BridgeBrain SDK/API Developer Guide

Version 2.2.0 (Plugin: bridgebrain-sdk-api v2.2.0)

This guide documents the REST API exposed by the BridgeBrain SDK/API WordPress plugin v2.2.0. It is designed to match the plugin behavior exactly, including Pro (Extended Training) security changes introduced in v2.2.0.

What BridgeBrain is: the identity + rights layer for AI personas (PTP + PLF), plus an SDK/API that lets your apps securely load personas, verify entitlements, and report usage for royalties.
What BridgeBrain is not: an LLM provider. You can run inference on BridgeBrain hosted processing or bring your own model and still keep rights, licensing, and usage reporting consistent.

What's new in v2.2

v2.2.0 focuses on Pro Training (Extended Training) security and consistency: the API now returns a sanitized, normalized extended_training payload and never exposes server file paths. Files are provided via rights-checked download URLs.
It also standardizes the payload returned by both the dedicated /extended-training endpoint and the optional include=extended_training expansion on GET /personas/{id}.

Base URL

The API is served from your WordPress site where the BridgeBrain SDK/API plugin is installed:

```
https://YOUR_DOMAIN/wp-json/bb/v1
```

All endpoints below are relative to that base URL.

Authentication

Most endpoints require an API key. Send it as a request header:

```
X-BridgeBrain-User-Key: YOUR_48_CHARACTER_KEY
```

The plugin accepts multiple header variations to survive different proxies/hosts (e.g., HTTP_X_BRIDGEBRAIN_USER_KEY). For consistency, use X-BridgeBrain-User-Key in client code.

Public endpoints (no auth required):

- GET /marketplace/search
- GET /keys/jwks

Endpoint Summary

Method	Endpoint	Description	Auth
GET	/me	Return authenticated user summary (user_id, user_login, inventory_count, has_api_key).	Auth
GET	/inventory	List personas/items in inventory with license & Pro flags.	Auth
POST	/inventory	Import a PTP persona container into inventory (requires active entitlement).	Auth
GET	/inventory/{id}	Fetch a single inventory record by persona_id/key.	Auth
POST	/inventory/sync	Sync all personas owned by the authenticated user into inventory.	Auth
GET	/personas/{id}	Get persona metadata (version, licensing/governance JSON, Pro flags). Requires active license. Optional include=extended_training.	Auth
GET	/personas/{id}/constitution	Get persona constitution JSON + SHA-256 hash + identity_kernel_id.	Auth
GET	/personas/{id}/memory	Fetch memory beads for persona (requires BridgeBrain Core memory module).	Auth
POST	/personas/{id}/memory/search	Semantic search persona memory, optionally filtered by Passport (memory_key/bb_passport_key).	Auth
POST	/sessions/{id}/memory/write	Write a memory bead to a session, optionally Passport-tagged.	Auth
GET	/personas/{id}/extended-training	Get sanitized Pro/Extended Training payload (text, urls, files) if user has access.	Auth
GET	/personas/{id}/extended-training/files/{idx}	Download a specific extended training file (rights-checked).	Auth
POST	/licenses	Issue a license + JWT proof (default 30-day license, 24-hour proof).	Auth
POST	/licenses/{license_id}/verify?proof=JWT	Verify a license proof JWT (HS256).	Auth

Method	Endpoint	Description	Auth
POST	/usage	Record a usage event (tokens/minutes/requests/etc).	Auth
GET	/usage/reports	Get aggregated usage totals grouped by persona_id and type.	Auth
GET	/marketplace/search?q=...	Search public marketplace index (option keys bb_persona_*).	Public
POST	/keys/rotate	Rotate the authenticated user API key (48 chars).	Auth
GET	/keys/jwks	JWKS endpoint exposing HS256 oct key (public).	Public

Account

GET /me (Auth)
Return basic account details for the authenticated user.

Response (shape)

```
{
  "user_id": 123,
  "user_login": "robert",
  "inventory_count": 17,
  "has_api_key": true
}
```

Example

```
curl -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  https://YOUR_DOMAIN/wp-json/bb/v1/me
```

API Key Rotation

POST /keys/rotate (Auth)
Rotate the current user's API key. Use this if a key is compromised or you want to invalidate existing client configs.

Response (shape)

```
{ "ok": true, "key": "NEW_48_CHAR_KEY" }
```

Notes

Rotating a key will break any apps still using the previous key until they are updated.

Example

```
curl -X POST \
  -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  https://YOUR_DOMAIN/wp-json/bb/v1/keys/rotate
```

JWT Key Discovery (JWKS)

GET /keys/jwks (Public)
Return a JWKS document describing the HS256 signing key used for license proofs.

