?
 - RabbitMQ vs Kafka 차이는 뭐라고 생각하는가?
 - 카프카에서 오로지 딱 한번만 메시지를 처리하려면 어떻게 해야하는가?
 - 카프카에서 중복 메시지를 보내지 않으려면 어떻게 해야하는가?
- 라이브 코딩 테스트를 보는 경우도 있음
 - 코딩 습관을 보기 위해 주로 이 방법을 사용하는 듯.
 - Leetcode 문제를 많이 가져옴. 난이도는 easy 로.
 - 데이터베이스 테이블 설계를 직접 하는 경우도 있었음.
 - 멀티 스레드에서 동시성 문제를 해결하는 문제도 만나봄
 - if-else 를 enum 을 가지고 해결하는 refactoring 문제도 만나봄.

- 거의 무조건 들어오는 질문들
 - 이력서 별로 특이한 게 있다면 무조건 질문 들어올거임.
 - 왜 지원했는가?
 - 자신의 장점과 단점은?
 - 인성과 관련된 유도신문 많이 들어옴. (평소때 올바른 사상 가지고 있으면 문제 없을 듯)
 - 개발자 마인드에 대해서도 자주 질문 들어옴 (꾸준함 + 성장 하고 싶은 마인드 위주로 가는게 젤 좋음. 말 뿐이 아니라 증명할 수 있는게 있어야 함.)
 - 지원자님도 이제 질문하세요! (질문할 거 2개정도만 들고가는게 젤 나은듯)

임원 인터뷰 or 컬처 핏 인터뷰

- 회사의 인재상과 맞는지, 회사에 잘 적응할 수 있는지를 보는 면접이라고 생각하면 된다. 자기 생각을 그대로 말하면 되는 면접이기도 하지만 그러면 우리와 맞지 않다고 생각해 떨어질 수 있다. 무조건 들어가고 싶다면 회사의 인재상을 조사하고 면접 들어가면 좋을듯. 이런 인재상은 회사 홈페이지에 있는 경우가 많음
ex) [Amazon Leadership Principle](#)
- 받고싶은 연봉도 물어볼 수 있음.
- 지원자가 이 회사에 그렇게 관심있다고 생각들지 않으면 뽑지 않는듯.

나의 경험으로 보는 인터뷰들

- 내가 왜 안뽑히지? 거의 무조건 뽑히겠다 -> 뽑힐 가능성 95%
- 하 ㅅㅂ 여기서 실수좀 했는데 근데 이걸 답변 좀 잘했다. -> 안 뽑힐 가능성 80%
- ㅂㅅ ㅈ 했다. -> 안 뽑힐 가능성 100%
- 인터뷰도 경험치를 먹으면 더 잘해져서 중요하지 않은 인터뷰를 앞에서 보고 중요한 인터뷰를 뒤로 미루는게 더 좋은듯.
- Github 와 기술 블로그를 잘 정리 했다면 알게 모르게 가산점이 있는듯.
- 인터뷰 보는 일정을 자신이 컨디션 좋은 시간으로 잡는게 좋은 것 같다. 피곤하면 제대로 답변도 못함.
- 정확하게 대답하기 어려운 경우가 가끔씩 있는데 그냥 모른다고 넘어가기 보다는 자신이 알고 있는 선에서 대답하는게 더 나은듯.

- 인터뷰 하는 팀마다 인터뷰가 더 잘될수도 있고 더 안될수도 있는듯. 나한테 더 잘맞는 팀이 있는듯.