# Test Plan – AI Coach Agent: POST /generate-plan/

**1) Overview**

* **Purpose:** Validate the API that generates a multi-step swap plan to reach a **target item** given user **initial items** and **interests** within max\_steps.
* **Under Test:** POST https://aicoach-714849252637.europe-west1.run.app/generate-plan/
* **Artifacts:** Postman Collection (**AI\_Coach\_Generate\_Plan.postman\_collection.json**) + Environment

**2) Objectives & Success Criteria**

* Return a valid plan that **reaches the target** within max\_steps.
* Enforce **input validation** and return clear errors.
* Maintain **performance** (p95 < **2s**) and **stability** under bursts (graceful 429).
* Contract: response matches agreed **JSON schema** (plan metadata + ordered steps).

**3) Scope**

**In-scope**

* Functional correctness & schema compliance
* Validation/error handling
* Basic robustness (rate limiting, determinism baseline)

**4) Requirements → Test Coverage (Traceability)**

| **Requirement** | **Description** | **Covered by** |
| --- | --- | --- |
| R1 | Must generate plan reaching target\_item within max\_steps | **TC1, TC2, TC4** |
| R2 | If target already owned, return 0 steps / equivalent message | **TC3** |
| R3 | Validate inputs (required fields, types, non-empty lists, max\_steps rules) | **TC5, TC6, TC7, TC9** |
| R4 | Boundary behavior (max\_steps = 1) | **TC8** |
| R5 | Handle Unicode safely | **TC10** |
| R6 | Malformed JSON → clear 400 | **TC11** |
| R7 | Consistent outcomes (determinism baseline) | **TC12** |
| R8 | Handle bursts gracefully (200 or 429 with retry-after) | **TC13** |
| R9 | Performance p95 < 2s | Assertions in **TC1–TC4** |

**5) Test Approach**

* **Tooling:** Postman/Newman with embedded assertions (status, schema, latency, logical checks).
* **Execution:** Local or CI. Rate-limit test executed via runner iterations.
* **Pass/Fail:** Critical tests (R1–R4, R6) must be green. Non-critical (R5, R7–R9) ≥ 95% pass.

**6) Test Environment**

* **Environment file:** *AI\_Coach\_Env.postman\_environment.json*
  + base\_url: API host
  + determinism\_signature: baseline for TC12

**7) Test Cases Summary (from collection)**

**Functional – Success**

* **TC1** Happy path – multiple items → 200, schema OK, total\_steps ≤ 5, last step = target
* **TC2** Single initial item → same assertions
* **TC3** Target already owned → 200, total\_steps = 0 or equivalent
* **TC4** Duplicate/normalized items → still valid plan

**Validation & Errors**

* **TC5** Missing target\_item → 400 BAD\_REQUEST
* **TC6** Wrong types → 400 BAD\_REQUEST
* **TC7** Empty initial\_items → 400
* **TC8** Boundary max\_steps = 1 → 200 (≤1 step)
* **TC9** Invalid max\_steps (0) → 400
* **TC10** Non-ASCII/Unicode items → JSON response; no crash
* **TC11** Malformed JSON → 400

**Robustness**

* **TC12** Determinism baseline → same last target & step count across runs
* **TC13** Rate-limit burst → 200 or 429 with Retry-After

*All success tests also assert p95 latency < 2s.*

**9) Entry & Exit Criteria**

**Entry**

* API deployed & reachable
* Collection + environment imported

**Exit**

* 100% pass on TC1–TC4, TC5, TC6, TC7, TC9, TC11
* ≥95% pass overall; no severity-1 defects open
* HTML report generated and shared

**10) Execution & Reporting**

**Run in Postman:** select **AI Coach – Env** → run collection.  
**Run in Newman (CI or local):**

newman run "AI\_Coach\_Generate\_Plan.postman\_collection.json" \

-e "AI\_Coach\_Env.postman\_environment.json" \

-r cli,htmlextra \

--reporter-htmlextra-export "AI\_Coach\_Test\_Report.html" \

--reporter-htmlextra-title "AI Coach API – Test Report"

**Deliverables**

* **AI\_Coach\_Test\_Report.html** (summary, pass/fail, timings)
* Defect tickets (if any) with failing request/response payloads

**KPIs to show**

* Pass rate by category
* p95 response time on happy paths
* Error distribution (400/422/429/5xx)