# AtlasForge Financial — Integration Guide

**Last updated: June 2026 · v2.1**

This guide is for AtlasForge partners (banks, credit unions, neobanks) integrating the Safe-to-Spend API. The same backend powers the marketing site at atlasforgefinancial.com and the partner portal at /portal.

Most teams ship a working integration in **under an hour**.

---

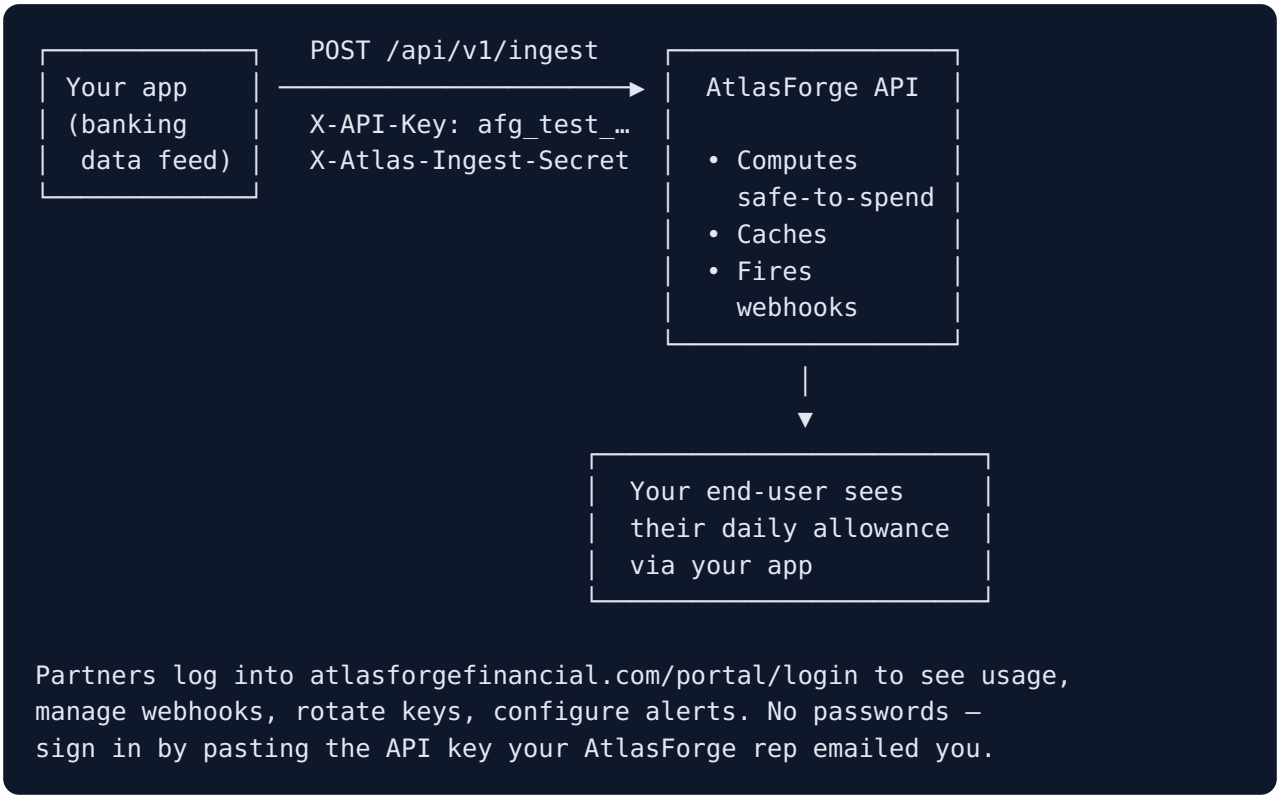
## 0. Base URLs

| What | URL |
|---|---|
| **API endpoint (all `/api/v1/*` calls)** | `https://atlasforgefinancial.com` |
| Partner portal (browser UI) | `https://atlasforgefinancial.com/portal/login` |
| Marketing site / status / docs | `https://atlasforgefinancial.com` |

> **Point all SDKs, curl commands, and server-to-server calls at `https://atlasforgefinancial.com`.** The apex `atlasforgefinancial.com` works too (same backend) but the `api.` subdomain is the canonical endpoint and what every example in this guide uses.

