Week- 3

Backend Engineering Launchpad

Pawan Panjwani

Airtribe

AUTHENTICATION AND AUTHORIZATION

Node JS

Airtribe

Agenda

- What is authentication and authorization ?
- Understanding JWT (JSON web tokens)
- Understanding login and signup - Token based authentication using mongodb.
- Node packages - bcrypt, jwt, mongoose
- Role based authentication and authorization for the course rating app.
- Question and answers

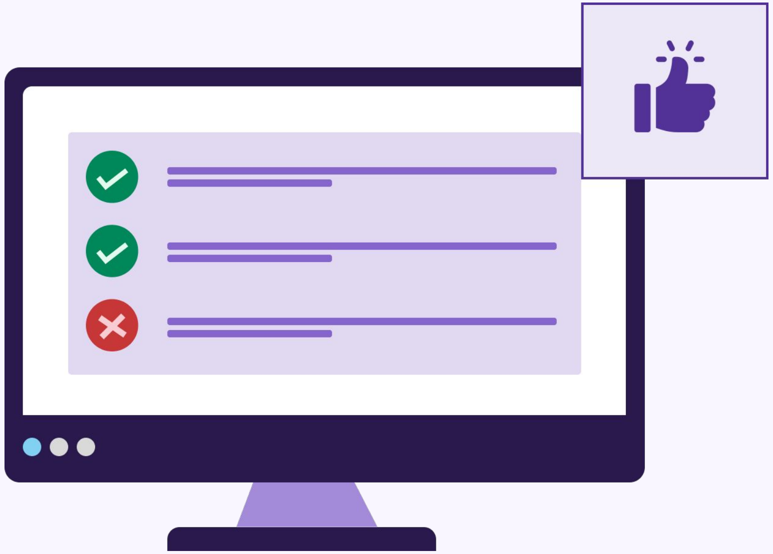
What is authentication

Authentication:

Confirms Users are who they say they are

Airtribe

What is authorization



Authorization:

Give users permission to access a resource.