Response (shape)

```
{
  "keys": [
    { "kty": "oct", "alg": "HS256", "kid": "bb-default", "k": "BASE64_SECRET" }
  ]
}
```

Notes

This is primarily for systems that want to discover signing metadata. License proofs are HS256 and verified server-side via /licenses/{id}/verify.

Example

```
curl https://YOUR_DOMAIN/wp-json/bb/v1/keys/jwks
```

Inventory

GET /inventory (Auth)
List inventory items. The plugin enhances each item with license status and Pro capability flags.

Response (shape)

```
{
  "123": {
    "persona_id": 123,
    "name": "My Persona",
    "version": "1.2.0",
    "license_tier": "personal",
    "license_status": "active",
    "pro_enabled": true,
    "pro_capabilities": ["extended_training"]
  }
}
```

Example

```
curl -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  https://YOUR_DOMAIN/wp-json/bb/v1/inventory
```

Inventory Import (PTP Container)

POST /inventory (Auth)
Import a persona container into inventory. This is license-gated: the user must have an active entitlement for the persona_id.

Request body (JSON)

Field	Type	Required	Description
persona_id	string	Yes	Persona identifier in the PTP container.
name	string	Yes	Display name/title.
version	string	No	Semantic version (default 1.0.0).
provider	string	No	LLM provider identifier.
role	string	No	Persona role label.
purpose	string	No	Persona purpose statement.

Field	Type	Required	Description
licensing	object	No	PLF licensing metadata (stored as JSON if persona_id is numeric).
governance	object	No	Governance metadata (stored as JSON if persona_id is numeric).
pro_capabilities	object/array	No	Pro capability metadata (stored as JSON if persona_id is numeric).

Response (shape)

```
{ "ok": true, "persona_id": "123" }
```

Notes

If the user does not have an active entitlement for the persona, the API returns 403 (bb_license_required or bb_license_inactive).

Example

```
curl -X POST \
  -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "persona_id": "123",
    "name": "My Persona",
    "version": "1.0.0"
  }' \
  https://YOUR_DOMAIN/wp-json/bb/v1/inventory
```

Inventory Item

GET /inventory/{id} (Auth)
Fetch a specific inventory record by its persona_id/key.

Response (shape)

```
{ "persona_id": 123, "name": "My Persona", "version": "1.0.0", "...": "..." }
```

Example

```
curl -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  https://YOUR_DOMAIN/wp-json/bb/v1/inventory/123
```

Inventory Sync

POST /inventory/sync (Auth)
Sync all bb_persona posts owned by the authenticated user into their inventory store.

Response (shape)

```
{
  "success": true,
  "synced": 3,
  "skipped": 2,
  "total_inventory": 17,
  "message": "Synced 3 personas to inventory (2 already existed)"
}
```

Notes

This is a convenience endpoint for developers and creators to ensure owned personas appear in inventory without manual import.

Example

```
curl -X POST \
  -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  https://YOUR_DOMAIN/wp-json/bb/v1/inventory/sync
```

Persona Metadata

GET /personas/{id} (Auth)
Return persona metadata (requires a valid license for the requesting user).

Query parameters

Name	Type	Description
include	string	Optional comma-separated expansions. Use include=extended_training to embed sanitized extended_training payload when available.

Response (shape)

```
{
  "id": 123,
  "name": "My Persona",
  "version": "1.2.0",
  "provider": "anthropic",
  "role": "Coach",
  "purpose": "Help write better",
  "constitution_hash": "sha256...",
  "licensing": { ... },
  "governance": { ... },
  "pro_enabled": true,
  "pro_capabilities": ["extended_training"],
  "license_tier": "pro",
  "extended_training_available": true,
  "created_at": "2025-12-01 10:00:00",
  "modified_at": "2026-01-03 08:00:00"
}
```

Notes

If license is missing/invalid, returns 403 with a specific error code. For Pro/Extended Training, use the dedicated /extended-training endpoint or include=extended_training.

Example

```
curl -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  "https://YOUR_DOMAIN/wp-json/bb/v1/personas/123?include=extended_training"
```

Persona Constitution

GET /personas/{id}/constitution (Auth)
Return persona constitution JSON plus a SHA-256 hash for integrity verification.

