

# Atlassian

The interview process at Atlassian is a multi-stage journey that allows candidates to showcase their skills and abilities. It typically consists of several rounds, each designed to assess different aspects of a candidate's qualifications.

Here's a snapshot of the Atlassian interview process for engineering roles:

Stage	Purpose	Duration
Coding Interview 1 & 2	Evaluate candidate's coding skills, code quality, and conceptual thinking	60 minutes each
System Design Interview	Candidates need to demonstrate technical depth, breadth, and proficiency by designing a solution to an established problem	60 minutes
Management Interview	Scenario-based questions to evaluate how candidates approach various aspects of their role in a team environment	60 minutes
Values Interview	Behavioral interview questions to assess if candidates' thought processes and actions align with <a href="#">Atlassian's core values</a>	45 minutes

## Top Skills for Different Roles at Atlassian

### Group Product Manager

Experience and Skills: Atlassian values candidates who have demonstrated their ability to successfully manage products throughout their lifecycle. Desirable skills include:

### Soft Skills:

- Strategizing and market research
- Collaboration and leadership skills

### Technical Skills:

- Expertise in Agile methodologies
- Familiarity with project management tools, especially Atlassian products like Jira or Trello
- Knowledge of software development processes

## **Software Developer**

Experience and Skills: As a software developer, you will encounter various challenges that require innovative solutions. Desirable skills include:

### Soft Skills:

- Problem-solving abilities
- Critical thinking
- Teamwork and communication abilities
- Adaptability (especially to new technologies or changing project requirements)

### Technical Skills:

- Coding Proficiency: Familiarity with programming languages commonly used at Atlassian, such as Java, Python, or JavaScript, is preferred.
- Knowledge of tools and frameworks used by Atlassian, like Jira as well as popular frameworks like React or Angular

[As per Atlassian](#), they are not just looking for technical competencies, but also critical skills and relevant experience.

*“Instead of evaluating candidates based on their proficiency in a specific language, we’re looking for people with broad distributed engineering skills and experience with various languages. And we’ve designed our interview process accordingly: It’s an assessment framework based not on languages, but on qualifications such as problem-solving ability and learning agility. Because of this, we’re able to see a candidate’s depth and breadth. We’d like to see not only how you code, but how you think.”*

## **Atlassian Interview Questions: Technical Knowledge**

The key engineering roles at Atlassian are:

- Frontend
- Backend
- Fullstack
- SRE (Site reliability engineering)

Depending on the role you're applying for, the questions may vary in complexity and specificity. However, here are some common Atlassian technical interview questions that you may when applying for engineering roles:

## **1. Describe the steps you would take to build a medium-sized web application from scratch.**

Building a medium-sized web application from scratch typically involves several key steps:

- Define the project's objectives, features and the targeted audience. Create a detailed project plan. Prepare wireframe mockups and design a UI. Determine the specific technology stack, architecture, and data model.
- Make sure your writing is following best practices. Separate the application into separate modules or components and make it incremental. Design the database infrastructure with an eye on data security and extensibility.
- Conduct integration acceptance testing to confirm that the application functions accordingly. Constantly optimize code and database queries as assets for higher performance.
- Check the application's security by implementing security measures, like authentication, authorization, and data encryption, to ensure the application is safe and user data is protected.
- Install the production version of the application into the servers, as well as build and introduce continuous deployment and continuous integration (CD/CI) pipelines. Ensure that the application works well, has no defects, and is frequently upgraded in line with new needs.

## **2. How would you optimize the performance of an app that has a large tree structure?**

To optimize an application with a large tree structure, you can consider the following techniques:

- Efficient data structures such as balanced trees (AVL or Red-Black trees) are used for storing and traversing the tree structure. Pagination and lazy loading should be applied to get and show tree nodes in batches instead of downloading the whole tree simultaneously.
- Store portions of the tree that are often referenced on a computer to decrease the number of queries to improve throughput. Ensure that the database tables for storing tree nodes are properly indexed to increase the speed of retrievals.
- Create efficient database queries, including methods like recurrent common table expressions for hierarchical information.
- Shift intensive tree-related operations to background jobs or tasks to avoid blocking the principal application line. Perform upfront optimizations such as rendering or virtualization and display only those items that are visible on screen.

