URL Ledger

API & Integration Specification

The integration layer for turning URL assets into a trusted, governed, machine-readable system of record.

Specification purpose

This document defines how URL Ledger connects to analytics, search, CRM, CMS, crawl, warehouse, paid media, email, revenue, and agent systems. It is designed for engineering, implementation partners, technical buyers, and future API consumers.

Field	Definition
Document type	Technical product specification and integration guide
Primary audiences	Engineering, product, implementation partners, data teams, technical SEOs, RevOps, enterprise buyers
Platform role	System of record and system of truth for URL-level website asset value
Core principle	Read first, score carefully, govern actions, then reconcile outcomes
Status	Draft v1 for pilot, partner, and MVP planning

Table of Contents

1. Executive Integration Thesis

URL Ledger does not need to become another AI agent, SEO dashboard, BI chart, or content workflow tool. Its integration advantage is that it becomes the canonical record that all of those systems can read, update, request through, and reconcile against.

Master integration thesis

Every URL becomes an instrument with identity, lineage, performance history, attribution evidence, structural risk, governance policy, rating history, and action state. Integrations exist to keep that instrument current and trustworthy.

- Search, analytics, CRM, CMS, crawler, paid media, email, and revenue systems provide observations.
- URL Ledger normalizes those observations into URL-level assets and cluster-level portfolios.
- Ratings convert fragmented signals into decision-grade truth.
- Policy gates determine who or what can act on each asset.
- Actions write back as governed state transitions with evidence and outcome reconciliation.

What the integration layer must prove

Question	Integration answer
What exists?	Ingest sitemap, CMS exports, crawl outputs, canonical maps, redirects, and discovered URLs.
What changed?	Track crawl deltas, metadata changes, redirects, status changes, template shifts, and content updates.
What performed?	Join GSC, GA4, CRM, payments, ads, email, and warehouse metrics at URL and cluster level.
What is at risk?	Score decay, duplication, cannibalization, technical weakness, channel dependency, and AI/agent readiness.
What should happen next?	Generate governed actions: refresh, merge, redirect, protect, expand, suppress, investigate, or leave alone.
Who or what can act?	Apply role permissions, policy thresholds, risk gates, and agent capabilities before execution.
Did the action work?	Reconcile pre/post outcomes, confidence level, evidence, and next review date.

2. Integration Philosophy

Principle

Integrations should make URL Ledger harder to ignore, not harder to adopt. Start with read-only access and exports. Earn writeback rights only after the ledger proves value and governance is trusted.

Principle	Meaning	Implementation implication
Read-only first	Reduce procurement friction and security risk.	Support exports, service accounts, read-only OAuth scopes, and uploaded files.
URL is the unit of account	Every signal must resolve to a canonical URL or cluster.	Normalize URLs before scoring or attribution.
Cluster is the portfolio lens	Single URLs matter, but value often concentrates by section or intent cluster.	Support URL-to-cluster mappings and rollups.
Evidence over opinion	Every score needs a because trail.	Persist raw source, timestamp, transformation, score inputs, and confidence.

Policy before action	Agents and humans need the same rules.	Separate recommendation APIs from action APIs.
Reconciliation closes the loop	No action is complete until actual outcomes are measured.	Track before/after windows and outcome confidence.

3. Reference Architecture

ARCHITECTURE LAYERS

Layer	Primary responsibility
Source systems	GSC, GA4, CMS, crawler, sitemap, CRM, payments, ads, email, warehouse, BI, SERP/AI observations, agent clients
Connector layer	OAuth, service account, CSV/XLSX upload, SFTP, API polling, webhook intake, partner imports
Normalization layer	URL canonicalization, dedupe, redirect resolution, path parsing, query stripping, region/language handling
Ledger layer	URL asset, cluster, snapshot, metric observation, attribution event, rating, evidence, policy, action log
Scoring layer	13 structural variables, channel exposure, risk, value-at-risk, confidence, action priority
Governance layer	Roles, policies, approval thresholds, no-touch zones, agent permissions, audit trail
Experience layer	Dashboard, reports, exports, QBRs, APIs, agent ingress, partner workbench
Reconciliation layer	Pre/post measurement, outcome variance, proof pack, benchmark contribution, rating adjustment

Reference flow

Observe -> Normalize -> Resolve -> Score -> Decide -> Govern -> Act -> Reconcile -> Benchmark

