**PharmGKB public API tests doc/ Zipora Gottesman**

**Critical tests on service:**

1. **Health check** – Ensure the API is reachable and returns basic data.

**Example**: GET /v1/data/gene?limit=1

**Expected**: HTTP 200 OK, valid JSON format and at least one result

1. **Search by Gene symbol** – Verify that searching by a known symbol returns a structured gene object.

**Example**: GET /v1/data/gene?symbol=CYP2C19

**Expected**: HTTP 200 OK, full structured gene object.

1. **Fetch by ID** – Confirm that fetching by a known ID returns the same full gene object.

**Example**: GET /v1/data/gene/PA134864839

**Expected**: HTTP 200 OK, full structured gene JSON object

1. **Search-to-ID Consistency Check**– Verify that a search result’s ID can be used to fetch the same object with matching key fields.

**Example**: Search by symbol= CYP2C19 -> extract geneId -> fetch by ID

**Expected:** Same values for symbol, name and gene Id

1. **Invalid ID handling** – Ensure that non-existent or invalid IDs return controlled error responses.

**Example**: GET /v1/data/pathway/INVALID\_ID

**Expected**: HTTP 404 Not Found, with a clear error message in response

**Additional tests for PharmGKB API:**

1. **Rate limit enforcement** – confirm that sending more than 2 requests per second triggers an HTTP (429 Too Many Requests).

**Example**: send 10 requests/second

**Expected**: HTTP 429 response

1. **Cross reference completeness** – check that returned objects include populated cross-references arrays.

**Example**: GET /v1/data/gene/CYP2C19

**Expected:** – crossReferences contains at least one entry

1. **Field schema validation** – validate that key fields in the response match expected types and presence.

**Example**: GET /v1/data/gene/CYP2C19

**Expected**: Fields like id, name and symbol are non-empty strings

1. **Empty search handling** – Ensure that missing or empty query parameters return proper controlled response.

**Example**: GET /v1/data/gene?symbol

**Expected:** HTTP 400 Bad Request or result set with valid JSON

1. **Minimum Pediatric Annotations** – Ensure that the gene has a minimum number of annotations where pediatric is true.

**Example**: GET /v1/data/clinicalAnnotation?location.genes.symbol=CYP2C19

Then filter: entry.pediatric === true

**Expected:** At least N entries in the response have pediatric === true