HTTP Header Manager & Authentication in JMeter

1️⃣ Why Headers Matter in API Testing

In modern architectures:

- APIs **do not trust anyone**
- Almost everything is controlled via **headers**

If headers are wrong:
❌ API may return 401
❌ API may return 403
❌ API may behave incorrectly

👉 **Most JMeter failures are header issues**

2️⃣ What is HTTP Header Manager?

The **HTTP Header Manager** allows JMeter to attach headers to every HTTP request under its scope.

Think of it as:

> "Common envelope attached to all API calls"

3️⃣ Where to Add Header Manager (Best Practice)

✅ Add under:

```
Thread Group
  ├── HTTP Header Manager
  ├── Login API
  ├── Profile API
  ├── Submit API
```

❌ Avoid adding separately under each sampler unless needed.

4️⃣ Most Common Headers

Mandatory for JSON APIs

```
Content-Type: application/json
Accept: application/json
```

Authentication Header

```
Authorization: Bearer <token>
```

Correlation / Tracing (Enterprise APIs)

```
x-correlation-id: 123456
x-request-id: abc-xyz
```

5️⃣ Authentication Patterns in Real Systems

1️⃣ Static Token (Rare, Non-Prod)

```
Authorization: Bearer abc.def.ghi
```

⚠️ Only used for:

- Dev environments
- POCs
- Internal testing

2️⃣ Login API → Token (Most Common)

Step 1: Login API Response

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIs...",
```

```
  "expires_in": 3600
}
```

Step 2: Extract Token

Use **JSON Extractor**:

```
Variable Name: access_token
JSON Path: $.access_token
```

Step 3: Use Token in Header

```
Authorization: Bearer ${access_token}
```

6️⃣ CIAM / OAuth2 Flow (Enterprise Level)

Typical Flow

1. Client sends credentials
2. CIAM validates user
3. CIAM issues access token
4. Token sent to downstream APIs

OAuth Token Request Example

```
POST /oauth/token
```

Body:

```
{
  "grant_type": "password",
  "client_id": "abc123",
  "client_secret": "secret",
  "username": "user@test.com",
  "password": "pass"
}
```

7️⃣ Where Students Get Confused

❓ “How does login work without OTP?”

Answer:

- OTP is **business logic**
- Load test usually:
  - Bypasses OTP
  - Uses test users
  - Uses mocked tokens

👉 **Performance tests test APIs, not humans**

8️⃣ OTP-Based Login – How to Handle in JMeter

❌ What NOT to Do

- Wait for OTP email
- Manually type OTP
- Use real email inbox

✅ What Companies Actually Do

Approach	Used In
Pre-generated OTP	Lower env
OTP disabled flag	Perf env
Token-only auth	Load tests
Service accounts	Most common

9️⃣ Header Manager vs Sampler Headers

Header Manager

✓ Reusable
✓ Clean
✓ Maintainable

Sampler-level Headers

❌ Repetitive
❌ Hard to manage

👉 Use sampler headers **only if unique**

🔟 Dynamic Headers Using Variables

Example:

```
x-user-id: ${userId}
x-session-id: ${sessionId}
```

These variables can come from:

- CSV file
- Extractors
- Groovy script

1️⃣1️⃣ Debugging Auth Issues

Use **View Results Tree**:

- Check Request Headers
- Verify token present
- Verify Bearer prefix
- Check expiry

1️⃣2️⃣ Performance Tip

❌ Don't re-login on every request
✅ Login once per thread
✅ Reuse token

This reduces:

- Load on auth systems
- Noise in test results

1️⃣3️⃣ Interview Question

> Why should login not be part of peak load?

Answer:
Because authentication systems are usually sized differently and can skew business API performance metrics.

1️⃣4️⃣ Mini Exercise

1. Call Login API
2. Extract access token
3. Add Header Manager
4. Pass token dynamically
5. Call secured API
6. Verify response

Summary

✓ Header Manager controls request identity
✓ Tokens are dynamic, never hardcoded
✓ OTP is bypassed in performance testing
✓ Auth systems are not load-tested blindly