4. Connector Catalog

Tier 1: MVP connectors

Connector	Data captured	Why it matters
Sitemap / URL inventory	Known URLs, sitemap priority, lastmod, section patterns	Establishes baseline asset universe.
Crawler export	Status codes, canonicals, redirects, titles, H1s, schema, crawl depth, internal links	Detects structural decay and crawlability risk.
Google Search Console	Page/query impressions, clicks, CTR, position, indexing signals where available	Measures organic discovery and visibility shifts.
GA4 or analytics export	Landing page sessions, events, conversions, engagement, source/medium	Measures behavior and conversion signals by URL.
CMS export	Publish dates, update dates, author, content type, tags, category, owner where available	Adds lifecycle, ownership, and maintenance context.
Manual revenue mapping	Lead value, order value, close rate assumptions, RPM, or pipeline proxy	Enables value-at-risk and recovery modeling.

Tier 2: Expansion connectors

Connector	Data captured	Expansion use case

CRM	Leads, contacts, opportunities, closed-won revenue, campaign influence	Pipeline attribution and revenue truth.
Payments / ecommerce	Orders, revenue, margin, product category, refunds	Direct revenue and margin-linked URL value.
Paid media	Ad spend, campaigns, landing pages, CAC, ROAS	Landing page efficiency and paid/organic interaction.
Lifecycle email	Email clicks, nurture path, campaign landing pages	Nurture influence and assisted conversion mapping.
Data warehouse	Modeled business truth, customer tables, product tables, events	Enterprise-grade joins and downstream reporting.
BI tools	Published dashboards, KPI definitions, data lineage	Exec alignment and reporting sync.
SERP / AI observations	AI citations, answer presence, competitor citations, zero-click risk	AI-mediated discovery and answer-share monitoring.

Connector priority matrix

Priority	Connector set	Target buyer value
P0	Sitemap + crawl + GSC + GA4	Fastest audit baseline and lowest access friction.
P1	CMS + CRM/payments	Ownership, lifecycle, and revenue truth.
P2	Ads + email + warehouse	Channel-agnostic attribution and capital allocation.
P3	SERP/AI + agent ingress	Future discovery surfaces and governed machine consumption.

5. Ingestion and Normalization Pipeline

1. Receive raw data from connector, upload, API call, or webhook.
2. Store original source payload with timestamp, account, permission scope, and hash.
3. Normalize URLs: protocol, hostname, casing, trailing slash, query parameters, fragments, canonicals, redirects, and locale.
4. Resolve URL asset identity: existing URL_ID, alias, redirect target, canonical equivalent, or new asset.
5. Map URL to cluster: section, template, content type, intent, business unit, product line, country, language, or custom grouping.
6. Create metric observations and evidence artifacts without overwriting prior truth.
7. Run scoring and confidence logic using the latest observation window.
8. Emit action candidates and route high-risk recommendations through policy.

Canonicalization rules

Rule area	Recommended default	Notes
Protocol	Normalize http to https when the live canonical resolves to https.	Keep source URL as alias evidence.
Host	Normalize www/non-www based on canonical domain setting.	Support multiple properties and subdomains.
Trailing slash	Normalize by actual 200/canonical behavior, not preference.	Preserve raw variants as aliases.
Query parameters	Strip known tracking parameters; preserve business-critical parameters.	Maintain parameter policy table.
Fragments	Strip fragments from canonical URL identity.	Can preserve anchor-level observations separately later.
Case	Lowercase host; path handling depends on platform behavior.	Avoid destructive assumptions on case-sensitive servers.
Redirects	Map source to final URL with redirect chain evidence.	Use final URL as active asset when stable.

6. Canonical API Surface

API design goal

Expose canonical URL asset truth without forcing external systems to understand every source-system id, connector nuance, or scoring rule.

Core API resources

Resource	Example path	Purpose
URL assets	/v1/url-assets	Create, search, retrieve, and update URL asset records.
Clusters	/v1/clusters	Group URL assets into portfolio views and reporting units.
Snapshots	/v1/url-assets/{id}/snapshots	Store time-stamped technical, content, and status observations.
Metrics	/v1/url-assets/{id}/metrics	Read or submit channel observations by window and source.
Ratings	/v1/url-assets/{id}/ratings	Retrieve current and historical scores.
Evidence	/v1/evidence	Attach screenshots, exports, crawl files, comments, or source notes.
Policies	/v1/policies	Define action permissions, thresholds, and no-touch zones.
Actions	/v1/actions	Request, approve, execute, and reconcile state transitions.
Benchmarks	/v1/benchmarks	Retrieve normalized peer or historical benchmark comparisons.
Agent sessions	/v1/agent-sessions	Create governed agent sessions and permission-scoped requests.

