

美团面试经历！

程序员面试 6月6日



微信扫一扫
关注该公众号

美团我是在拉勾网上投的简历，之前也投过一次，简历都没通过删选，后来让学姐帮我改了一下简历，重新投另一个部门，获得了面试机会。10月23日中午HR打电话过来预约了下午4点半面试，说会在线写代码，让我准备好网络环境。结果5点半还没打电话过来，被放鸽子。与hr重新沟通过后，确定下周一下午再面，可是跟hr沟通预约这一套貌似在美团并没有什么用。

美团技术一面20分钟

晚7点，因为想到下周才面试，我刚准备出去打个羽毛球，北京的电话就来了。面试官各种抱歉，说开会拖延了。

1、自我介绍

说了很多遍了，很流畅捡重点介绍完。

2、问我数据结构算法好不好

挺好的（其实心还是有点虚，不过最近刷了很多题也只能壮着胆子充胖子了）

3、找到单链表的三等分点，如果单链表是有环的呢

用快慢指针，一个走一步，一个走三步。如果有环，先判断环在哪，找最后一个节点，然后用之前的无环的做法

4、讲一下项目的架构

我重点讲了MVC

5、说一下你熟悉的设计模式

我重点讲了单例、工厂方法、代理

5、有没有配置过服务器啥啥啥

这个我真不知道，都没听过呢，只能诚实说没有，毕竟都没法扯。

一面挺匆忙的，我估计面试官刚开完会还没吃饭呢。他说让我等，可能再找一个同事面我，可能就直接告诉我结果了。从一面面试官的声音和口吻，我判断他一定是个部门老大，问的设计偏多，后面hr告诉我他就是我要去的部门的老大。哈哈。

美团技术二面60分钟，详细问答

面完一面正准备出去打羽毛球，北京的电话又来了。（注定这周五参加不了球队活动了！）二面：跟一面比起来，二面面试官的声音听起来就像是搞技术开发的，果不其然，一个小时的纯技术电话面试！面的特别全面！

1、Spring：有没有用过Spring，Spring IOC、AOP机制与实现，Spring MVC

其实我挺不想被问到Spring的细节的，框架这些我都没有复习不太记得了。所以我对面试官说Spring里面的一些比较重要的机制我理解的还不错，然后我用一个实际的例子把我对IOC、AOP理解讲了一下，他听了说对，理解的不错（难得遇到一个边面试边能给反馈的面试官，好开心）

Spring MVC其实我用过，我就对面试官讲了我的项目中用到的Servlet，jsp和javaBean实现的MVC，以及MVC各个模块职责以及每个模块是怎么联系到一起的，最后我补充了一句我想SpringMVC的思想其实跟这个是一样的（他说对的，嘿嘿有反馈真好）

2、多线程：怎么实现线程安全，各个实现方法有什么区别，volatile关键字的使用，可重入锁的理解，Synchronized是不是可重入锁

这里我就主要讲了Synchronized关键字，还有并发包下面的一些锁，以及各自的优缺点和区别。volatile关键字我主要从可见性、原子性和禁止JVM指令重排序三个方面讲的，再讲了一下我在多线程的单例模式double-check中用到volatile关键字禁止JVM指令重排优化。

3、集合：HashMap底层实现，怎么实现HashMap线程安全

我讲了一下HashMap底层是数组加单链表实现，Node内部类，add的过程，Hash冲突解决办法，扩容，三种集合视图。HashMap线程安全的实现方式主要讲了HashTable、ConcurrentHashMap以及Collections中的静态方法SynchronizedMap可以对HashMap进行封装。以及这三种方式的区别，效率表现。

4、JVM内存管理，GC算法，HotSpot里面的垃圾回收器、类加载

JVM内存主要分为五个区，哪些是线程共享的，哪些是线程独享的，每个区存放什么。GC方面：怎么判断哪些对象需要被GC，GC的方法，Minor GC与Full GC。HotSpot GC算法以及7种垃圾回收期，主要讲了CMS和G1收集器。类加载：类加载的过程，Bootstrap classloader-ExtClassloader-AppClassloader，父类委托机制。

5、进程和线程的区别

从调度、并发性、拥有的资源和系统开销四个方面回答的。

6、HTTP有没有状态，我说无状态，怎么解决HTTP无状态 怎么解决HTTP无状态其实是怎么进行会话跟踪，有四种方法：URL重写、隐藏表单域、Cookie、Session。

7、Java IO，NIO，Java中有没有实现异步IO

Java IO实现的是同步阻塞，它是怎么实现同步阻塞的。我拿了read()方法举例来讲的。NIO实现的是同步非阻塞，我详细讲了一下Selector中的select()方法轮询说明它是如何实现多路复用IO的。然后对比了一下他们的效率。面试官可能看我这一块比较了解，又继续问我Java中有没有实现异步IO，我感觉好像没有，但面试官说有，让我想想，其实这里我并不清楚啦，所以我就对面试官讲了一下我对Unix中异步IO模型的理解，然后说至于Java里面有没有我真的不太清楚。（他居然笑了！说你理解是对的，Java里面有没有不重要！哈哈）

