# Module XX – API Interface Specification & Runtime Integration Contract

Version: 4.1 | Classification: CTO Mandatory (Integration Readiness)

Scope: REST API, OpenAPI structure, CLI hook simulation, capsule-bound interaction model

## 0. Purpose & API Integration Goal

This module defines the standardized capsule-bound API interface model for runtime operations, capsule injections, audit streams, federation control, and certification linkage. The specification enables simulation, automation, sandbox integration, and runtime orchestration via RESTful endpoints or schema-based bindings (OpenAPI, gRPC, CLI hooks).

## 1. API Entry Points (Standardized Capsules)

The following API operations are defined (method + path):  
- `POST /capsule/manifest` → inject Manifest Capsule  
- `POST /capsule/policy` → inject Policy Rule Capsule  
- `POST /capsule/execute` → issue Execution Capsule (sandbox/local only)  
- `POST /capsule/replay` → start capsule replay (audit-bound)  
- `POST /capsule/fork` → submit alternate path (requires FAC header)  
- `POST /capsule/federation` → declare Federation Consent Capsule (FCC)  
- `POST /capsule/reentry` → trigger Reentry Capsule after GC  
- `GET /ledger/trace/{capsule\_id}` → fetch trace path and capsule class  
- `GET /audit/alert/{zone}` → get Alert Capsule feed (e.g. RAC, PCC, FDAC)  
- `GET /certification/status` → pull current CCC certification profile

## 2. OpenAPI Metadata & Response Schemas

- All API endpoints use JSON payloads structured by Capsule Schema Reference (see Module: Terminology)  
- OpenAPI 3.1.x-compatible  
- Capsule class header is required (`X-Capsule-Type`)  
- Signature is passed via `X-Capsule-Signature` (Tier ≥ 1 only)  
- Example response (trace endpoint):  
```json  
{ "capsule\_id": "CPS-90823-11F", "status": "signed", "class": "FCC", "ledger\_path": ["prev\_hash", ...] }  
```

## 3. CLI Companion & Mock Injection Support

All API paths can be mapped to CLI stubs (used in Capsule Composer or developer test tracks):  
- `capsule inject manifest --file m01.json`  
- `capsule audit trace --id CPS-AX45`  
- `capsule simulate replay --chain mock\_set\_04`  
- `capsule reentry --capsule r09.mock`  
CLI tools are optional and depend on the implementation project – not provided in reference.

## 4. Federation & Role Restriction Binding

The following API operations are role-tier restricted:  
- Tier 1+ only: `/capsule/reentry`, `/capsule/federation`, `/capsule/fork`  
- Tier 2+ only: `/capsule/replay`, `/audit/alert/{zone}`  
- Tier 3+: may submit manifest, rules, and execute requests (sandbox scope only)

## 5. Integration Use Cases

- Connect capsule governance to orchestration systems (e.g. workflow engines)  
- Feed Alert Capsules to observability dashboards  
- Integrate ledger state snapshots into compliance and reporting tools  
- Enable API-first capsule simulation and sandbox validation