**Coding Challenge**

Manipulate this scheduling data to find the best conference times for as many employees as possible.

Make a call to a given endpoint to get a bunch of dates, sort the dates, send the new sort to a different endpoint. Their response message does not give any hints on why your answer is not right so make to review your code carefully.

Get a json from url, process json based on a requirement. Requirement could be changed for each person, but mine was group by most popular date that was the start of a span of a period of time. Then format to json as given in example and post back. Bit more work then that but that is basically it.

You must hit a unique api to receive json data, modify and analyze the data, and then generate json in a specific format and post the data back to a different url. If the data was formatted correctly it responded with a 200 and you were told to zip up your code and upload it.

Create an API from scratch that takes user names, emails, one country they could attend and what dates they could attend that country. Find which dates for each country that the most people could attend in two consecutive days and return the dates as strings that they would be attending them. POST the countries with the dates as JSON properties to their server until you get a status of 200. The resulting JSON they're looking for does not match the example result provided.

**Interview Questions**

* See other reviews, pretty spot on. JSON API assessment. URL shortener, LC medium, social media SQL DB design.

First, it's no secret they ask the same questions and apparently have done so for at least 4+ years (see other Glassdoor interview reviews). I will state the questions that you will be asked fairly definitively so that anyone else who "fails" their interview can help shed light on what the bias actually is in their process: 1) Your first test will be to make an API call to their server (I included the code to this question in my response) 2)You will get a behavioral interview where as a reviewer (on blind) stated they would like to hear about their powerpoint slide 3) You will do an onsite "interview" (read: memorization and implicit bias screening) where you will be asked the following: I. Implement a database for a social network (it is literally the database from Grokking the system design interview for "Designing Facebook’s Newsfeed", learn how to do the join operation to fetch all posts for a user's friends and groups they are following. They will tell you knowing syntax is not important but it will be brought up as necessary if they want to ding you based on the implicit bias). II. Implement a URL shortener (again literally as from Grokking the system design interview) III. something akin to a leetcode medium but it seems even these are heavily recycled problems I don't believe they stray from asking ( leetcode top-k-frequent-words or merge n sorted arrays into one) but I believe if there is any room for variance it would be this question (be at a level of leetcode mediums to be safe). My actual advice would be if you fail this interview please don't think of it as a learning process. I literally answered all of their questions correctly and was told "there were red flags with some of your responses". My recruiter went to bat for me because I stated that can't be true since these answers were literally textbook. Second response: "sometimes correct answers aren't enough we need to know how you got there a little more".

Manipulation of json data that makes an API call. Leet code. JS fundamentals.

Started with a Phone Screen with a Recruiter. Next was Project based Technical Question was getting data from one Endpoint and Transforming it and POSTing it to another Last was a Virtual Onsite with 4 interviewers After my interview I was told I wasn't selected for this position, but they were interviewing for another Software Engineering role if I wanted to do another virtual onsite. I declined as the timing wouldn't of worked for me.

**Interview Questions**

* Design an SQL Schema and Queries.
* Merge two lists in ascending order to a new list with a max length of K.

The final interview took place over multiple back-to-back Zoom sessions (4-5 hours) with 6-8 different interviewers. In summary, the different parts were something like: - 1st technical portion (some coding related to general JS knowledge) - Discuss previous work experience/projects with Engineering Lead - 2nd technical portion (more in-depth coding, problem solving, involved some data structure/complexity knowledge) - Product-related prompt (based on a doc I received a few days prior to prep). This portion was unique compared to other interviews and I liked it a lot, as it gave me insight into the kind of work and gave interviewers insight into the way I think and approach problems in more than a purely technical way. - Q/A wrap up with recruiter All my interviewers were great and personalities ranged from generally friendly to very friendly and enthusiastic.