8、前端会不会，Ajax是什么，Ajax实现原理

前端我只是会用一些js而已，用过jquery框架，问我Ajax全称是啥，我猜是异步的js和xml。Ajax实现原理其实我也不懂，我就只简单讲了一下它通过XMLHttpRequest对象进行异步查询，Ajax引擎在客户端运行，减少了服务器工作量。

9、让我设计一个线程池

因为我简历中有写到我对多线程、并发这一块理解比较好。所以他老问这方面的题。这个问题因为我之前看过ThreadPoolExecutor的源代码，所以我就仿照那个类的设计思路来想的，详细讲了一下核心池、创建线程可以用工厂方法模式来进行设计。线程池状态、阻塞队列、拒绝策略这几个方面。设计的还算比较周全。

10、讲几个设计模式，哪些地方用到了，为什么要用

单例模式，jdk中的getRuntime()；工厂方法模式，ThreadPoolExcutor用到ThreadFactory；观察者模式：java.util包下面的Observable和Observer。最后主要讲了一下工厂方法模式的使用场景。

11、Mysql优化、索引的实现

我从数据库设计优化和查询优化两方面讲的。索引B+树实现，InnoDB和MyISAM主键索引的实现区别，一个聚集一个非聚集。

12、事务的隔离级别

四种隔离级别，可能会出现哪些异常，mysql中默认级别。

13、有没有用过Hibernate、mybatis、git

这个简单讲一下就好，分别是干什么的

14、Linux

我说这个本科学过，但是很久没用，命令忘光了。他说没事，考你几个简单的：cd、ls、dir（真的是简单的）

15、算法题

1、从10万个数中找最小的10个，时间复杂度分析（最大堆，考虑内存）2、从一个有正有负数组中找连续子数组的最大和，时间复杂度分析（动态规划）3、满二叉树第i层有多少个节点，n层的满二叉树共有多少个节点

终于到我提问环节了

1、你们是什么部门（ he说是核心部门，大数据研发）2、我对高并发和负载均衡挺有兴趣的，但是我平时在学校也没有这个环境让我在这方面有所体验，那你建议我目前可以怎么学呢（他说这确实是不太好学，只能看些理论和别人的博客，以后工作中才能慢慢学）3、中间件具体是做什么的，是解决高并发和负载均衡吗（他说差不多是的，然后他说我们这个部门不是中间件，是大数据部门啊，我说恩我知道）

最后没啥问题了，他让我保持电话畅通。

这一面面完，口干舌燥，我一度怀疑他可能不知道我是在应聘实习生的岗位。有太多要总结的了，放在总结的地方一起讲吧。

美团技术三面25分钟

面试官说是他是另外一个部门的，需要进行交叉面试。

1、MySQL优化2、说下项目做了些什么，架构之类的3、在collabedit上在线写代码，题目很简单是编程之美上的原题，一个有序的整数数组，输出两个数，使它们的和为某个给定的值。之前做过很快写好，然后给他讲思路。他继续问如果数组无序怎么办，先排序。4、两个文件，每个文件中都有若干个url，找出两个文件中相同的url（用HashMap）

这一面挺简单的，只是增加之前面没有过的在线写代码环节，collabedit后来我才了解，像facebook一些互联网公司远程面试都会用这个在线编辑器写代码，就是文本文档写，没有提示，不能编译运行，跟白板写一样。平时练练手就好。

美团技术HR四面30分钟

三面面试官说他那就是终面，说我过了等hr联系我。万万没想到半小时后的hr面居然也是技术。

1、自我介绍，都四面了还自我介绍？！我还以为是单纯的hr面，所以介绍的都是我的性格和生活方面的，结果并不是。2、问项目，问的特别特别细，技术细节，还有遇到什么问题，怎么解决的，做项目有没有人带，怎么跟别人沟通的。3、数据库优化，如果数据库一个表特别大怎么办 数据库优化我就讲了之前讲过很多遍的点，他问一个表特别大怎么办：大表分小表，怎么实现：使用分区表4、问研究生的科研题目，为什么选这个题，看了些什么论文（细到问我那些论文是发表在什么期刊上的，作者是哪所学校的），为什么要选这个算法，怎么优化的，实验结果怎么跟别人作对比的，为什么比别人的算法好（一个问题不断砸过来，我猜我说的那些专业名词他应该不太懂，只是判断一下是不是我做的而已。。。）5、确定实习时间 这一面确实让我感到有压迫感，项目是本科做的，挺久了，一些技术细节上也没太总结，所以问细了我只能连编带编，嘿嘿。科研方面倒还好，上个学期都在弄这个，一些算法的实现和改进、对比都还记得比较清楚，回答的挺流畅的，可能这真的就是技术型的hr面吧。

作者：只爱羽毛球的程序媛

blog.csdn.net/csuliyajin2012/article/details/49430659

长按订阅更多面经分享

