JSON Web Token

1. What is JSON Web Token (JWT)

JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. It is widely used for authentication and secure information exchange in web applications.

2. Structure of a JWT

A JWT consists of three parts, separated by dots ('.'):

2.1. **Header**: Contains metadata about the token, such as type (JWT) and signing algorithm (e.g., RS256).
2.2. **Payload**: Contains claims (statements about an entity, e.g., user data) and additional metadata.
2.3. Signature: Ensures the token's integrity by signing the header and payload with a secret key.

Example Header

```
{
     "alg":  "RS256",
     "typ": "JWT"
}
```

Example Payload

```
{
   "sub": "user123",
   "name": "Hussam",
   "role": "admin"
}
```

3. How JWT Works

3.1. **User Login**: The user logs in with their credentials.
3.2. **Token Generation**: The server validates the credentials and generates a JWT.
3.3. **Token Usage**: The client includes the JWT in subsequent requests (usually in the Authorization header as Bearer <token>).
3.4. **Verification**: The server verifies the token's signature to ensure its validity.

4. Use Cases

4.1. **Authentication:** To verify the user's identity.
4.2. Information Exchange: To securely transmit data between parties.
4.3. **Session Management:** To maintain a stateless session.

5. Benefits of JWT

5.1. Compact and portable.
5.2. Self-contained; no need for server-side session storage.

5.3. Secure when implemented correctly.

6. Key Changes:

6.1. Signing Algorithm: Changed from HS256 to RS256 (more secure for production environments).

6.2. Payload Data: Modified `sub`, `name`, and `admin` to `sub`, `name`, and `role` for better clarity.

6.3. Example Data: Updated with more generic and less personal values for enhanced security.