Example: URL asset response


```
{
  "url_id": "url_01HZX7A3K8P4",
  "canonical_url": "https://example.com/guides/widget-buying-guide",
  "asset_status": "active",
  "cluster_id": "cluster_buying_guides",
  "content_type": "guide",
  "intent_role": "commercial_evaluation",
  "owner": "growth_marketing",
  "rating_current": "B-",
  "risk_score": 62,
  "value_at_risk_estimate": 184000,
  "recommended_action": "refresh_and_relink",
  "policy_tier": "review_required",
  "last_observed_at": "2026-06-01T15:00:00Z"
}
```


Example: action request payload


```
{
  "url_id": "url_01HZX7A3K8P4",
  "action_type": "refresh",
  "requested_by": "agent:content_ops_assistant",
  "reason_code": "meaningful_decay",
```

```
  "expected_outcome": "recover impressions and assisted conversions",
  "evidence_ids": ["ev_2291", "ev_2292"],
  "risk_notes": "Commercial page. Requires content owner approval before CMS writeback."
}
```


7. Webhooks and Event Model

Event	When emitted	Typical subscriber
url_asset.created	New URL asset is created or discovered.	Warehouse, BI, partner workbench.
url_asset.changed	Canonical, status, owner, cluster, or metadata changes.	CMS, governance queue, audit log.
rating.changed	Rating materially changes or crosses threshold.	CSM, account owner, Slack/Teams, QBR workflow.
risk.threshold_crossed	Risk score exceeds configured trigger.	Technical SEO, RevOps, content owner.
action.requested	Human or agent requests a state transition.	Approval workflow.
action.approved	Action is approved for execution.	CMS, Jira, Asana, implementation partner.
action.completed	Execution evidence is attached.	Measurement workflow.
outcome.reconciled	Before/after results are calculated.	Dashboard, case study proof pack, benchmark store.
agent.session.started	Agent begins a permission-scoped interaction.	Security log, governance dashboard.

Event envelope


```
{
  "event_id": "evt_01J0",
  "event_type": "rating.changed",
  "occurred_at": "2026-06-01T18:21:00Z",
  "tenant_id": "tenant_1up_demo",
  "actor": "system:ratings_engine",
  "object_type": "url_asset",
  "object_id": "url_01HZX7A3K8P4",
  "previous_state": {"rating": "B"},
  "current_state": {"rating": "C+"},
  "evidence_ids": ["ev_3011"],
  "confidence": "medium"
}
```

8. Exports, BI, and Warehouse Sync

- CSV/XLSX export should exist before full API adoption, because many buyers will start with audit exports.
- Warehouse sync should preserve raw source observations, normalized URL assets, scoring outputs, policy status, and action history.
- BI outputs should use stable semantic definitions: active URLs, indexable URLs, strategic assets, protected assets, value-at-risk, and recoverable backlog.
- Exports should include data freshness, source, and confidence fields so downstream teams understand evidence quality.

Export object	Minimum fields	Best use
URL asset register	url_id, canonical_url, cluster, status, owner, content_type, dates, policy_tier	System of record and inventory reconciliation.
Ratings export	url_id, rating, 13 scores, risk score, confidence, reason codes	Portfolio review and QBR.
Channel observations	url_id, source, window, metric, value, confidence	Warehouse and BI analysis.
Action backlog	action_id, url_id, action_type, impact, effort, owner, status	Sprint planning and execution.
Evidence index	evidence_id, url_id, source, artifact_type, captured_at, hash	Audit pack and procurement trust.

9. Agent Ingress API

Positioning rule

The platform is not the agent. The platform is the governed truth layer that agents must query, request through, and reconcile back into.

Agent capability	Allowed operation	Default policy
Read ledger state	Query URL assets, scores, evidence, policies, and actions.	Allowed for scoped agents.
Recommend action	Create action_request with evidence and rationale.	Allowed with reason code and confidence.
Draft implementation notes	Generate specs for refresh, redirect, merge, internal link, or schema fixes.	Allowed for review queue.
Modify CMS content	Write content changes to CMS.	Blocked by default; requires explicit approval and connector scope.
Redirect or delete URLs	Change routing, redirects, canonical tags, or indexation directives.	High-risk; human approval required.
Reconcile outcomes	Compare expected vs actual results after measurement window.	Allowed when source metrics are available.

