**Users API – Test Plan**

**Objective**: Validate Users API CRUD functionality, contract (schema), error handling, and basic performance/security using the public Fake Store API (`https://fakestoreapi.com`).

**Scope**

* **In scope**: `GET /users`, `GET /users/{id}`, `POST /users`, `PUT|PATCH /users/{id}`, `DELETE /users/{id}`
* **Optional**: query params (limit/sort). Schema validation and negative tests are included.
* **Out of scope**: UI, non-user resources.

**Assumptions**

* Environment is reachable.
* Public demo data may reset.
* Tests will be self-contained and clean up created entities.

**Environment & Config**

|  |  |
| --- | --- |
| Base URI | https://fakestoreapi.com |
| Headers | Content-Type: application/json; Accept: application/json |
| Auth | None for demo; if provided, pass via request spec |
| Data management | Create → Verify → (Optional) Delete within scenario |

**Test Types**

* Functional (positive & negative)
* Contract / JSON schema.

**Entry / Exit**

* Entry: endpoints up; data strategy agreed.
* Exit: all P0/P1 pass; no Sev-1/2 open; test evidence (report) attached.

**Smoke Pack (for demo)**

T01 GET all 200

T02 GET id 200

T03 POST create 201

T04 PUT 200

T05 DELETE 200/204

Test Scenarios (Condensed)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Title | Steps (summary) | Expected |
| T01 | GET all users returns 200 | GET /users | 200; JSON array; not empty |
| T02 | GET all matches schema | GET /users | Fields/types match list schema |
| T03 | GET with limit | GET /users?limit=5 | 200; length ≤ 5 |
| T04 | GET single user | GET /users/1 | 200; id=1 |
| T05 | GET single schema | GET /users/1 | Matches single-user schema |
| T06 | GET non-existing | GET /users/999999 | 404 or empty per spec |
| T07 | GET invalid id | GET /users/abc | 400/404 |
| T08 | POST create minimal/mandatory fields | POST /users valid body | 201; id generated |
| T09 | POST create full | POST /users full payload | 201; persisted fields |
| T10 | POST missing required | Omit required | 400 validation |
| T11 | POST idempotency | Repeat same body | New id (POST not idempotent) |
| T12 | PUT update | PUT /users/{id} | 200; updated fields |
| T13 | PATCH update | PATCH /users/{id} | 200; partial update |
| T14 | PUT invalid type | Wrong types | 400 |
| T15 | PUT non-existing | PUT /users/999999 | 404 |
| T21 | DELETE existing | DELETE /users/{id} | 200/204; next GET 404 |
| T22 | DELETE non-existing | DELETE /users/999999 | 404 |
| T23 | DELETE invalid id | DELETE /users/abc | 400/404 |

**Sample Payloads**

**Valid minimal**: {"email":"neo@example.com","username":"neo","password":"Pa$$w0rd1"}

**Full**: includes name/address/geolocation/phone per FakeStore docs **Invalid**: missing fields, malformed email, wrong types, duplicates

**cURL Samples**

GET: curl -s https://fakestoreapi.com/users

POST: curl -s -X POST https://fakestoreapi.com/users -H "Content-Type: application/json" -d

'{"email":"neo@example.com","username":"neo","password":"Pa$$w0rd1"}'