

Save, retrieve, and retry a text note with Forge

Personal project documentation sample · Matthew Kotys · October 13, 2026

Use Forge's existing FastAPI service to save a synthetic text note, retrieve it, and retry the submission without creating a duplicate.

Before you begin

- Have Forge running with its entries router and database configured.
- Use a Bash-compatible terminal with `curl`, on a network that can reach the service.
- Know the service's base URL, including its port. This guide uses a placeholder to avoid publishing a private network address.
- Expect the POST to save one test note. This guide assumes an existing installation; it does not cover installation or authentication. The tested deployment required no credentials.

1. Set the service address

Replace the placeholder with your actual base URL. Do not include a trailing slash.

```
export FORGE_URL='http://<forge-host>:<port>'
```

Keep this terminal open for the remaining commands.

2. Save a text note

Run the following request. For a fresh walkthrough, replace `forge-howto-test-001` with a new client identifier. Keep that identifier unchanged when testing retries.

```
curl -i -X POST "$FORGE_URL/entries" \
-H 'Content-Type: application/json' \
--data '{
  "kind": "text",
  "category": "documentation-test",
  "text": "Testing the Forge how-to guide.",
  "client_id": "forge-howto-test-001",
  "created_at": "2026-10-13T22:25:00Z"
}'
```

The fixed timestamp reproduces the tested request. For a current note, use a current ISO 8601 timestamp with a timezone: Z represents UTC.

Expect HTTP/1.1 200 OK and a JSON entry. The first tested request returned:

```
{
  "category": "documentation-test",
  "kind": "text",
  "text": "Testing the Forge how-to guide.",
  "received_at": "2026-10-13T22:25:17.868016Z",
  "id": "4638ae5c-2207-4b0f-a7eb-06dd2b521f08",
  "media_path": null,
  "client_id": "forge-howto-test-001",
```

```
"created_at": "2026-10-13T22:25:00Z"
}
```

Your generated ID and receipt timestamp will differ for a newly saved entry. JSON field order does not matter.

Field	Meaning
client_id	Caller-provided identifier used to recognize repeat submissions.
created_at	Caller-provided creation time; a timezone is required.
id	Server-generated UUID for the stored entry.
received_at	Server-generated receipt time for a new entry.
media_path	Optional media location; omitted in this request and returned as null.

3. Retrieve the note

```
curl -i "$FORGE_URL/entries?offset=0&limit=20"
```

Expect 200 OK and an object containing `items`, `offset`, `limit`, and `total`. Find your `client_id` inside `items` and compare its `id` with the POST response.

The tested deployment returned the same entry, with `offset: 0`, `limit: 20`, and `total: 1`. An existing database can contain more entries.

The list sorts by `created_at` descending, not by receipt time. If the note is outside the first page, increase `offset` to 20, then 40, and so on. The default limit is 20; permitted limits are 1–100. `total` counts all stored entries, not just the current page.

4. Retry the same submission

Run the POST command from step 2 again without changing its `client_id`. Then run the GET command again.

The current route checks for an existing `client_id` before inserting. A repeat submission returns the existing entry with 200 OK. It does not update that entry, even if other submitted fields change. Use a new client identifier for a new note.

Confirm that only one entry with the test identifier remains. The overall count should stay unchanged if no other requests are adding notes. Comparing the repeated POST's `id` and `received_at` with the initial response provides an additional check.

Troubleshooting

Symptom	Next check
Connection refused or timeout	Confirm the address and port, that Forge is running, and that this device can reach it.
HTTP 422	Check JSON syntax, required fields (<code>kind</code> , <code>category</code> , <code>client_id</code> , <code>created_at</code>), allowed kind values, and the timestamp's timezone.
Note absent from first page	Check the submitted <code>created_at</code> and page through the descending list.
Repeated POST returns older text	Check whether the <code>client_id</code> already exists; retries return the stored entry rather than update it.
GET <code>/entries/<UUID></code> fails	Use the list endpoint. In the reviewed source, the individual-entry route is an integer-parameter placeholder, not a working UUID lookup.

Verification record and scope

The developer ran these requests against the existing Forge deployment on October 13, 2026 and supplied the results:

- **Create:** POST returned HTTP 200, a generated UUID, and a receipt timestamp. Omitted `media_path` returned null.
- **Retrieve:** GET returned HTTP 200 with the same entry ID and a total count of one.
- **Retry:** The developer reported subsequent POSTs returned HTTP 200 and the database list still contained only the original entry. Exact ID and timestamp equality after retries was not separately supplied.

The request schema, pagination rules, descending sort, and return-existing retry behavior were also reviewed in the supplied models and route code. Changed-payload retries, concurrent submissions, error cases, and pagination across multiple pages were not exercised. The guide does not claim those checks passed.

This sample documents a personal project. It excludes Opportunity models, personal note contents, private network addresses, iPhone automation, service installation, and authentication setup.