### **3. Can you explain how you would design and code a feature that allows users to collaborate on documents online?**

To implement online document collaboration, you can use the following steps:

- Real-Time Collaboration: Users should be able to see changes other people make in real-time through technologies such as WebSockets and server-sent events.
- Conflict Resolution: Put together a conflict resolution method and simultaneously deal with various modifications done by several editors.
- Version Control: There should be a document revision history whereby individuals can trace changes, revert to previous versions, and add comments.

- Access Control: Add access control, such as defining who can view, edit, or comment on this document. Enforce permissions by using authentication and authorization mechanisms.
- Collaborative Editing UI: Design an intuitive and concurrent editing environment featuring cursor position indicators for other users' inputs, highlighting changes, and chatting.
- Persistence: Ensure durability of document data by storing it in a database for consistency.

#### **4. Take any example in ticket closing and explain how will you do hypothesis testing.**

Hypothesis testing involves the following steps:

Formulate Hypotheses:

- Null Hypothesis (H<sub>0</sub>): Average ticket closure time post new processes indicates no significant difference.
- Alternative Hypothesis (H<sub>1</sub>): The mean ticket closure in the pre-process was significantly higher than in the post-process.

Data Collection: Measure ticket closing periods before and after implementing the changes.

Choose Significance Level: Choose an alpha value (e.g.,  $\alpha = 0.05$ ) and state of statistical significance.

Conduct Statistical Test: Compare the mean ticket closing time using a paired-sample or a simple t-test depending on the number of samples and the data distribution.

Compute the Test Statistic: Calculate the test statistic (e.g., t-test) using the population parameter estimates obtained from the sample.

Calculate P-value: The p-value is determined by how likely it is to observe such results if the null hypothesis is true. Such a test statistic is called the likelihood ratio.

Decide: Compare the p-value with the predefined level of significance. Reject the null hypothesis if its p-value is less than the alpha value. Therefore, the null hypothesis should not be discarded when  $p > \alpha$ .

Draw a Conclusion: Ascertain if the time for closing tickets has changed due to the process change using that test's findings.

Report Results: Describe the statistical analysis results and highlight the drawbacks of a process modification to reduce the time for tickets' closure to relevant parties.

## **5. List the difference between Git and Mercurial.**

Here are the key differences between [Git](#) and Mercurial

- Distributed vs. Centralized: Git is a distributed version control system; in this form, each developer has a comprehensive copy with history. On the other hand, Mercurial can be used as a centralized or distributed version control software.
- Performance: People typically think that Git tends to be fast, particularly regarding big repositories.
- Branching Models: Lightweight branches are found in a more flexible branch model of Git than others. Mercurial uses named branches.
- Commands and Terminology: Although they employ identical commands, there are certain vocabularies as well as styles of working that distinguish Git from Mercurial.

- Toolset Integration: While Mercurial offers an integrated toolset, Git relies on other tools such as GitHub and GitLab.

## 6. How do you resolve merge conflicts?

Here's how you can resolve merge conflicts:

- Manual Resolution: Review conflicted files in your code editor, locate conflicted sections that are highlighted by git/mercurial, and make manual editing on the conflict to solve it.
- Use of Mergetools: Set up a Mergetool that will provide you with a visual method of resolving your conflicts. Some tools, such as KDiff3 or P4Merge, can help visualize and resolve any conflicts.
- Accept Changes: Merging can be canceled with `git merge --abort` or `hg merge --abort` for a choice of taking changes from another branch than a given branch.
- Committing Changes: Then resolve the conflict and commit with `'git commit'` or `'hg commit.'`

## 7. How do you start services in Linux and Windows?

Linux

- You can use the command “systemctl” to start a service. For example: ``sudo systemctl start serviceName.``
- You may opt for service-specific commands, including `service serviceName start`.

Windows



- Here, use services.msc GUI for initiation of a service. Click on the start to search for the service required.
- Alternatively, type net start serviceName on your command prompt or PowerShell.