---

## 1. The big picture




## 2. Authentication

Every `/api/v1/*` call carries:

```
X-API-Key: afg_test_…   # sandbox
X-API-Key: afg_live_…   # production
```

The `afg_test_` prefix routes you to the sandbox dataset (safe to experiment, never billed). `afg_live_` routes to production (real usage counts toward your bill).

Ingest + webhook endpoints additionally require:

```
X-Atlas-Ingest-Secret: <fetched from GET /api/v1/secrets>
```

This is a **per-partner** secret. Cache it in your backend's memory or secret manager. Never expose it to a browser.

## 3. Onboarding — 5 minutes

1. Your AtlasForge rep emails you an `afg_test_*` key (sandbox) and an `afg_live_*` key (production).
2. Sign in at /portal/login with the test key.
3. On the **Keys** page, click **Reveal Ingest Secret** to retrieve your `atlas_ingest_*` secret. Copy it into your backend env.
4. Set your webhook URL on the **Webhooks** page (optional but recommended).
5. Start firing events.

## 4. The core endpoint — `POST /api/v1/ingest`

For each end-user whose safe-to-spend you want computed:

```
curl -X POST https://atlasforgefinancial.com/api/v1/ingest \
  -H "X-API-Key: $ATLAS_KEY" \
  -H "X-Atlas-Ingest-Secret: $ATLAS_INGEST_SECRET" \
  -H "Content-Type: application/json" \
  -d '{
    "end_user_id":         "usr_alice_001",
    "user_email":          "alice@yourbank.example.com",
    "cash_balance":        4218.50,
    "cash_buffer":         200,
    "upcoming_bills":      [
      { "name": "Rent",     "amount": 1450, "due_date": "2026-07-01" },
      { "name": "Internet", "amount": 79,   "due_date": "2026-06-25" }
    ],
    "savings_commitments": 300,
    "debt_commitments":    180
  }'
```

Response (HTTP 201):

```
{
  "safe_to_spend_today": 76.95,
  "daily_allowance":     76.95,
  "monthly_safe_to_spend": 2308.50,
  "breakdown": {
    "cash_after_buffer":     4018.50,
    "upcoming_bills_total":  1529.00,
    "savings_commitments":   300.00,
    "debt_commitments":      180.00,
    "discretionary_pool":    2009.50,
    "days_in_period":        30
  },
  "calculation_id": "8f2a…",
  "calculated_at":  "2026-06-18T16:42:00Z",
  "partner_id":     "878a4f1e-…",
  "is_sandbox":     true
}
```

Render `safe_to_spend_today` in your app — that's the number your user wants to see.

---

## 5. Reading back cached calculations

```
# Latest 100 (newest first)
curl "https://atlasforgefinancial.com/api/v1/ingest/calculations?limit=100" \
  -H "X-API-Key: $ATLAS_KEY" \
  -H "X-Atlas-Ingest-Secret: $ATLAS_INGEST_SECRET"
```

Paginate via the `next_cursor` field. To fetch a single user:

```
curl "https://atlasforgefinancial.com/api/v1/ingest/calculation/usr_alice_001" \
  -H "X-API-Key: $ATLAS_KEY" \
  -H "X-Atlas-Ingest-Secret: $ATLAS_INGEST_SECRET"
```

GDPR right-to-erasure — single call:

```
curl -X DELETE \
  "https://atlasforgefinancial.com/api/v1/ingest/user/usr_alice_001" \
  -H "X-API-Key: $ATLAS_KEY" \
  -H "X-Atlas-Ingest-Secret: $ATLAS_INGEST_SECRET"
```

## 6. Webhooks (optional)

Configure a webhook to receive push notifications when calculations update or usage thresholds are reached. Set the URL via the portal **Webhooks** page or:

```
curl -X POST https://atlasforgefinancial.com/api/v1/webhooks \
  -H "X-API-Key: $ATLAS_KEY" \
  -H "X-Atlas-Ingest-Secret: $ATLAS_INGEST_SECRET" \
  -H "Content-Type: application/json" \
  -d '{
    "url":     "https://your-app.example.com/hooks/atlas",
    "events":  ["calculation.updated"],
    "enabled": true
  }'
```

The response includes a **one-time** signing secret (`whsec_…`). Store it — we can't show it again. Verify incoming requests by computing `HMAC-SHA256(payload, whsec)` and comparing against the `X-Atlas-Signature` header.

Recent deliveries are visible at `GET /api/v1/webhooks/deliveries` (and on the portal's Webhooks page with filter pills).