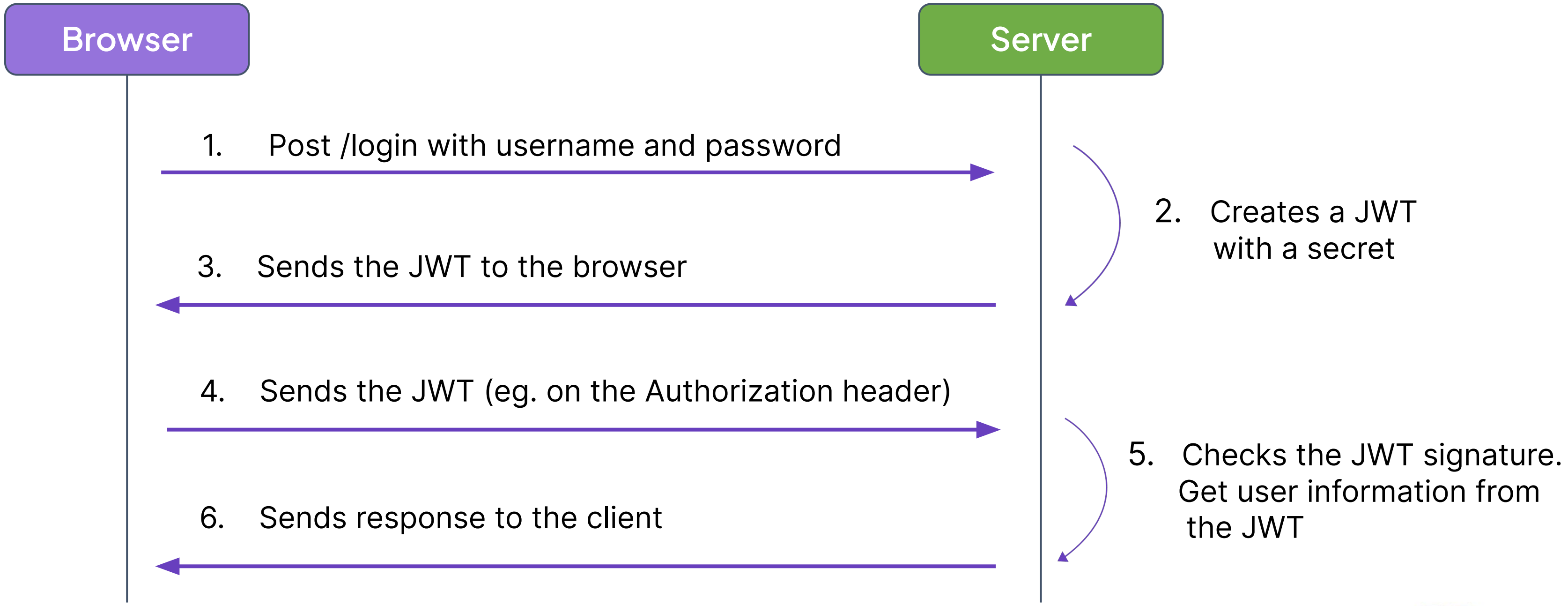
Airtribe

What is Authentication & Authorization

	Authentication	Authorization
PURPOSE	• Verifies user identity	• Permits access to resources
REQUIREMENTS	• Identity credentials based on knowledge, possession, and/or inherence • Authentication solution	• Authenticated identity and access control policies • Authorization solution
RESPONSIBILITIES	• Network security staff determine which factors to adopt. • Users provide authentication factors when requesting access	• Leadership sets security strategies • Departments and workgroups define access criteria • Network security staff implement and maintain access control system

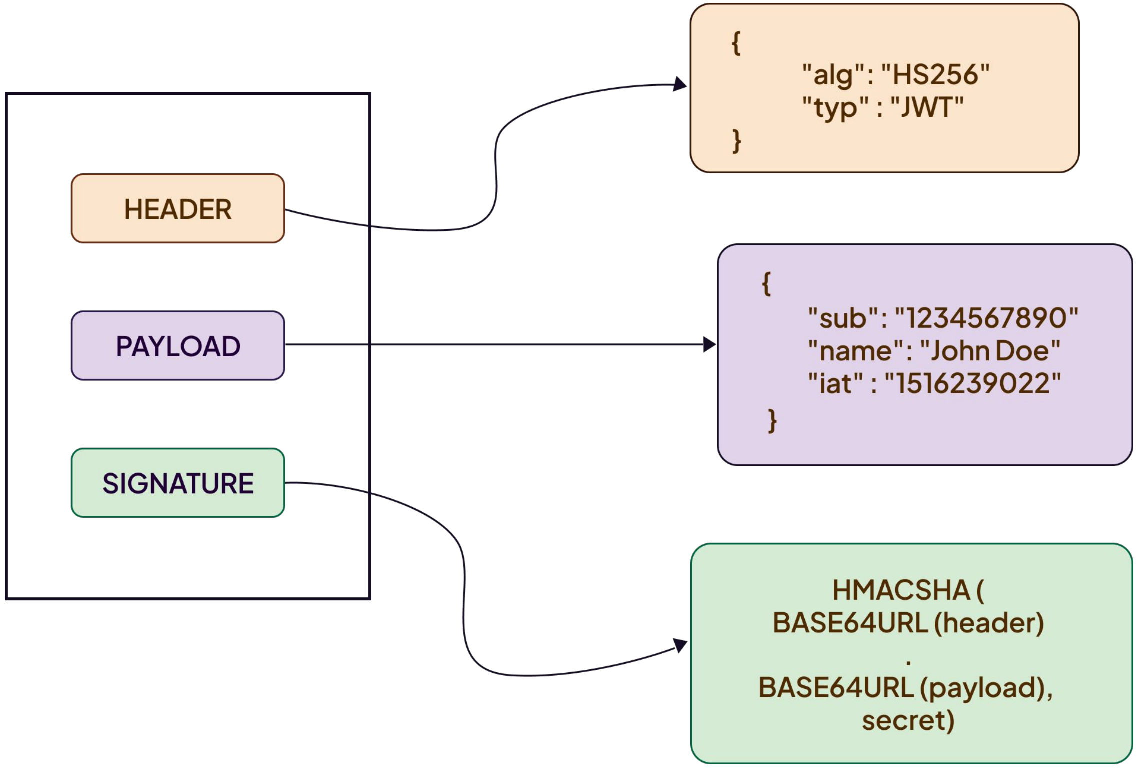
Airtribe

What is JWT based Authentication & Authorization


Browser
Server
1. Post /login with username and password
2. Creates a JWT with a secret
3. Sends the JWT to the browser
4. Sends the JWT (eg. on the Authorization header)
5. Checks the JWT signature. Get user information from the JWT
6. Sends response to the client

