

BridgeBrain SDK/API Developer Guide

Version 2.3.0 (Plugin: bridgebrain-sdk-api v2.3.0)

Generated February 12, 2026

This guide documents the REST API exposed by the BridgeBrain SDK/API WordPress plugin v2.3.0. It is designed to match the plugin behavior exactly, including Pro Training portability introduced in v2.3.0 and 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 v2.3

v2.3.0 adds portable, permissioned Pro Training support to the SDK/API. Applications can now fetch a persona's Pro Training blob through a dedicated endpoint, with opt-in inclusion via `include=pro_training` for one-call convenience. A SHA-256 hash is returned for efficient client-side caching and sync.

This release is additive and designed to work alongside Extended Training without changing existing endpoint behavior.

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
```

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 (for example `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	/inventory/{id}/facets	Export live Persona Action Panel facets for an inventory item (if facets table exists).	Auth
GET	/personas/{id}	Get persona metadata (version, licensing/governance JSON, Pro flags). Requires active license. Optional include=extended_training, include=pro_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
GET	/personas/{id}/pro-training	Get Pro Training blob payload (enabled, blob, sha256, updated_at) if user has access.	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
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", "pro_training"]
  }
}
```

Note: `pro_capabilities` is an array sourced from persona meta. Capability names are application-defined; examples include `extended_training` and `pro_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
<code>persona_id</code>	string	Yes	Persona identifier in the PTP container.
<code>name</code>	string	Yes	Display name/title.
<code>version</code>	string	No	Semantic version (default 1.0.0).
<code>provider</code>	string	No	LLM provider identifier.
<code>role</code>	string	No	Persona role label.
<code>purpose</code>	string	No	Persona purpose statement.
<code>licensing</code>	object	No	PLF licensing metadata (stored as JSON if <code>persona_id</code> is numeric).
<code>governance</code>	object	No	Governance metadata (stored as JSON if <code>persona_id</code> is numeric).
<code>pro_capabilities</code>	object/array	No	Pro capability metadata (stored as JSON if <code>persona_id</code> is numeric).
<code>continuity</code>	object	No	PTP continuity metadata (new in PTP v2.3). Stored as packed JSON and mirrored into individual meta keys for UI/export consistency.

Response (shape)

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

Notes: if the user does not have an active entitlement for the persona, the API returns 403 with `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`.

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)"  
}
```

Example

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

Inventory Facets Export

GET /inventory/{id}/facets (Auth)

Return live Persona Action Panel facets for an inventory item. If the `bb_pap_facets` table is not installed, an empty facets object is returned.

Example

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

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. Supported values: extended_training, pro_training. Example: include=extended_training,pro_training

Example

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

Persona Constitution

GET /personas/{id}/constitution (Auth)

Example

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

Persona Memory

GET /personas/{id}/memory (Auth)

Query parameters

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

Example

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

POST /personas/{id}/memory/search (Auth)

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.

POST /sessions/{id}/memory/write (Auth)

Request body (JSON)

Field	Type	Required	Description
content	string	Yes	Bead content.
speaker	string	Yes	Who said it (for example 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.

Extended Training

GET /personas/{id}/extended-training (Auth)

Response (shape)

```
{
  "text": "<p>Sanitized training text...</p>",
  "urls": [
    { "url": "https://example.com", "label": "Optional label", "added_at": "2026-02-12T..." }
  ],
  "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": "2026-02-12T..."
}
```

GET /personas/{id}/extended-training/files/{idx} (Auth)

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
```

Pro Training

GET /personas/{id}/pro-training (Auth)

Response (shape)

```
{
  "enabled": true,
  "blob": "{...large training blob...}",
  "sha256": "abc123...",
  "updated_at": "2026-02-12T..."
}
```

Licenses

POST /licenses (Auth)

Request body (JSON)

Field	Type	Required	Description
persona_id	string	Yes	Persona identifier.
tier	string	No	License tier (default personal).
entitlements	array/object	No	Entitlements to store on the license. For Pro access, include <code>extended_training_access</code> and/or <code>pro_training_access</code> as needed.
pro_enabled	bool	No	Whether Pro is enabled for this license (default false).

Usage

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.

Marketplace

GET /marketplace/search?q=... (Public)

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

Error Codes and Statuses

HTTP	Error Code	Where you will see it	Meaning
401	bb_auth	Any authenticated endpoint	Missing or invalid API key.
403	bb_license_required	POST /inventory	User lacks required entitlement/license.
403	bb_license_inactive	POST /inventory	Entitlement exists but is not active.
403	bb_forbidden	Training endpoints	Access denied (Extended Training / Pro Training without rights).
404	bb_not_found	Any endpoint	Resource not found (persona, inventory item, file index).
400	bb_invalid_proof	POST /licenses/{id}/verify	Invalid or expired JWT proof.
400	bb_invalid_input / bb_invalid_query / bb_ptp_invalid	Various	Invalid request body, query, or PTP container.
503	bb_memory_unavailable	Memory endpoints	BridgeBrain Core memory module not active.
403	LICENSE_REQUIRED	GET /personas/{id}	No license found for this persona.
403	LICENSE_EXPIRED	GET /personas/{id}	Persona license has expired.
403	LICENSE_PAUSED	GET /personas/{id}	License paused by owner.
403	LICENSE_CANCELLED	GET /personas/{id}	License has been cancelled.

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"
```