JWT (JSON Web Token) – Conceptual / OLD NOTES

> **Purpose of these notes:** - JWT ko zero se samajhna - Fast revision ke liye - Interview-oriented theory notes - Code-independent (pure concepts)

---

1. What is JWT?

JWT (JSON Web Token) ek **token-based authentication mechanism** hai jisme server client ko ek token deta hai aur client us token ko har request ke sath bhejta hai.

- JWT mainly **authentication** aur **authorization** ke liye use hota hai
- Token JSON format me hota hai
- Server token verify karta hai, session store nahi karta

---

2. JWT Full Form

JSON Web Token

---

3. Why JWT? (Need of JWT)

Session-based authentication ki problems:

- Server ko har user ka session store karna padta hai
- Zyada users hone par memory issue
- Scalability problem
- Microservices ke liye suitable nahi

JWT ke advantages:

- Stateless authentication
- Server-side session nahi hota
- Scalable approach
- REST APIs ke liye best
- Microservices friendly

---

4. JWT Structure

JWT 3 parts se milkar banta hai, jo dot (`.`) se separate hote hain:

```
HEADER.PAYLOAD.SIGNATURE
```

4.1 Header

Header me token ka metadata hota hai: - Algorithm - Token type

Example:

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

4.2 Payload

Payload me **claims** hote hain (user data):

Example:

```
{
  "sub": "username",
  "iat": 1700000000,
  "exp": 1700003600
}
```

Types of Claims:

- Registered claims (sub, iat, exp)
- Public claims
- Private claims

4.3 Signature

- Header + Payload + Secret key se generate hoti hai
- Token ke data ko tamper hone se bachati hai
- Payload change hua to signature invalid ho jata hai

👉**Signature token ki integrity ensure karta hai**

---

5. JWT Authentication Flow

1. User username & password bhejta hai
2. Server credentials verify karta hai
3. Server JWT generate karta hai
4. JWT client ko bhej diya jata hai
5. Client JWT store karta hai
6. Client har request me JWT bhejta hai
7. Server JWT validate karta hai
8. Access allow ya deny hota hai

---

6. JWT kahan bheja jata hai?

JWT har request ke sath **HTTP Header** me bheja jata hai:

```
Authorization: Bearer <JWT_TOKEN>
```

👉Header me bhejna recommended hai

---

7. Stateless Authentication (Most Important)

Stateless ka matlab:

> **Server user ka session yaad nahi rakhta**

- Server koi session store nahi karta
- Har request apne sath complete information laati hai
- Token hi user ki identity prove karta hai

👉**JWT = Stateless Authentication**

---

8. JWT vs Session Authentication

Feature	Session	JWT
User data	Server me	Client (Token) me
Session memory	Required	Not required
Nature	Stateful	Stateless
Scalability	Low	High

Feature	Session	JWT
Microservices	Poor	Excellent

---

9. Token Expiration

- JWT me expiry time hota hai
- Expired token automatically invalid ho jata hai
- User ko dobara login karna padta hai

---

10. JWT Security Points

- JWT digitally signed hota hai
- Payload readable hota hai but editable nahi
- Signature verify hone par hi token valid hota hai
- HTTPS ka use mandatory hai

---

11. Common JWT Terms (Interview)

- **Token** → Authentication proof
- **Bearer** → Token holder
- **Claims** → Token ke andar data
- **Stateless** → No server session

---

12. JWT Interview One-Liners ⭐

- JWT is used for stateless authentication
- JWT has Header, Payload and Signature
- JWT removes server-side session
- JWT is commonly used in REST APIs
- Signature ensures data integrity

---

13. Quick Revision Summary

- JWT = JSON Web Token
- Token-based authentication
- No server session
- Token sent in header
- Stateless authentication

• Secure using signature
• Has expiry time
• Scalable approach

👈 **These are PURE OLD / CONCEPTUAL JWT NOTES** No code, no classes, no latest implementation details