

Requests Library – Full Practical Practice Test (Data Analytics Automation)

SECTION 1: BASIC HTTP OPERATIONS

- Q1 Send a GET request to a public API and print status code, headers, and first 300 characters.
- Q2 Fetch JSON data, convert to dict, print top-level keys.
- Q3 Send GET with query parameters and verify response change.
- Q4 Validate status code and print success or failure.
- Q5 Detect JSON vs non-JSON response.

SECTION 2: HEADERS & AUTHENTICATION

- Q6 Send GET with custom headers.
- Q7 Extract and print specific response header.
- Q8 API key authentication using headers.
- Q9 Bearer token authentication.
- Q10 Handle 401/403 and log error.

SECTION 3: POST, PUT, DELETE

- Q11 POST request with JSON payload.
- Q12 POST request with form data.
- Q13 PUT request update.
- Q14 DELETE request validation.
- Q15 Dynamic POST payload.

SECTION 4: JSON HANDLING

- Q16 Extract nested JSON values.
- Q17 Convert JSON to list of dicts.
- Q18 Handle missing keys safely.
- Q19 Flatten nested JSON.
- Q20 Save JSON to file.

SECTION 5: ERROR HANDLING

- Q21 Handle timeout, connection, HTTP errors.
- Q22 Use timeout parameter.
- Q23 Retry failed request 3 times.
- Q24 Handle invalid JSON.
- Q25 Handle empty responses.

SECTION 6: FILE DOWNLOADS

- Q26 Download CSV file.
- Q27 Stream large file download.
- Q28 Validate file size.
- Q29 Delete partial failed download.
- Q30 Timestamped daily download.

SECTION 7: PAGINATION

Q31 Page-number pagination.

Q32 Offset-limit pagination.

Q33 Fetch all pages automatically.

Q34 Combine paginated results.

Q35 Save paginated data to CSV.

SECTION 8: SESSIONS

Q36 Use requests.Session.

Q37 Compare session vs normal request time.

Q38 Persist cookies.

Q39 Close session properly.

Q40 Handle session expiration.

SECTION 9: LOGGING

Q41 Log every request.

Q42 Log failed requests.

Q43 Capture response time.

Q44 Daily API usage summary.

Q45 Mask sensitive headers.

SECTION 10: ANALYTICS AUTOMATION

Q46 Fetch daily sales data.

Q47 Merge multiple API data sources.

Q48 Scheduled API fetch.

Q49 Detect anomalies.

Q50 API health check.

SECTION 11: ADVANCED

Q51 Handle rate limiting.

Q52 Batch API requests.

Q53 Validate API schema.

Q54 Build reusable API client.

Q55 Config-driven API calls.

SECTION 12: FINAL PROJECT

Q56 End-to-end automation: pagination, retries, logging, CSV output.