API Testing Cheat Sheet

— A Guide by **Usman Qiass** —

What is API Testing?

Testing APIs to validate **functionality**, **reliability**, **performance**, and **security** of application endpoints.

🛠️ Why it matters

- APIs connect frontend ↔ backend
- Detects bugs early in data flow
- Faster than UI testing

🧪 Common Types

- Functional Testing
- Integration Testing
- Load Testing
- Security Testing

Key API Testing Concepts

1. HTTP Methods
2. Status Codes
3. Authentication Methods
4. Common Validation Checks
5. Headers to Verify

1. HTTP Methods

- GET: Retrieve data
- POST: Create data
- PUT: Update data
- DELETE: Remove data

2. Status Codes

200 OK: Success

201 Created: Resource added

204 No Content: Success, no data returned

400 Bad Request: Invalid input

401 Unauthorized: Missing token

403 Forbidden: Access denied

404 Not Found: Resource missing

409 Conflict: Duplicate or version error

500 Internal Server Error: Backend failure

503 Service Unavailable: Server overloaded/down

3. Authentication Methods

- **API Key** → Simple but less secure
- **Bearer Token (JWT)** → Common & stateless
- **OAuth 2.0** → Secure and preferred for user-based apps
- **Basic Auth** → Username + password (not recommended for production)

4. Common Validation Checks

- 🔍 Status code is correct
- 📦 Response body contains expected fields
- ⏱️ Response time is within limits
- 🔐 Authentication & authorization work as expected
- 🧾 Data formats (JSON, XML) are valid

5. Headers To Verify

- Content-Type: Ensure correct format (e.g., application/json)
- Authorization: Required for secured APIs
- Cache-Control: Check caching behavior
- Set-Cookie: Relevant for session-based auth

Tools You Should Know

🛠️ **Postman** – Manual & automated testing with collections

⚙️ **Swagger** – API documentation and try-it-out console

🐍 **Rest Assured** – Java-based API testing

🚀 **JMeter** – Load testing of REST/SOAP APIs

🌐 **Insomnia** – Lightweight Postman alternative

🧪 **SoapUI** – Ideal for SOAP API testing

✅ Best Practices

- Always test both positive & negative scenarios
- Use environment variables for different stages (dev/stage/prod)
- Validate response schema & headers
- Automate tests in CI/CD pipelines

💡 Pro Tips

- Use mocks/stubs when backend isn't ready
- Monitor rate limits & throttling
- Always check error handling & fallback logic

— A Guide by **Usman Qiass** —

♻️ Repost this if you learned something new.

👉 Follow Usman Qiass for more QA insights, testing tips, and real-world examples.