Week- 5

Backend Engineering Launchpad

Pawan Panjwani

Airtribe

Advanced Node Js Streams and web sockets

Airtribe

Agenda

- Clustering in Node JS
- Understanding streams
- Types of streams - Readable, writable, Duplex, Transform
- How streams work, advantages of using streams in node JS
- Websockets and how do they work
- Handling websocket events and scaling web sockets
- Live coding
- Question and answers

Streams- Node JS

Airtribe

What are streams ?

1) Streams are objects in Node.js that enable efficient processing of large amounts of data
2) Streams provide a mechanism for reading and writing data incrementally, rather than all at once
3) Streams can be thought of as a sequence of data elements made available over time`

Types of streams ?

1) Readable streams: enable the reading of data from a source
2) Writable streams: enable the writing of data to a destination
3) Duplex streams: enable both reading and writing of data
4) Transform streams: enable the transformation of data as it is being read or written

Stream Methods

1) pipe(): connects a readable stream to a writable stream, automatically handling data flow
2) unpipe(): removes the connection between a readable and writable stream
3) pause(): temporarily stops data flow in a readable stream
4) resume(): resumes data flow in a readable stream after it has been paused

Streaming Examples

1) Example 1: Processing a large CSV file with a readable stream
2) Example 2: Compressing a file with a transform stream and writing it to a destination with a writable stream
3) Example 3: Creating a real-time chat application with duplex streams`

Error Handling and Practices

1) Handling errors and error events: always handle errors in streams using error events and try/catch blocks
2) Proper use of backpressure: backpressure is used to manage the flow of data between streams to avoid overloading resources
3) Managing stream resources: it's important to properly close streams and release resources when they are no longer needed

Streaming Use Cases (1)

1) File I/O operations: Streams are often used to read or write large files in Node.js, allowing for efficient and scalable file I/O operations.
2) Network communication: Streams are also used for network communication, such as reading and writing data from HTTP or TCP sockets.
3) Database operations: Streams can be used to read or write data from databases, such as MongoDB or MySQL.

Streaming Use Cases (2)

1) Data processing and transformation: Streams are useful for processing and transforming data in real-time, such as parsing large JSON or CSV files.
2) Compression and encryption: Streams can be used for compression and encryption operations, such as gzip compression or encryption with AES.
3) Real-time data streaming: Streams can be used for real-time data streaming applications, such as live video or audio streaming.

Websockets – Node JS

Airtribe

What are websockets?

1) Websockets are communication protocol that provides full-duplex, bi-directional communication over a single TCP connection
2) WebSockets provide real-time communication without the overhead of HTTP request-response cycle
3) Faster and more efficient communication, reduced latency, lower server load, support for real-time applications

Airtribe

How do websockets work ?

1) Overview of the WebSocket protocol: handshake, data framing, message format, and closing the connection
2) Handshaking process: client sends a HTTP request to upgrade to WebSocket protocol, server responds with a 101 status code to confirm upgrade
3) Sending and receiving messages: messages are sent in a binary or text format, and can be sent in both directions simultaneously
4) Closing a WebSocket connection: either side can initiate closing the connection, and there are several ways to handle it

Building real time application using streams, web sockets, express Node Js

Airtribe

Thank you

Airtribe