PLM integration guide

Send style updates from your product lifecycle management system, receive generated visuals and send a review decision back to MYTH.

Send a style

Use a partner API key for your company. The visual workflow requires PLM access on its plan. Send a batch to `POST /v1/events/plm`. Give each event its own EventId and use EntityId to identify the style.

```
{ "Events": [{ "EventId": "example-101", "EntityId": "STYLE-101",
               "EventType": "Style_Update" }] }
```

This example shows the event identity. Include product photos or technical drawings in the event attachments for generation. Read Status, ProcessedEvents and FailedEvents even on HTTP 200. Follow StyleUrls when supplied. Generation continues asynchronously.

Choose outputs and settings

A photo can produce on-model views. A sketch can produce a product render and on-model views from that render. Optional technical flats can include front and back drawings, PNG, DXF and SVG files, and a tech sheet in PDF or SVG. Results may be grouped by colourway.

`GET /v1/plm/config` returns effective settings and your stored overrides. Set automatic generation, daily limits, allowed regenerations, views, shoot and render options, image roles, attribute mapping and flat-drawing options. `PUT /v1/plm/config` replaces the saved overrides for future runs. Include each one you intend to keep.

Receive and recover results

Register an HTTPS callback at `POST /v1/webhooks`. List subscriptions with `GET /v1/webhooks`, send a ping with `POST /v1/webhooks/{id}/test` or deactivate one with `DELETE /v1/webhooks/{id}`. Verify the signature against the original body, check the timestamp and deduplicate deliveries using event_id.

```
plm.visual.ready      plm.visual.failed
plm.style.completed   plm.style.skipped
```

Read the run and current visuals at `GET /v1/plm/styles/{entity_id}`. Read a single result at `GET /v1/plm/visuals/{visual_id}` for status, inputs and output files. Download ready files within one hour or read the visual again for fresh links. To recover missed changes, call `GET /v1/plm/visuals?since=…` and follow next_cursor with the same filters until it is null.

Review a ready visual

Send a decision to `POST /v1/plm/visuals/{visual_id}/review`. Approval is final. By default, rejection requests a replacement and the note guides the next attempt. Regeneration is charged like a new visual and is subject to the configured limit. To record a rejection alone, send:

```
{"decision":"rejected","note":"Adjust colour","regenerate":false}
```