Memoir.ai — API Reference Manual

Backend, Sync, and IPC Interface Specifications

Document Version: 1.0

Status: Engineering Reference

Owner: Platform API Team

Last Updated: YYYY-MM-DD

----

1. Purpose

This API Reference Manual documents all backend, synchronization, and inter-process communication interfaces used within Memoir.ai. It serves as the canonical engineering reference for implementing frontend integration, backend services, and external integrations.

----

2. Scope

This manual covers:

- Local REST API endpoints
- IPC communication interfaces
- Sync service endpoints
- Billing and entitlement endpoints
- Import and job management APIs

• Timeline and search APIs

• Snapshot generation APIs

• Export and deletion APIs

• Media APIs

• Event and webhook schemas

• Error handling formats

UI implementation details are out of scope.

----

3. Communication Model

Memoir.ai uses three primary communication paths:

1. IPC Bridge

   Renderer ↔ Backend for vault operations.

2. Local REST APIs

   Backend endpoints for metadata and operations.

3. Sync Service APIs

   Optional cloud metadata synchronization.

Sensitive vault operations must occur via IPC channels.

----

4. Authentication Model

Vault state determines API access:

Locked Vault:

- Only authentication endpoints accessible.
- Other endpoints return forbidden errors.

Unlocked Vault:

- All local endpoints available.
- Backend holds encryption keys in memory only.

Requests include a secure vault session token.

----

5. Standard API Response Format

All endpoints return structured JSON:

```
{
  "success": true,
  "data": {},
  "error": null,
  "meta": {
    "timestamp": "ISO-8601",
    "requestId": "uuid"
  }
}
```

Errors populate the error object with recovery guidance.

----

6. Vault APIs

POST /api/vault/init

Initialize new vault.

POST /api/vault/unlock

Unlock encrypted vault.

POST /api/vault/lock

Lock active vault session.

POST /api/vault/change-passphrase

Change vault encryption credentials.

----

7. Timeline APIs

GET /v1/timeline

Retrieve paginated timeline events.

GET /v1/timeline/count

Total event count.

GET /v1/timeline/stats

Timeline event distribution metrics.

----

8. Search APIs

POST /v1/search/query

Execute hybrid lexical and semantic search.

GET /v1/search/history

Retrieve previous queries.

Parameters include alpha weighting and result limits.

----

9. Import & Jobs APIs

POST /v1/imports/jobs

Create ingestion job.

GET /v1/imports/jobs

List ingestion history.

GET /v1/imports/jobs/:id

Retrieve job status and logs.

DELETE /v1/imports/sources/:id

Remove import source.

----

10. Snapshot APIs

POST /v1/snapshots/generate

Generate AI narrative.

GET /v1/snapshots

List snapshots.

GET /v1/snapshots/:id

Retrieve narrative and citations.

PATCH /v1/snapshots/:id

Save edits.

POST /v1/snapshots/:id/regenerate

Create new narrative version.

----

11. Media APIs

GET /v1/media

Search media assets.

GET /v1/media/:id/url

Retrieve high-resolution media URL.

GET /v1/media/:id/thumbnail

Retrieve preview image.

----

12. Export & Delete APIs

POST /v1/export/request

Initiate vault export.

GET /v1/export/status/:id

Export job status.

GET /v1/export/download/:id

Download export bundle.

POST /v1/account/delete

Begin account deletion.

POST /v1/account/delete/confirm

Confirm deletion.

----

13. Billing APIs

GET /v1/billing/status

Retrieve subscription status.

GET /v1/billing/plans

Retrieve plan options.

POST /v1/billing/checkout

Create Stripe checkout session.

DELETE /v1/billing/cancel

Cancel subscription.

----

14. Sync APIs

POST /v1/sync/heartbeat

Update cloud sync state.

GET /v1/jobs/:id/status

Check remote job status.

----

15. Event Schemas

VAULT_LOCKED Event

Emitted when vault session locks.

IMPORT_PROGRESS Event

Provides ingestion progress updates.

SNAPSHOT_GENERATED Event

Signals narrative generation completion.

----

16. Webhooks

Stripe Events:

- checkout.session.completed
- customer.subscription.deleted
- invoice.payment_failed

Webhooks must validate signatures before processing.

----

17. Error Handling & Taxonomy

Errors follow structured codes:

AUTH_* Authentication failures

VALT_* Vault integrity issues

IMPT_* Import pipeline failures

AI_* Narrative generation issues

BILL_* Billing problems

Errors include recovery guidance for user action.

----

18. Implementation Best Practices

Engineering guidance:

- Validate inputs at all endpoints.
- Sanitize IPC messages.
- Maintain backward compatibility.
- Monitor performance metrics.
- Avoid exposing sensitive data.

----

19. Conclusion

This API reference provides a consistent contract for integrating Memoir.ai backend and sync services. Maintaining strict adherence ensures platform stability, secure communication, and predictable feature evolution.