Response (shape)

```
{
  "persona_id": 123,
  "persona_name": "My Persona",
  "version": "1.2.0",
  "constitution": { "principles": [], "behaviors": [], "constraints": [] },
  "constitution_hash": "sha256...",
  "identity_kernel_id": "kernel_..."
}
```

Example

```
curl -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  https://YOUR_DOMAIN/wp-json/bb/v1/personas/123/constitution
```

Persona Memory (read)

GET /personas/{id}/memory (Auth)
Fetch memory beads for a persona. Requires the BridgeBrain Core memory module (BB_Memory).

Query parameters

Name	Type	Description
limit	int	Max beads to return (default 50).
marker_type	string	Filter by marker type (e.g., evolution, boundary, goal).
context_type	string	Filter by context type (e.g., sdk, chat, system).

Response (shape)

```
{
  "persona_id": 123,
  "count": 2,
  "beads": [ /* bead objects from BridgeBrain Core */ ]
}
```

Notes

If the BridgeBrain Core memory module is not active, returns 503 (bb_memory_unavailable).

Example

```
curl -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  "https://YOUR_DOMAIN/wp-json/bb/v1/personas/123/memory?limit=20&context_type=sdk"
```

Persona Memory (semantic search)

POST /personas/{id}/memory/search (Auth)
Semantic search through a persona's memory. Optionally filter results per-user using Passport keys.

Request body (JSON)

Field	Type	Required	Description
query	string	Yes	Search text.
limit	int	No	Max results (default 5).
threshold	float	No	Similarity threshold (default 0.65).
memory_key	string	No	Passport key. If present, results are filtered to matching subject tag.
bb_passport_key	string	No	Alias of memory_key.

Response (shape)

```
{
  "persona_id": 123,
  "query": "relativity",
  "passport_filtered": true,
  "count": 1,
  "results": [ /* bead/search result objects */ ]
}
```

Notes

Passport filtering works by hashing the key and filtering results to beads tagged with subject:.

Example

```
curl -X POST \
  -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "query": "relativity",
    "memory_key": "passport_abc123"
  }' \
  https://YOUR_DOMAIN/wp-json/bb/v1/personas/123/memory/search
```

Session Memory (write)

POST /sessions/{id}/memory/write (Auth)
Write a new memory bead to a session. Supports Passport tagging for cross-app continuity.

Request body (JSON)

Field	Type	Required	Description
content	string	Yes	Bead content.
speaker	string	Yes	Who said it (e.g., user, persona, system).
persona_id	int	No	Persona numeric ID to associate.
context_type	string	No	Context (default sdk).
tags	array	No	Array of tags.
marker_type	string	No	Marker type.
source_app	string	No	App identifier (default sdk).
memory_key	string	No	Passport key to tag bead with subject hash.
bb_passport_key	string	No	Alias of memory_key.

Response (shape)

```
{ "ok": true, "bead_id": 999, "session_id": 456, "passport_tagged": true }
```

Example

```
curl -X POST \
  -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "content": "User asked about relativity",
    "speaker": "user",
    "persona_id": 123,
    "source_app": "my-app",
    "memory_key": "passport_abc123"
  }' \
  https://YOUR_DOMAIN/wp-json/bb/v1/sessions/456/memory/write
```

Extended Training (Pro Training) payload

GET /personas/{id}/extended-training (Auth)
Return sanitized extended training data for a persona if the requester has rights (owner or licensed subscriber with explicit entitlement/pro flag).

Response (shape)

```
{
  "text": "<p>Sanitized training text...</p>",
  "urls": [
    { "url": "https://example.com", "label": "Optional label", "added_at": "2026-01-03T..." }
  ],
  "files": [
    {
      "index": 0,
      "filename": "file.pdf",
      "type": "application/pdf",
      "size": 12345,
      "sha256": "abc123...",
      "download_url": "https://YOUR_DOMAIN/wp-json/bb/v1/personas/123/extended-training/files/0"
    }
  ],
  "version": "1.1",
  "updated_at": "2026-01-03T..."
}
```

Notes

v2.2.0 ensures no server file paths are ever returned. Use download_url for file access.

Example

```
curl -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  https://YOUR_DOMAIN/wp-json/bb/v1/personas/123/extended-training
```

Extended Training File Download

GET /personas/{id}/extended-training/files/{idx} (Auth)
Download an extended training file by index. The plugin enforces the same access checks as the payload endpoint.

Notes

This endpoint streams the file to the client and sets appropriate content-type and filename headers.

Example

```
curl -L \
  -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  https://YOUR_DOMAIN/wp-json/bb/v1/personas/123/extended-training/files/0 \
  -o training-file.pdf
```

Issue License

POST /licenses (Auth)
Issue a license and return a short-lived JWT proof used for verification. Also upserts the user's entitlement record for the persona.

