REST Assured Headers and Cookies Overview

🔐 Headers and Cookies in Web Automation & Testing (e.g., Selenium)

📌 What are Headers?

HTTP Headers are key-value pairs sent between client and server in HTTP requests and responses. They carry metadata.

✅ Examples of Request Headers:

- **`User-Agent`**: Identifies the browser or tool (e.g., Chrome, Postman).
- **`Accept`**: Types of data the client can handle (e.g., **`text/html`**).
- **`Authorization`**: Used to pass access tokens or API keys.
- **`Content-Type`**: Tells the server the format of the body (e.g., **`application/json`**).

```
h


GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Accept: text/html
```

📌 What are Cookies?

Cookies are small pieces of data sent by the server to store on the client and sent back with future requests. They help maintain **session state**.

✅ Example:

- **`sessionId=xyz123; expires=Fri, 30 Aug 2025 12:00:00 GMT; Path=/;`**

🔄 Cookies are used for:

- Session tracking (login, shopping cart)
- User preferences
- Authentication tokens

🧪 Using Headers and Cookies in Selenium (Java)

✅ Set Cookies in Selenium:

```java
import org.openqa.selenium.Cookie;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

WebDriver driver = new ChromeDriver();
driver.get("https://example.com");

// Add a cookie
Cookie cookie = new Cookie("sessionId", "xyz123");
driver.manage().addCookie(cookie);

// Get all cookies
Set<Cookie> cookies = driver.manage().getCookies();
for (Cookie c : cookies) {
    System.out.println(c.getName() + " = " + c.getValue());
}
```

❌ Selenium does not support setting custom headers directly, but you can:

- Use **browser devtools protocol** (CDP) with **Chrome** to inject headers.
- Use **RestAssured** or **HttpClient** for API testing with custom headers.

🛠️ Set Headers Using CDP (Selenium 4+ with Chrome)

```java
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.devtools.DevTools;
import org.openqa.selenium.devtools.v85.network.Network;

ChromeDriver driver = new ChromeDriver();
DevTools devTools = driver.getDevTools();
devTools.createSession();

Map<String, Object> headers = new HashMap<>();
headers.put("Custom-Header", "MyValue");

devTools.send(Network.enable(Optional.empty(), Optional.empty(), Optional.empty()));
devTools.send(Network.setExtraHTTPHeaders(new Headers(headers)));

driver.get("https://example.com");
```

⚖️ Difference: Headers vs Cookies

Feature	Headers	Cookies
Purpose	Metadata about request/response	Store session/user data
Set By	Client or server	Server (and JS)
Storage	Temporary (per request)	Stored in browser
Scope	Applies to a single request	Sent with every request to same domain
Modifiable?	Yes, via code or tools	Yes, via browser/devtools or code

---

Would you like hands-on **Java code examples** for cookies and headers in both **UI and API testing** (e.g., with RestAssured)?

Here’s how to **add single or multiple headers** in different contexts:

✅ 1. In API Testing (RestAssured - Java)

🔹 Add a single header:

```java
given()
    .header("Authorization", "Bearer token123")
.when()
    .get("https://api.example.com/data")
.then()
    .statusCode(200);
```

🔹 Add multiple headers using `.headers(...)`:

```java
given()
    .headers(
        "Authorization", "Bearer token123",
        "Content-Type", "application/json",
        "Custom-Header", "MyValue"
    )
.when()
    .get("https://api.example.com/data")
.then()
    .statusCode(200);
```

🔹 Add multiple headers using `Map`:

```java
Map<String, String> headers = new HashMap<>();
headers.put("Authorization", "Bearer token123");
headers.put("Content-Type", "application/json");
headers.put("Custom-Header", "MyValue");

given()
    .headers(headers)
.when()
    .get("https://api.example.com/data")
.then()
    .statusCode(200);
```

✅ 2. In Selenium UI Testing (Selenium 4+ with Chrome DevTools Protocol)

Selenium **does not support setting headers directly**, but **Chrome DevTools Protocol (CDP)** does.

