◇ Auth API QA Testing Cheat Sheet

1 API Testing Basics

- **API:** Interface for apps to communicate; testing ensures correct functionality.
- **Functional vs Non-functional:** Functional = correctness; Non-functional = performance, reliability.
- **HTTP Methods:** GET (read), POST (create/submit), PUT/PATCH (update), DELETE (remove). POST is used for login/token creation.
- **REST vs Others:** REST = HTTP/verbs, stateless; GraphQL = flexible queries; SOAP = XML-based.
- **Status Codes:** Verify API returns correct codes (200 OK, 401 Unauthorized, 500 Error).

2 Auth & Security

- **Authentication vs Authorization:** Auth = identity verification; AuthZ = access control.
- **Positive vs Negative Tests:** Positive = valid credentials succeed; Negative = invalid inputs fail safely.
- **Token:** Unique ID (UUID, JWT) for sessions; ensures stateless and secure access.
- **Token Validation:** Check format, expiration, storage in DB to prevent reuse and ensure security.
- **Security Risks:** Wrong credentials must be rejected; improper handling can lead to breaches.

3 Input Validation

- Test **empty fields, missing fields, extra fields** to check API robustness.
- **Input Sanitization:** Prevents injection attacks; essential for server-side security.
- **Client vs Server Validation:** Client-side = UX; Server-side = security + correctness.
- Malformed requests must not crash API.

4 Performance & Non-functional

- **Response Time:** Measure speed (< 2 seconds typical for login).
- **Causes of Slowness:** DB latency, heavy server load, network issues, inefficient code.
- **Load Testing:** Simulate concurrent requests to verify performance under stress.

5 Error Handling & Response Structure

- **Consistent structure:** Success → token; Failure → reason.
- Handle missing fields, empty body, invalid credentials safely.
- Returning HTTP 200 for failed login is okay if error explained; 401 is more REST-standard.
- API contract ensures predictable integration for frontend and other clients.

6 Testing Tools & Automation

- **Playwright:** Automation for web & API testing; supports assertions, request/response handling.
- **Key Functions:** `test()` = define test, `expect()` = assertion, `request.post()` = send POST request.
- **Automation Benefits:** Faster, repeatable, reduces human error.
- **CI/CD Integration:** Run tests via GitHub Actions for automated validation on each push or PR.

7️⃣ Advanced QA Considerations

- **Token Expiration:** Test short-lived tokens; ensure API rejects expired tokens.
- **Brute-force Testing:** Multiple failed logins → account lockout/rate limiting.
- **Mock DB:** Use in-memory DB or mocked responses for faster tests.
- **Reproducibility:** Consistent environment, test DB, versioned endpoints.
- **Reports:** Playwright HTML/JSON reports show status, request/response, performance metrics.

✅ **Tip:** For Auth API testing, always include **positive, negative, input validation, security, and performance** checks in your test plan.