

# Angular Cheat Sheet

## 1 JSON Web Token (JWT)

- A standard for secure data exchange, commonly used for authentication in Angular applications.
- Represents a compact, self-contained token for securely transmitting information between parties.

## 2 @Injectable Decorator

- `@Injectable({ providedIn: 'root' })` makes a service available app-wide without needing to manually add it to the providers array in a module.
- Ensures the service is a singleton, accessible throughout the Angular application.

## 3 HttpClient

- `this.http` is an instance of Angular's built-in `HttpClient` service.
- The `.post()` method is used to make HTTP POST requests to a server.
- Requires `HttpClientModule` to be imported in the application module (e.g., `app.module.ts`).

## 4 localStorage vs sessionStorage

### 4.1 What is localStorage?

- A browser API that stores data persistently, even after the browser is closed.
- Data remains until explicitly removed (not cleared automatically).
- Stored as key-value pairs, always as strings.

### 4.2 Use Cases

- Storing login tokens (e.g., JWT).

- Saving user settings or preferences.
- Easy to use and widely supported across browsers.

### 4.3 Scenario: JWT Authentication in Angular

- **Context:** Using JWT-based login in a standalone Angular project.
- **Goal:** Store the JWT token after login, send it in API requests for authorization.
- **Recommendation:** Use `localStorage` for persistent login across page refreshes and browser sessions.

### 4.4 Comparison: `localStorage` vs `sessionStorage`

Feature	<code>localStorage</code>	<code>sessionStorage</code>
Persists after tab/browser is closed?	51 Yes	55 No
Lifetime	Until manually cleared	Current tab session
Shared across tabs/windows	51 Yes	55 No
Access from JavaScript	51 Yes	51 Yes
Best for	Staying logged in	Temporary sessions

Table 1: `localStorage` vs `sessionStorage`

### 4.5 Why Choose `localStorage`?

- Keeps users logged in after page refreshes or browser restarts.
- Simplifies JWT implementation in Angular, especially for learning projects.
- Easier debugging due to consistent behavior.

### 4.6 When to Use `sessionStorage`?

- For automatic logout when a tab is closed (e.g., banking apps).
- For highly sensitive applications where login should not persist across tabs or sessions.

### 4.7 Security Tips for `localStorage`

- Protect against XSS (Cross-Site Scripting) attacks.
- Avoid storing sensitive data like passwords in `localStorage`.

## 5 Boolean Conversion with Double Exclamation

- The double exclamation marks (!!) in JavaScript/TypeScript convert any value into a boolean (true or false).

- Example:

```
isLoggedIn(): boolean {  
    return !!this.getToken(); // Returns true if token exists, false  
}
```

- getToken(): string | null retrieves the token from localStorage:

```
getToken(): string | null {  
    return localStorage.getItem('token');  
}
```

## 6 jwt-decode Library

- A library to decode JWT tokens in Angular.
- Installation: `npm install jwt-decode`.
- Used to extract payload information from a JWT token.

## 7 HTTP Interceptor

- Automatically runs on every HTTP request to modify or handle requests (e.g., adding JWT token to headers).
- Register once in `app.config.ts` or `main.ts` for application-wide use.