🔹 Add a single or multiple headers using CDP:

```java
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.devtools.DevTools;
import org.openqa.selenium.devtools.v85.network.Network;
import org.openqa.selenium.devtools.v85.network.model.Headers;

import java.util.*;

ChromeDriver driver = new ChromeDriver();
DevTools devTools = driver.getDevTools();
devTools.createSession();

// Create custom headers
Map<String, Object> headerMap = new HashMap<>();
headerMap.put("Authorization", "Bearer token123");
headerMap.put("Custom-Header", "MyValue");

// Enable network and set headers
devTools.send(Network.enable(Optional.empty(), Optional.empty(), Optional.empty()));
devTools.send(Network.setExtraHTTPHeaders(new Headers(headerMap)));

// Navigate with headers
driver.get("https://example.com");
```

> 💡 Make sure you use the correct `Network` version (e.g., `v85`) matching your ChromeDriver version.

📌 Summary Table

Context	Add Single Header	Add Multiple Headers
RestAssured	`.header("Key", "Value")`	`.headers(map)` or `.headers("k1", "v1", "k2", "v2")`
Selenium CDP	Use `Map.put("Key", "Value")`	Add multiple key-value pairs to `Map<String, Object>`

Would you like examples of **headers in Postman**, **Python requests**, or **curl commands** as well?

Working with **session cookies** is essential for maintaining **user sessions**, especially in login-based applications. Here's how to **view, add, retrieve, and use session cookies** in various contexts:

🔹 What are Session Cookies?

- **Session cookies** are stored in memory and expire when the browser closes.
- Used for maintaining session state like login status (e.g., **`JSESSIONID`**, **`PHPSESSID`**).

✅ In Selenium (Java)

1️⃣ Get Cookies After Login (session capture):

```java
WebDriver driver = new ChromeDriver();
driver.get("https://example.com/login");

// Login steps (e.g., enter username/password and click login)
// ...

// After login, capture session cookie
Set<Cookie> cookies = driver.manage().getCookies();
for (Cookie cookie : cookies) {
    System.out.println(cookie.getName() + " = " + cookie.getValue());
}
```

2️⃣ Use Captured Session Cookie in Another Session:

```java
WebDriver driver2 = new ChromeDriver();
driver2.get("https://example.com"); // Must load domain first

Cookie sessionCookie = new Cookie("JSESSIONID", "yourSessionID");
driver2.manage().addCookie(sessionCookie);

driver2.navigate().refresh(); // Now session should be active
```

> 🔐 Useful for testing logged-in states without repeating login every time.

✅ In API Testing (RestAssured - Java)

1️⃣ Extract session cookie from login:

```java
Response response = given()
    .formParam("username", "admin")
    .formParam("password", "admin123")
.when()
    .post("https://example.com/login");
```

```
String sessionId = response.getCookie("JSESSIONID");
System.out.println("Session ID: " + sessionId);
```

2️⃣ Send session cookie in subsequent requests:

```java
given()
    .cookie("JSESSIONID", sessionId)
.when()
    .get("https://example.com/dashboard")
.then()
    .statusCode(200);
```

✅ Using Postman (Manual Testing)

- After logging in via a POST request, go to **Cookies tab**.
- Copy the session cookie (e.g., **`JSESSIONID`**).
- Use it in subsequent requests via the **Cookies** tab or **`Headers`** (**`Cookie: JSESSIONID=...`**).

✅ Using curl (CLI Testing)

Store cookie to a file:

```bash
curl -c cookies.txt -X POST -d "user=admin&pass=admin123" https://example.com/login
```

Use stored cookie in another request:

```bash
curl -b cookies.txt https://example.com/dashboard
```

🧠 Summary Table

Action	Selenium	RestAssured	Postman / curl
Capture session cookie	**`driver.manage().getCookies()`**	**`response.getCookie()`**	View in Cookies tab / -c
Use session cookie	**`driver.manage().addCookie()`**	**`.cookie("key", "value")`**	Set manually / **`-b cookies.txt`**
Keep session alive	Same browser instance	Same session token	Use session cookie

Would you like a **real working demo project** with Selenium login + session cookie reuse, or combine with **API + UI test flow**?