## 8. What are the differences between Process and Thread?

Definition: Thread is the smallest execution element inside the process, which runs as a standalone program executing by itself in separate memory space.

Criteria	Process	Thread
Resource Allocation	Memory is distinct for processes	Memory is shared
Communication	IPC is necessary for communication between processes	Threads share variables and thus do not require additional communication channels.
Overheads	It is more expensive to make a new process than a new thread.	Less expensive than creating a new process.
Parallelism	Multi-core systems offer parallelism, which allows processes to run in parallel and hence better parallelism.	A single core can be only accessed by one thread at once.

## **9. What do you mean by virtual memory? How does the virtual memory map to physical memory?**

An operating system feature called virtual memory involves using software and hardware to transfer data between hard disk storage and RAM to enable a computer with insufficient amounts of RAM to function adequately with some memory issues.

### Mapping to Physical Memory

Virtual memory comprises several pages mapped into real memory locations. The operating system of the memory management unit does it. For example, when there is a demand for a virtual address from a particular program, MMU translates the request into a corresponding physical address of real memory so that the program can work with that actual RAM.

## **10. How do processes communicate?**

Processes primarily communicate with each other through shared memory and message-passing. Here are some other ways

- Pipes: Channel of one-way communication between interrelated processes.
- Message Queues: A message can be sent through a message queue among processes. It's usually used for processes that are not directly related to each other.
- Sockets: Sockets enable communication between processes running on different machines via networks.
- Files: Shared files allow processes to "talk" with each other, known as read/write communication between processes.

- Signals: Signals can be sent by processes to inform another process about an event or to perform a certain action.

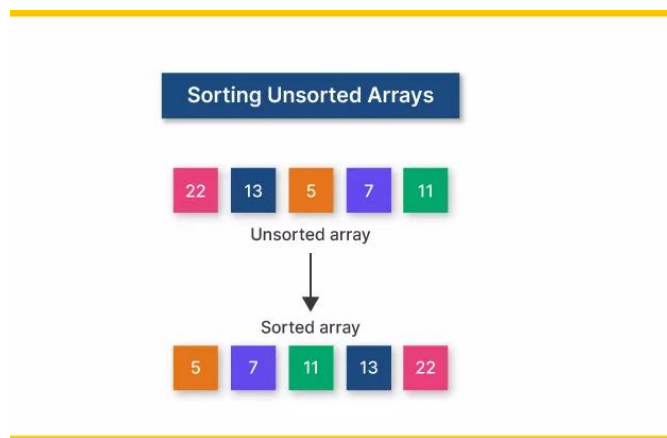
The communication mechanism to be used depends on the specific needs of the application.

## 11. What is a rate limiter?

A rate limiter is a way to limit operations carried out on a system for a short time when too much demand over the operations cannot be taken up for the system. For example, abuse mitigation via web API rate limiting i.e. thwarting brute force login attempts by allowing a limited number of tries per hour and rate control on outbound network traffic.

Rate limiters are used to prevent any form of mistreatment or bad acts. To ensure fair usage and avoid a situation whereby one user or system can dominate resources. It also helps guard against server overloads and ensures system stability.

## 12. Explain various techniques to sort an unsorted array efficiently.



Some of the techniques that can be used to sort an unsorted array are:

- Quick Sort: Divide and conquer approach, separating the array into smaller subdivisions.

- Merge Sort: Sort the array in half recursively and merge.
- Heap Sort: Build a heap of maximum and minimum and continually extract the root element.
- Bubble Sort: Reversely interchanges consecutive elements whenever they are in the reverse order.
- Insertion Sort: Put the sorted array by insertion by picking off each element one after another from the unsorted section.

**13. Given an array representing the heights of towers, write a function to find the index of the tower shorter than the current tower.**

Algorithm

1. Initialize a stack and an array to hold the result indices.
2. Iterate through the tower heights.
3. Push the current index into the result array while the stack is not empty. The towers are shorter than the one indexed on the top of the stack.
4. Load the address of the first record currently stored for reference on top of the stack.
5. End returns the result array.