Agent session lifecycle

9. Agent authenticates with tenant-scoped credentials and declared purpose.
10. Ledger returns a capability manifest and policy limits.
11. Agent queries URL assets and evidence within its scope.
12. Agent submits recommendations or action requests.
13. Policy engine approves, blocks, or escalates.
14. Human or authorized system executes action.
15. Outcome is measured and reconciled into the ledger.

10. Authentication, Authorization, and Permissions

Control	Recommended MVP approach	Enterprise direction
Tenant isolation	Separate tenant_id on every object and query.	Dedicated tenant partitions or VPC options for enterprise.
User auth	Email/password or SSO-ready identity provider.	SAML/OIDC SSO and SCIM.
API auth	Scoped API keys for MVP.	OAuth client credentials and rotating secrets.
Role model	Admin, analyst, viewer, approver, partner, agent.	Custom RBAC and object-level permissions.
Connector auth	Read-only scopes by default.	Granular scope management and approval workflow.
Audit log	Log actor, action, object, timestamp, source, before/after state.	Immutable audit stream and exportable evidence pack.

Permission tiers

Tier	Description	Example
View	Can inspect data and reports.	Executive, finance, partner reviewer.
Analyze	Can run scoring, create findings, and assemble reports.	Analyst, implementation partner.
Recommend	Can submit action requests with evidence.	Agent, consultant, content strategist.
Approve	Can approve or reject actions.	Content owner, SEO lead, RevOps lead.
Execute	Can push or confirm changes in external systems.	Authorized admin or system connector.
Admin	Can manage policies, connectors, roles, and tenant settings.	Client admin or platform operator.

11. Data Quality and Confidence Framework

Why confidence matters

URL Ledger must not pretend all data has equal truth value. A direct CRM close event, a modeled lead value, and a manually uploaded estimate should all be usable, but they should not carry the same confidence.

Confidence tier	Definition	Typical sources
High	Directly observed, recent, complete, and joined to canonical URL identity.	GSC API, GA4 API, CRM closed-won, crawl evidence, CMS export.
Medium	Observed but incomplete, delayed, partially joined, or modeled.	Manual revenue mapping, campaign exports, partial CRM joins.
Low	Directional, estimated, stale, or assumption-heavy.	Stakeholder estimate, incomplete spreadsheet, legacy analytics.
Unknown	Field missing or unsupported.	No connector, no source, or ambiguous mapping.

Freshness rules

Data type	Freshness target	Stale threshold
Crawl / technical structure	Weekly for active accounts; daily for high-volume enterprise.	> 14 days.
GSC	Weekly for audit; daily/near-daily for platform.	> 10 days.

GA4	Weekly for audit; daily for platform.	> 7 days.
CRM/payments	Weekly or monthly depending sales cycle.	> 30 days.
CMS metadata	On change where possible; weekly otherwise.	> 30 days.
Ratings	Recalculate when inputs materially change.	No calculation after material change.

12. Implementation Sequence

Phase	Timeline	Outcome
Phase 0: scoping	Days 0-5	Confirm domain, data sources, URL volume, revenue mapping, permissions, and pilot goals.
Phase 1: baseline ingest	Days 5-15	Ingest sitemap/crawl/GSC/GA4; create URL register and cluster map.
Phase 2: scoring v1	Days 15-30	Run 13-variable scores, risk ratings, channel exposure, and priority backlog.
Phase 3: evidence pack	Days 30-45	Attach evidence, validate recommendations, prepare audit report, and select proof sprint.
Phase 4: platform install	Days 45-90	Enable recurring sync, action workflow, QBR export, and policy thresholds.
Phase 5: agent/warehouse expansion	Post-90	Expose API endpoints, agent ingress, benchmark contribution, and warehouse sync.

13. Testing, QA, and Acceptance Criteria

Test area	Acceptance criteria
URL normalization	Known duplicate, redirect, canonical, tracking, and trailing slash examples resolve correctly.
Connector ingest	Expected record counts match source exports within agreed tolerance.
Metric joins	GSC/GA4/CRM observations map to correct URL_ID or unresolved queue.
Scoring consistency	Same inputs produce same scores; reason codes are attached.
Evidence trail	Every major finding has source, timestamp, and confidence.
Policy workflow	High-risk actions route to approval and cannot be executed by default.
Exports	CSV/XLSX/API exports match dashboard totals and include freshness/confidence.
Security	Tenant isolation, role restrictions, audit log, and scoped credentials pass review.