**Interview Questions**

* Merge sort, JS prototypes, currying

**Interview**

I found their job posting online and was emailed a coding challenge a few days later. You were given three hours to complete the following problem: You must hit a unique api to receive json data, modify and analyze the data, and then generate json in a specific format and post the data back to a different url. If the data was formatted correctly it responded with a 200 and you were told to zip up your code and upload it. After the coding challenge I was asked to schedule a phone screen with a recruiter. They asked common questions about my experience, why hubspot, why \_\_\_\_, etc. At the end of the phone screen I was told the next step if I make it will be a final round on site interview, and that they will let me know of their decision in a week or so. I ended up getting to the final round interview. It was four parts. First part was a a tour of their office, second was a technical interview with a CS question about n-grams that could be programmed in any language of my choosing. The third interview was half behavioral half javascript specific, focusing on prototype inheritance, closures, etc. The last section was a wrap up about my interview experience and any suggestions, salary expectations, other outstanding offers, etc. A few days later I was given a phone call and given an offer that exceeded my expectations

**Interview Questions**

* Given a string s, and a number n, return the most common n-gram in s
* If I would like to add a function to the string prototype, how would I do it?
* Write an API Call to our servers
* Design a URL shortener service
* Merge two lists in ascending order to a new list with a max length of K.

[**Answer Question**](https://www.glassdoor.com/Interview/Merge-two-lists-in-ascending-order-to-a-new-list-with-a-max-length-of-K-QTN_3532283.htm)

* Design question involving a URL shortener, but then scaling it.

**Interview Questions**

How would you design a Facebook clone? (database related)

**Interview**

Initiated the process through an employee referral and the initial phone screen was quite thorough and really helped me evaluate what the company was doing and whereabouts it was at in terms of its product and their objectives. Recruiter was very down to earth. After the phone screen, the HubSpot coding assessment was setup. You simply follow a link to kick things off. Get familiar with the instructions. It's helpful to have a skeleton application ready to go beforehand. I won't divulge too much on the content as I believe it is a fair assessment of developer skills and the HubSpot team did put a good bit of thought into it. I decided it would be prudent to do this on a Saturday morning as the code targets an API using a key so no requirement to have anyone on the HubSpot side at this stage and it allows for a block of time to focus on it without distractions. At times I did feel quite pressured by it as I changed my approach twice in the first hour.

Nevertheless, completed the assessment in roughly 2.5 hours, received a confirmation mail straight away. You have however much time you need to clean the code prior to submitting the package. On-site was arranged a couple of days later. Standard procedure here. They provided a full breakdown of the schedule for the day. All in all it ran from 13:45 to 17:00. For anyone who hasn't done extensive on-sites, it is best to arrive a little early and get a feel for the environment and a drink plenty of water. Your brain needs it!

First round was heavily focused on a relational database schema and walking through queries of average complexity. This was essentially social media connections and how to model them and relationships between them. I felt a good momentum with this round but fell a little short at the very end with some of the more exotic queries. Most people should get the majority of these questions even with a little experience with MySQL, MSSQL, Sybase, Postgres, etc. The preferred solution here was to double up records which initially confused me as I found it to be quite an expensive way to handle this situation, although I could not get over my own mental hump and fully figure out my own solution without tripping up. Think complexity versus simplicity. Both interviewers were quite helpful however and really it felt more like a technical discussion one would have during their normal day to day. The very final portion of this round related to indexing tables and how to optimize queries. These would would be the bar raiser questions. Second round was a high level system design discussion for a URL shortening service. Basically, TinyURL after a nice chat covering a few laid back situational questions. I enjoyed this one quite a lot. Actually learned quite a few tips from the interviewer here and my suggestion to breaking certain parts of the system down into micro-services was well received. Know concepts such as load balancing, caching, database replication, redundancy, CAP and a couple of other key concepts relating to well scaled distributed systems. This is quite a broad topic so be careful not to over rehearse some predefined answers because one criticism and your whole plan will fall apart. Learn and understand the concepts and the interviewer will guide the discussion based on your understanding. The key with these types of interviews is to assess how you deconstruct a much larger problem. So ask lots of questions and get a very clear definition of what it is the system should do, then define the API and work from there. I contradicted myself a little here when it came to SQL vs NoSQL. Third and final round was a one to one with the hiring manager. A good few situational questions here like dealing with conflicts etc. There are plenty of examples of what comes up during these types of interviews. No real conversational flow to the questions though in this round and I didn't feel as relaxed as the previous rounds but can't really say for sure how that impacted the final decision as sometimes it is really more formality and hiring managers just want to see if you can work with the team and fit in and ultimately not be a pain in the backside for them. After that, I was given a tour of the office. Quite a nice space they have and really has that start up vibe. Overall, I quite enjoyed the experience and it gave me a very good chance to correct my shortcomings. For junior-mid level positions, if you really want to work here, practice your A game and start white-boarding solutions as the common trend for these on-sites nowadays is to ask easy questions to ease you in, then intermediate, then more complex, then basically the bar-raising questions that get you the gold stars if you can answer them. My one critique of the whole process: the role was Java focused and yet the topic was hardly covered at all, nor any programming concepts for that matter. If you like what HubSpot is doing, go for it! :)

