**Documented Test Case — T-4**

**Test ID:** T-4  
**Title:** GIPHY Search API returns 200 and honors q, limit, offset  
**Feature/Area:** Search — Backend API  
**Type:** Automated (API)  
**Priority:** High (P1)  
**Severity (on failure):** Blocker  
**Owner:** QA Automation  
**Preconditions:**

* Internet connectivity is available.
* API key is available (either set via GIPHY\_API\_KEY env var **or** fallback key in the test).
* Test runner and dependencies installed (Ruby ≥ 3.4, RSpec ≥ 3.13, httparty).

**Environment:**

* Endpoint: GET https://api.giphy.com/v1/gifs/search
* Headers: none required (API key passed as query param)
* Auth: API key (query string api\_key)

**References / Traceability:**

* TRM: **Req-4** — “Backend search API respects q, limit, offset”
* Acceptance criteria in TRM: HTTP 200; correct data size & pagination values

**Test Data**

|  |  |
| --- | --- |
| **Field** | **Value** |
| q | cat |
| limit | 5 |
| offset | 2 |

**Steps & Expected Results**

1. **Send request** to /v1/gifs/search with params { api\_key, q="cat", limit=5, offset=2 }.  
   **Expected:** Server responds; status code is **200 OK**.
2. **Parse JSON** response body.  
   **Expected:** Top-level keys present: **data** (Array), **pagination** (Object), **meta** (Object).
3. **Validate item count.**  
   **Expected:** data.length == 5 (matches limit).
4. **Validate pagination values.**  
   **Expected:**
   * pagination.offset == 2
   * pagination.count == 5
   * pagination.total\_count exists and is an integer (≥ count)
5. **(Soft check, non-blocking)** Validate meta status.  
   **Expected:** meta.status == 200.

**Acceptance Criteria (Pass/Fail)**

* **Pass** if Steps 1–4 expectations hold (status 200; correct shape; counts match limit; offset matches).
* **Fail** if:
  + HTTP status is not 200, or
  + data is not an array, or
  + data.length != limit, or
  + pagination.offset != offset or pagination.count != limit.

**Artifacts (for traceability/triage)**

* **Raw response JSON** saved as: artifacts/search\_T4\_<timestamp>.json
* **Meta file** with endpoint, masked key, params, status, counts: artifacts/search\_T4\_<timestamp>\_meta.txt
* Optional “latest” copies: artifacts/search\_T4\_last.json, artifacts/search\_T4\_last\_meta.txt

**Risks & Notes**

* API is rate-limited; transient 429/5xx may occur. A single re-run can be allowed in CI, but persistent failures should fail the build.
* Dataset is dynamic; we assert **structure and pagination math**, not exact GIF content.
* Using a fallback key in the test removes setup friction for reviewers; ENV can override.

**Maintenance / Parameterization**

* q, limit, offset can be parameterized (e.g., via RSpec let or ENV) to broaden coverage.
* Consider adding a separate negative test for invalid keys (expect 401/403 with friendly error/log).