

COMPLETE NOTES ON TOKENS & JWT (JSON WEB TOKENS)

1. What is a Token?

A token is a randomly generated string that represents the identity of a user. It acts like a digital ID card.

- Created after verifying username and password.
- Contains encoded user information.
- Used for authorization to access protected APIs.

2. Why Tokens?

Without a token, every request requires username & password. With a token, user logs in once and uses token for all next requests.

3. What Information Does a Token Contain?

Example:

```
{  
  "username": "harish",  
  "email": "harish@gmail.com"  
}
```

This gets stored as an encoded string.

4. Encoding & Decoding

Encoding:

`{'user': 'harish', 'location': 'hyd'}` → encoded string

Decoding:

encoded_string → `{'user': 'harish', 'location': 'hyd'}`

5. What Is JWT?

JWT = JSON Web Token. A secure, stateless token format used for authentication.

JWT Structure:

header.payload.signature

6. Components of JWT

Header:

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

Payload:

```
{  
  "id": 12,
```

```
"username": "harish",  
"role": "admin",  
"exp": 1700000000  
}
```

Signature:

Ensures tokens are not tampered with.

7. Why JWT Is Popular?

- Stateless
- Fast
- Secure
- Cross-platform
- Works in mobile + web + microservices

8. Django – JWT Token Generation

Install:

```
pip install PyJWT
```

Generate:

```
jwt.encode(payload, secret, algorithm="HS256")
```

Decode:

```
jwt.decode(token, secret, algorithms=["HS256"])
```

9. Where Tokens Are Stored?

- localStorage
- sessionStorage
- Cookies
- Mobile secure storage

10. Using Tokens in API Requests

Authorization: Bearer <token>

11. Real-Time Use Cases

- User authentication
- Role-based authorization
- Mobile app authentication
- Password reset links
- Single Sign-On
- Microservices

12. Advantages of JWT

- Stateless
- Fast
- URL safe
- Scalable

13. Disadvantages of JWT

- Token size is bigger
- Cannot be forcefully invalidated
- If stolen, attacker can use until expiry

14. Interview Points

- JWT = JSON Web Token
- Used for authentication & authorization
- Contains header, payload, signature
- Supports expiration time
- Works without storing sessions in server

15. Token Flow

1. User logs in
2. Server verifies
3. JWT generated
4. Client stores token
5. Sends token for every request
6. Server validates and responds