**14. Discuss challenges and strategies when working with circular arrays in data structures.**

Challenges

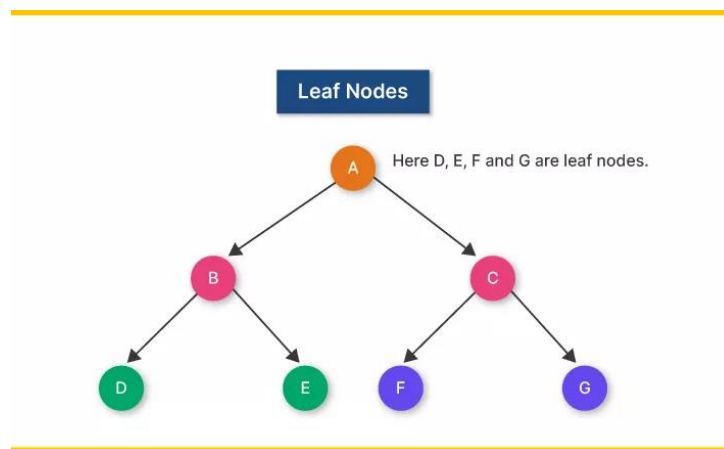
- Identifying the beginning and ending positions within a circular array.
- Efficiently iterating over elements.

- Handling circular array rotations.

### Strategies

- Deal with index wrapping through modular arithmetic.
- Keep track of the base position of the circular array.
- For this purpose, consider using a circular queue or deque.

## 15. Explain how to identify and manipulate leaf nodes efficiently.



### Identification of Leaf Nodes

- The last nodes, or with no “child” nodes, are leaf nodes.
- Cross the tree and look for a node with neither the left nor the right child.

### Manipulation of Leaf Nodes

- The parents need to be updated when we are inserting or deleting.
- In addition, while removing a leaf node, update its parent’s pointer as null during deletion.

## **17. Write an algorithm to find the nearest tower for each tower in a given array of tower heights.**

Algorithm

1. Begin with an empty stack and the answer indices array.
2. Iterate through the tower heights.
3. Push the current index in the result array while the stack is non-empty and the current tower is less than the tower at the index on top of the stack.
4. Place the present index on the stack.
5. Return the result at the end.

## **18. Given a collection of elements, devise an algorithm to process them efficiently.**

Algorithm:

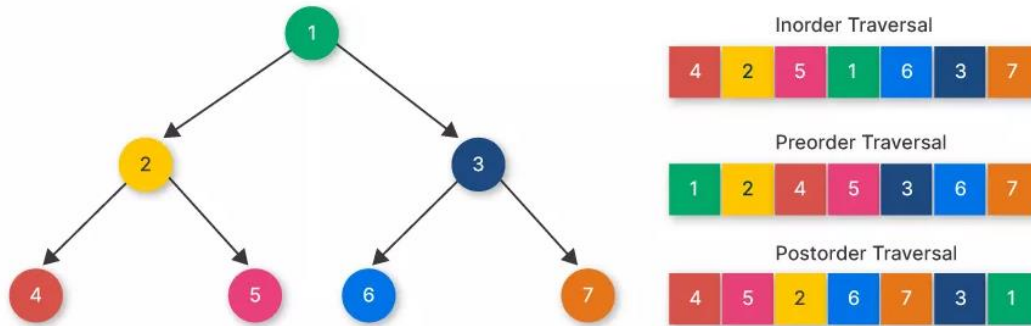
1. Determine Objective: Ensure you clearly define the purpose of processing.
2. Choose Data Structures: Choose appropriate [data structures](#) depending on the kinds of elements and what you would like to perform.
3. Optimal Algorithms: Efficiently use algorithms that execute necessary functions.
4. Parallel Processing: If possible, study some parallel processing techniques that would help increase the algorithm's speed.
5. Optimize for Special Cases: Categorizing and optimizing typical and exceptional cases.
6. Complexity Analysis: Efficiency is ensured by analyzing the time and space complexities.