Airtribe

Structure of JWT



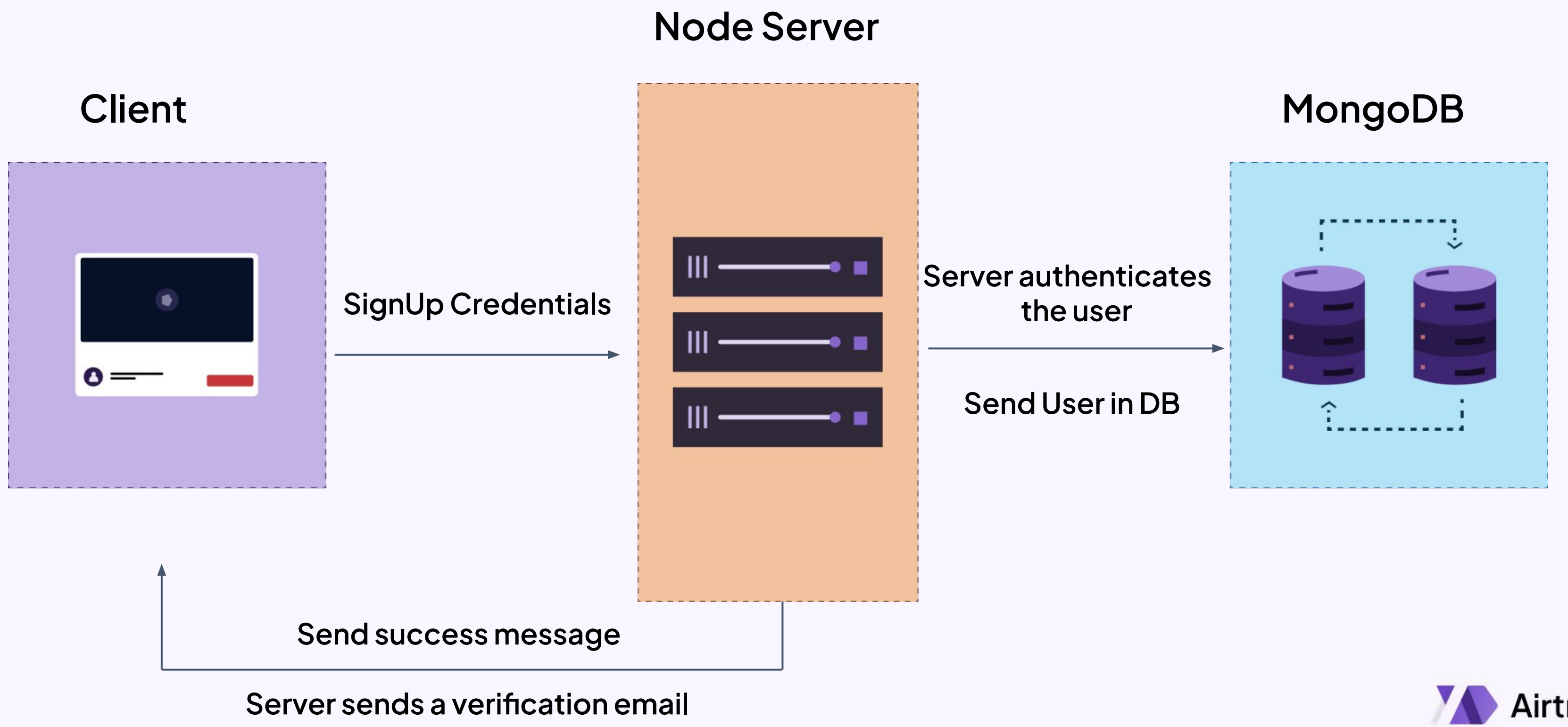
```
{
    "alg": "HS256"
    "typ" : "JWT"
}
```

```
{
    "sub": "1234567890"
    "name": "John Doe"
    "iat" : "1516239022"
}
```

```
HMACSHA (
BASE64URL (header)
.
BASE64URL (payload),
secret)
```

Airtribe

Auth using MongoDB and & Express



Airtribe

Strong password?

$$\begin{tikzcd}
\text{password} \arrow[d] \\
\boxed{\text{bycrypt}} \arrow[d] \\
\boxed{\text{\$2b\$10\$gOancHkngDJvrXkW809800}}
\end{tikzcd}$$

Authentication and authorization

1. Instructions to follow during the activity

30-Min Live Exercise

Airtribe

Course Rating Application with authentication and authorization

Authentication, authorization and security

Build a REST API express web server named Airtribe course rating app, which would be responsible for taking in a rating from a student for a given course and when asked about the rating for a given course, aggregates the ratings given by the previous students and shows it to the Airtribe student. This application would help students, how the other students felt about the course they are taking.