## 7. Usage & billing

```
curl https://atlasforgefinancial.com/api/v1/usage \
  -H "X-API-Key: $ATLAS_KEY"
```

```
{
  "month": "2026-06",
  "partner": "878a4f1e-…",
  "production": { "calls": 12409, "unique_end_users": 832, "estimated_bill_usd": 998.40 },
  "sandbox":    { "calls": 1258,  "unique_end_users": 10,  "estimated_bill_usd": null },
  "rate_per_user_usd": 1.20
}
```

Set up alerts at `PUT /api/v1/usage/alerts` so you get an email before crossing a threshold.

## 8. Rotating an API key

```
curl -X POST https://atlasforgefinancial.com/api/v1/key/rotate \
  -H "X-API-Key: $ATLAS_KEY" \
  -H "Content-Type: application/json" \
  -d '{ "key_type": "production" }'
```

The response includes the new plaintext key — store it immediately. The old key stops working on the very next request.

## 9. Client helpers

### Node.js

```
const ATLAS  = process.env.ATLAS_BASE  || "https://atlasforgefinancial.com";
const KEY    = process.env.ATLAS_KEY;
const SECRET = process.env.ATLAS_INGEST_SECRET;

export async function atlasIngest(payload) {
  const r = await fetch(`${ATLAS}/api/v1/ingest`, {
    method: "POST",
    headers: {
      "X-API-Key": KEY,
      "X-Atlas-Ingest-Secret": SECRET,
      "Content-Type": "application/json",
    },
    body: JSON.stringify(payload),
  });
  if (!r.ok) throw new Error(`Atlas ingest failed: ${r.status}`);
  return r.json();
}
```

### Python

```
import os, requests
ATLAS  = os.environ.get("ATLAS_BASE", "https://atlasforgefinancial.com")
KEY    = os.environ["ATLAS_KEY"]
SECRET = os.environ["ATLAS_INGEST_SECRET"]

def atlas_ingest(payload: dict) -> dict:
    r = requests.post(
        f"{ATLAS}/api/v1/ingest",
        headers={"X-API-Key": KEY, "X-Atlas-Ingest-Secret": SECRET},
        json=payload, timeout=10,
    )
    r.raise_for_status()
    return r.json()
```

---

## 10. Endpoint reference

| Method | Path | Auth | Notes |
|---|---|---|---|
| `GET` | `/api/v1/me` | X-API-Key | Partner profile + key metadata |
| `GET` | `/api/v1/usage` | X-API-Key | Current-month calls + unique users + bill |
| `GET` | `/api/v1/secrets` | X-API-Key | Reveal your ingest secret |
| `POST` | `/api/v1/key/rotate` | X-API-Key | Body: `{ key_type: sandbox \| production }` |
| `POST` | `/api/v1/ingest` | X-API-Key + Ingest | Compute + cache safe-to-spend |
| `GET` | `/api/v1/ingest/calculations` | X-API-Key + Ingest | Paginated cache ( `limit` , `cursor` ) |
| `GET` | `/api/v1/ingest/calculation/{user_id}` | X-API-Key + Ingest | Single end-user |
| `DELETE` | `/api/v1/ingest/user/{user_id}` | X-API-Key + Ingest | GDPR delete |
| `GET` | `/api/v1/webhooks` | X-API-Key + Ingest | Read config |

| Method | Path | Auth | Notes |
|---|---|---|---|
| `POST` | `/api/v1/webhooks` | X-API-Key + Ingest | Set config; one-time secret in response |
| `DELETE` | `/api/v1/webhooks` | X-API-Key + Ingest | Remove |
| `POST` | `/api/v1/webhooks/rotate-secret` | X-API-Key + Ingest | New secret in response |
| `GET` | `/api/v1/webhooks/deliveries` | X-API-Key + Ingest | `limit`, `status=delivered\|failed` |
| `GET` | `/api/v1/usage/alerts` | X-API-Key | Threshold config + current values |
| `PUT` | `/api/v1/usage/alerts` | X-API-Key | Save thresholds + notify email |
| `DELETE` | `/api/v1/usage/alerts` | X-API-Key | Clear all |

---

## 11. Get help

- **Portal**: atlasforgefinancial.com/portal/login
- **Status**: atlasforgefinancial.com/status
- **Email**: customersupport@atlasforgefinancial.com
- **Phone**: (843) 291-2556

This guide is served live from the AtlasForge backend — every update auto-propagates to the website at `/developers` and to anyone who's bookmarked the PDF or markdown.