Request body (JSON)

Field	Type	Required	Description
persona_id	string	Yes	Persona identifier.
tier	string	No	License tier (default personal).

Field	Type	Required	Description
entitlements	array/object	No	Entitlements to store on the license (can include extended_training_access).
pro_enabled	bool	No	Whether Pro is enabled for this license (default false).

Response (shape)

```
{
  "license": {
    "license_id": "LIC-XXXXXXXXXX",
    "persona_id": "123",
    "holder": 45,
    "tier": "pro",
    "entitlements": { "extended_training_access": true },
    "pro_enabled": true,
    "expires": "2026-02-02T00:00:00Z"
  },
  "proof": "HEADER.PAYLOAD.SIGNATURE"
}
```

Notes

Proof JWTs include an exp claim (~24 hours). The stored license default expiration is ~30 days. Exact entitlement structure is application-defined; use extended_training_access=true to explicitly grant Pro Training access.

Example

```
curl -X POST \
  -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "persona_id": "123",
    "tier": "pro",
    "pro_enabled": true,
    "entitlements": { "extended_training_access": true }
  }' \
  https://YOUR_DOMAIN/wp-json/bb/v1/licenses
```

Verify License Proof

POST /licenses/{license_id}/verify?proof=JWT (Auth)
Verify a JWT proof signature and expiration.

Response (shape)

```
{
  "valid": true,
  "payload": {
    "license_id": "LIC-...",
    "persona_id": "123",
    "holder": 45,
    "tier": "pro",
    "exp": 1700000000
  }
}
```

Example

```
curl -X POST \
  -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  "https://YOUR_DOMAIN/wp-json/bb/v1/licenses/LIC-XXXXXXXXXX/verify?proof=HEADER.PAYLOAD.SIGNATURE"
```

Record Usage

POST /usage (Auth)
Record a usage event for reporting and royalty calculation.

Request body (JSON)

Field	Type	Required	Description
app_id	string	No	Your app identifier.
persona_id	string	No	Persona identifier.
type	string	No	Metric type (default tokens).
value	int	No	Metric value (default 0).
meta	object	No	Arbitrary metadata object.

Response (shape)

```
{ "ok": true }
```

Example

```
curl -X POST \
  -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "app_id": "my-app",
    "persona_id": "123",
    "type": "tokens",
    "value": 812
  }' \
  https://YOUR_DOMAIN/wp-json/bb/v1/usage
```

Usage Reports

GET /usage/reports (Auth)
Return aggregated usage totals grouped by (persona_id, type).

Response (shape)

```
[
  { "persona_id": "123", "type": "tokens", "total": 123456 },
  { "persona_id": "123", "type": "minutes", "total": 98 }
]
```

Example

```
curl -H "X-BridgeBrain-User-Key: YOUR_KEY" \
  https://YOUR_DOMAIN/wp-json/bb/v1/usage/reports
```

Marketplace Search

GET /marketplace/search?q=... (Public)
Search the public persona marketplace index.

Response (shape)

```
[
  { "persona_id": "123", "display_name": "Legendary Producer" },
  { "persona_id": "456", "display_name": "Physics Tutor" }
]
```

Notes

Marketplace results are populated from WordPress options whose keys start with bb_persona_.

Example

```
curl "https://YOUR_DOMAIN/wp-json/bb/v1/marketplace/search?q=producer"
```

Error Codes & Statuses

Common errors returned by the plugin:

HTTP	Error Code	Description
401	bb_auth	Missing/invalid API key.
403	bb_license_required / bb_license_inactive /	User lacks required entitlement/license or license invalid.
403	bb_forbidden	Access denied (e.g., Extended Training without rights).
404	bb_not_found	Resource not found.
400	bb_invalid_input / bb_invalid_query / bb_invalid_proof	Invalid request body, query, or JWT proof.
503	bb_memory_unavailable	BridgeBrain Core memory module not active.

Versioning & Compatibility

Persona version strings follow semantic versioning (MAJOR.MINOR.PATCH). Pro/Extended Training payloads include their own version field (training_data.version) to support format evolution without breaking clients.

Changelog Notes (v2.1 -> v2.2)

v2.1 introduced Extended Training endpoints. v2.2 hardened them by sanitizing payloads and replacing any filesystem paths with rights-checked download URLs.

Appendix: Quick cURL Header Helper

```
# bash
BB_BASE="https://YOUR_DOMAIN/wp-json/bb/v1"
BB_KEY="YOUR_48_CHARACTER_KEY"

curl -H "X-BridgeBrain-User-Key: $BB_KEY" "$BB_BASE/me"
```