## **19. Explain the process of traversing a binary tree and the different types of tree traversal algorithms.**

Binary Tree Traversal Process:

1. Start at the root.
2. Carry out the specified operation for every node.
3. Apply this process recursively onto the left and right subtrees.

## Tree Traversal Techniques



There are three types of tree traversal algorithms:

**In-Order Traversal (Left-Root-Right):** Start at traverse left subtree, then visit a root to have traversing continued in the right subtree.

**Pre-Order Traversal (Root-Left-Right):** Go down, turn left, descend, move right.

**Post-Order Traversal (Left-Right-Root):** Visit the root, traverse the left subtree, and traverse the right subtree.

## 20. Describe operations to increment specific elements and their impact on the list.

To increment specific elements, follow these steps:

1. Loop across the list of integers.
2. Increasing particular aspects upon specified conditions or criteria should be identified
3. Increase the selected elements by a certain amount of number.
4. Make a copy of that and run through the list.

Impact on the List

- Only those elements that satisfy certain conditions are altered.
- The condition check and selection determine the overall time complexity to increase the relevant array elements.

## **21. Discuss the importance of coding prompts in software development and their role in enhancing problem-solving skills.**

Coding prompts give an insight into how users should address real-world problem-solving situations. They mimic problems that arise in practical software engineering. Prompts also enable evaluators to establish if a candidate can practically use theories in problem-solving situations.

Coding prompts help enhance problem-solving skills by:

- Encouraging systematic problem analysis.
- Promoting efficient algorithm design.
- Enhancing coding proficiency and style.
- Stimulating creativity towards identifying optimum solutions.
- Improving debugging and optimization skills.

## **22. Given a problem statement, explain how you would determine the appropriate size for an array in a software development project.**

The steps to determine the appropriate size for an array in a software development project are:



1. Understand Requirements: Make sure you know what the particular software does, and why it does it.
2. Data Analysis: Determine the data type the array must handle regarding volume and type.
3. Consider Future Growth: Plan for a possible increase in data size to achieve scalability.
4. Memory Constraints: Ensure that you consider the limitation of the memory of the targeted platform.
5. Optimization vs. Space Trade-off: Strike a balance between space efficiency and the demands of the project.
6. Testing and Validation: Use sample data to test the array size, then verify performance.

## **23. Outline the key considerations in designing efficient issue-tracking software, emphasizing data structures and algorithms.**

When designing efficient issue-tracking software, the key considerations for data structures are:

- Store and retrieve issues using appropriate data structures like hash tables and balanced search trees.
- Use priority queues in controlling issue priorities.
- Use graphs to keep tabs on the interdependence of problems.

On the other hand, the key considerations for algorithms are:

- Develop effective searching algorithms to retrieve targeted questions rapidly.
- Sorting issues according to priority, create\_date, etc.
- Utilize graph algorithms for identifying interdependencies and resolving problems between issues.
- Algorithms that reduce issue resolutions in optimizing workflows.

## **24. Describe an algorithm to transform a list of integers based on specified rules and constraints.**

### Algorithm

1. Go through that integer's list again.
2. Transform each integer following the particular rules and restrictions.
3. Add transformed values to a list.

### Example Rules

- Double it if it's an even number.
- Take a square of the value if it is prime.
- Add a constant value to each element.

### Impact

- The algorithm changes the original list through predefined rules and limitations.

## **25. Given a large input array, discuss techniques to process it efficiently, considering both time and space complexity.**

To efficiently process a large input array, we can consider the following time and space complexity techniques:

### Time Complexity

1. *Parallel Processing*: Split the array into chunks, then process them in parallel.
2. *Streaming Algorithms*: Process sequentially the elements arrived at individually, thus avoiding having all the array items kept in the memory.
3. *Batch Processing*: Split your array for gradual processing of increments.

Space Complexity

1. *In-Place Algorithms*: Change the array elements without employing extra memory.
2. *Sampling*: Reduce memory usage by processing a subset of an array.
3. *Streaming Processing*: Store array elements one-by-one, minimizing the need for maintaining an array in memory.