**Interview Questions**

* How would you model friendships and relationships in a database?
* How would you design a URL shortening service?
* How and why would you use indexing on a table
* What is a foreign key?

**Interview Questions**

* What technolgies have you incorporated within a team?
* Tell me about a time you were behind on a deadline?
* How would you handle having a proposal for a design or new technology rejected?
* Whiteboard: Design a URL shortener API, with metrics capturing and some optimisation suggestions.
* Whiteboard: Design a data model for storing users, friend lists, and posts (small messages). Design an API for fetching a paged list of time-ordered posts from a user's friends (like a lightweight twitter)
* Merge two sorted list in an array of max length k

**Interview**

After referral, instant phone chat. Online exam went well and was easy enough. After exam was invited and flown into Cambridge for an on-site interview. Met with multiple people some of which were code driven, some were systems design. Overall a pleasant experience. Unfortunately they were very vague to the position they were requiring me for so I was unsure what information they required. Ultimately ended with them telling me they were looking for something else. Interview could have gone differently if they told me that from the start.

**Interview Questions**

* Design a system for capturing click information and redirecting

System Design round. The interviewer started out asking about a few "tell me once upon a time..." questions, then got into the system design. The sys design question was designing an URL shortener, a problem they have been using for years. The interviewer was indifferent and emotionless for the majority of the interview. The interviewer also seems to have a particular checklist in mind on how to go about designing the system; he would stop me mid-sentence to diverge focus onto another component of the design all the time, which significantly limits my system design creativity. Another frustrating thing was that my creative solutions were never deep-dived or valued; the most I got was a nod from him on a few suggestions. For most of the interview, the interviewer was so incredibly emotionless both facially and vocally, I could've interviewed with a robot and the result wouldn't changed. Database Design round. This round was absolutely the most unfair round to applicants. One of the reviews here again indicated the following, "They will tell you knowing syntax is not important but it will be brought up as necessary if they want to ding you based on the implicit bias." This was also very true in my experience. I made it very clear to the interviewer at the beginning that I have no extensive experience with designing SQL tables or writing SQL queries, but all design problems/scenarios given were based on knowing SQL functionality and syntaxes. An applicant can't possibly know the optimal answers to half of the questions without knowing the in-depth SQL topics they were looking for. The guidance given by the interviewer were also very vague; as someone who has limited experience with SQL, there was no way for me to know whether the interviewer was trying to point out a problem in my table design or query design. Coding round. Standard LC midium. Likely the most un-biased round out of all three. I was given the freedom to drive the conversation which allowed to expressed my algorithmic thoughts clearly and spoke out-loud as I coded. Even then, I felt the reason I was given the freedom to drive the conversation was because I gave the interviewer the most optimal solution she had in mind on first try. Overall, the onsite interviews felt like a mess. The interviewers did not feel like they were capable interviewers as they were too young and lacking technical depth; the questions were somehow only solvable by the exact solutions they had in mind. I wish the interviewers would've been more expressive and more open-minded to applicants' suggestions.