Pilot acceptance checklist

- At least 95% of known indexable URLs are represented or explained in unresolved inventory.
- Top revenue or conversion URLs are mapped to URL_ID and cluster.
- At least one channel observation is attached to the majority of active URLs.
- Top 25 action backlog includes expected impact, effort, owner, evidence, and confidence.
- Protected assets and high-risk action rules are configured before any writeback workflow is enabled.
- Executive export and analyst export reconcile to the same underlying ledger records.

14. Future Roadmap

Roadmap item	Description	Strategic value
Native GSC/GA4 connectors	Direct OAuth sync and scheduled ingestion.	Reduces manual audit labor.
CMS connectors	WordPress, Webflow, Shopify, Contentful, HubSpot CMS.	Improves ownership, lifecycle, and action workflows.
CRM/payments joins	Salesforce, HubSpot, Zoho, Dynamics, Stripe, Shopify, WooCommerce.	Moves from traffic value to revenue value.
Warehouse sync	BigQuery, Snowflake, Redshift, Databricks.	Enterprise adoption and data-team trust.
Agent capability manifest	Machine-readable policy and action permissions.	Turns URL Ledger into the agent ingress layer.
Benchmark API	Retrieve anonymized peer norms and historical baselines.	Creates compounding benchmark moat.
Action writeback	Push approved tasks into CMS, Jira, Asana, Linear, or GitHub.	Completes observe-score-decide-act-reconcile loop.

Appendix A: Endpoint Summary

Method	Endpoint	Purpose
GET	/v1/url-assets	Search URL assets with filters.
POST	/v1/url-assets	Create URL asset manually or from import.
GET	/v1/url-assets/{id}	Retrieve canonical URL asset record.
GET	/v1/url-assets/{id}/ratings	Retrieve current and historical ratings.
POST	/v1/observations	Submit source-system metric or crawl observation.
POST	/v1/evidence	Attach evidence artifact or source note.
POST	/v1/actions	Request governed action.
PATCH	/v1/actions/{id}	Update action status or approval.
GET	/v1/policies	List policies and action thresholds.
POST	/v1/agent-sessions	Start scoped agent session.

Appendix B: Minimum Viable Import Template

Field	Required?	Description
canonical_url	Yes	Best known canonical URL or raw URL to normalize.
source_url	Optional	Raw URL from source system if different from canonical.
cluster_name	Optional	Manual or imported cluster/grouping.
content_type	Optional	Blog, guide, product, category, landing page, support, etc.
owner	Optional	Team or person accountable for the asset.
publish_date	Optional	Initial publication date.
last_modified_date	Optional	Latest known update date.
sessions	Optional	Landing page sessions for defined window.
conversions	Optional	Primary conversion count for defined window.
revenue_or_value	Optional	Direct revenue, pipeline value, or modeled value.
notes	Optional	Context, constraints, or protected asset notes.

Appendix C: Integration Readiness Scorecard

Readiness area	Green	Yellow	Red
URL inventory	Clean sitemap/crawl export available.	Partial inventory available.	No trusted URL list.
Analytics	GA4 or equivalent by landing page.	Analytics exists but incomplete.	No reliable analytics.
Search data	GSC access/export available.	Some keyword/rank exports only.	No search source.
Revenue mapping	CRM/payments or agreed value model.	Modeled conversion values only.	No monetization definition.
Ownership	Teams/owners known for key sections.	Some ownership known.	No owner map.
Governance	Approvers and protected pages known.	Some approval norms exist.	No change control.
Technical access	Crawl allowed and technical contact assigned.	Limited crawl or contact.	Blocked crawl/access.

Appendix D: Glossary

Term	Definition
URL asset	A canonical page or URL-level unit that can be measured, scored, governed, and reconciled.
Cluster	A portfolio grouping of URL assets by section, topic, intent, product, country, or business unit.
Observation	A time-stamped metric, crawl result, event, or source-system signal about a URL asset.
Evidence artifact	A file, screenshot, export, source payload, or explanation that supports a finding or score.
Policy gate	A rule layer that determines whether an action is allowed, escalated, or blocked.
Action request	A proposed state transition such as refresh, merge, redirect, protect, expand, or retire.
Outcome reconciliation	The process of comparing expected impact to observed results after an action.
Agent ingress	Permissioned access path for AI agents to query ledger truth and request governed actions.

Final note

Integration north star

URL Ledger wins when every external system can ask one question and trust the answer: What is this URL, what is it worth, what risk does it carry, what evidence supports that view, and what is allowed to happen next?