## **Atlassian Interview Questions: Behavioral Skills**

Here are some questions that you may encounter during the HR round i.e. the values interview:

### **1. How would you summarize your engineering journey?**

My engineering journey has shaped me into a resilient and adaptable professional who is always eager to take on new challenges and contribute to meaningful projects. From my early days of studying engineering in college to my current role, I have always been passionate about solving complex problems and finding innovative solutions. Throughout my journey, I have worked on various projects that have allowed me to develop my technical skills and gain practical experience. One such project was designing and implementing a solar power system for a remote community. I worked with limited resources and logistical constraints, but it enhanced my technical skills and ability to collaborate effectively.

### **2. What is the biggest failure in your life? How did you overcome that?**

One of the biggest failures in my life was when I failed to secure a job after graduating from college. It was a difficult time for me as I had put a lot of effort and hard work into my studies, and I had high expectations of finding a job right away. To overcome this, I broadened my job search, networked, and attended job fairs. I also took online courses and workshops to enhance my skills. Seeking guidance from career counselors and mentors, I improved my resume and interview skills. Eventually, my perseverance paid off, and I secured a job aligned with my interests and qualifications. This failure taught me the importance of adaptability and growth.

### **3. Do you think your values align with the Atlassian culture? How?**

Yes, I believe my values align with the Atlassian culture. Atlassian is known for its emphasis on collaboration, transparency, and continuous learning, which are values that I deeply resonate with. Throughout my career, I have always prioritized open and honest communication, both within my teams and with stakeholders. I believe in fostering a culture of trust and collaboration where everyone's ideas are valued and respected. I also have a strong passion for continuous learning and personal growth, which aligns with Atlassian's focus on innovation and improvement.

### **4. Give an example of a time you dealt with conflict at work.**

There was a time when there was a disagreement between two team members regarding the approach to a project - one preferred a traditional, structured approach while the other favored an agile, adaptable approach. I facilitated a meeting where both parties were able to express their concerns and perspectives. Through open communication and active listening, we were able to find a compromise and move forward with a solution that satisfied everyone involved.

### **5. Have you ever faced a situation where you thought you would miss a deadline? How did you deal with it?**

Yes, I have faced a situation where I thought I would miss a deadline. It was during a particularly busy period when I had multiple projects to handle simultaneously. To manage the situation, I prioritized my tasks and created a detailed schedule to ensure I could allocate sufficient time to each project. I also communicated with my team and stakeholders to manage expectations and negotiate deadlines if necessary. By staying organized and focused, I was able to meet all the deadlines without compromising the quality of my work.

## **Atlassian Interview Questions: Tips for Effective Preparation**

Here are some tips that can help you prepare for an interview at Atlassian:

- Familiarize yourself with Atlassian products such as Jira, Confluence, Bitbucket, or Trello.
- Practice answering behavioral-based interview questions that assess your problem-solving abilities.
- Research the company culture at Atlassian to align your responses accordingly.
- Familiarize yourself with the Agile methodology, as Atlassian heavily emphasizes this approach in their development process.
- Prepare specific examples from your previous work experience that demonstrate your ability to work collaboratively in a team setting.

*Array is a favourite topic of Atlassian interviewers. Find out the advantages and disadvantages of array in programming in*

## **What do past candidates have to say?**

Many candidates were impressed by the professionalism and expertise of the interviewers at Atlassian. They found that the hiring committee was genuinely interested in their experience, skills, and projects. The interviewers asked thought-provoking questions that allowed candidates to showcase their abilities.

However, some candidates also found certain aspects of the interview challenging. For example, they mentioned that technical questions required in-depth knowledge and problem-solving skills.

Engineers who interviewed at Atlassian also advised candidates to actively engage in discussions during coding sessions or design exercises. Sharing thoughts openly and seeking feedback throughout the session can demonstrate collaboration skills and a growth mindset.

If you are preparing for an interview with Atlassian, it is crucial to familiarize yourself with their interview process and understand the skills they value in each role. Remember to prepare detailed responses that highlight your experience, technical knowledge, problem-solving skills, and ability to work collaboratively.