Course Rating Application with authentication and authorization

Authentication, Authorization and Security

For the course rating application you also need to help Airtribe add the authentication and authorization to the API endpoints built for the course rating application using the Node.js middleware.

In order to do that you need to first add authentication to the application. In order to do that create two endpoints named login and register

Airtribe

Course Rating Application with authentication and authorization

Authentication, Authorization and Security

User data needs to be stored in Mongodb in the form of documents

1) /login → Logs the user in and returns the JWT token
2) /register → Creates a new user and stores the encrypted password

Once authenticated, You need to add authorization for the following endpoints.

There would be two roles - **admin role and normal role**

Course Rating Application with authentication and authorization

Authentication, Authorization and Security

The capabilities possessed by the roles are as follows

1) Admin → Can create and modify the course
2) Normal → Can fetch the courses, course Info, give rating, and fetch the rating of the course.

Course Rating Application with authentication and authorization

Authentication, Authorization and Security

Add the following capabilities to the below specified endpoints

1. GET - /courses → gets the list of the courses and their details
2. GET - /courses/1234 → gets the details of the course named 1234
3. GET - /courses/1234/rating → gets the average rating of all the students for the course 1234
4. POST - /courses → Creates the course with the provided details
5. POST - /courses/1234/rating → Adds the rating to the provided course 1234
6. PUT - /courses/1234 → Modifies the information of the course with the provided details
7. PUT - /courses/1234/rating → Modifies the rating of the provided course 1234

Problem Discussion

In a microservices architecture, there is a scenario where an Image Service 1 needs to communicate with Service 2. However, Service 2 should only allow Image Service 1 to access its resources if Image Service 1 can provide proper authentication. The problem is to design and implement an authentication and authorization mechanism that enables secure communication between Image Service 1 and Service 2, ensuring that only authenticated and authorized requests from Image Service 1 are accepted.

Problem Discussion

Key considerations:

1) Designing a robust authentication mechanism: How can Image Service 1 prove its identity to Service 2 in a secure and reliable manner?

2) Implementing access control and authorization: How can Service 2 authorize Image Service 1's requests based on predefined access control policies?

3) Managing authentication and authorization across services: How can the authentication and authorization mechanisms be seamlessly integrated into both Image Service 1 and Service 2?

Assignment

Project Title: News Aggregator API

Objective: Build a RESTful API that allows users to fetch news articles from multiple sources based on their preferences.

Project Description: In this project, we will create a RESTful API using Node.js, Express.js, and NPM packages. The API will allow users to register, log in, and set their news preferences (e.g., categories, sources). The API will then fetch news articles from multiple sources using external news APIs (e.g., NewsAPI). The fetched articles should be processed and filtered asynchronously based on user preferences.

Assignment

Medium Difficulty Requirements:

1. Set up a basic Node.js project with Express.js and other necessary NPM packages.
2. Implement user registration and login using bcrypt and JWT for password hashing and token-based authentication.
3. Create an in-memory data store (e.g., an array) to store user information and their news preferences.
4. Implement a RESTful API with the following endpoints:
   - POST /register: Register a new user.
   - POST /login: Log in a user.
   - GET /preferences: Retrieve the news preferences for the logged-in user.
   - PUT /preferences: Update the news preferences for the logged-in user.
   - GET /news: Fetch news articles based on the logged-in user's preferences.

Airtribe

Assignment

Medium Difficulty Requirements:

5. Use external news APIs to fetch news articles from multiple sources. Incorporate async/await and Promises in the process of fetching news data and filtering it based on user preferences.
6. Implement proper error handling for invalid requests, authentication errors, and authorization errors.
7. Add input validation for user registration and news preference updates.
8. Test the API using Postman or Curl to ensure it works as expected.

Airtribe

Assignment

Optional extension:

1. Implement a caching mechanism to store news articles and reduce the number of calls to external news APIs. Use async/await and Promises to handle cache updates and retrievals.
2. Allow users to mark articles as "read" or "favorite". Implement endpoints to:
   - POST /news/:id/read: Mark a news article as read.
   - POST /news/:id/favorite: Mark a news article as a favorite.
   - GET /news/read: Retrieve all read news articles.
   - GET /news/favorites: Retrieve all favorite news articles.
3. Implement an endpoint to search for news articles based on keywords: GET /news/search/:keyword.
4. Implement a mechanism to periodically update the cached news articles in the background, simulating a real-time news aggregator.

Thank you

Airtribe