

# API Testing Tool (Postman Alternative)

Opportunity Proposal • Feb 1, 2026 • Cole

## 1. The Problem

---

Developers are frustrated with Postman's aggressive shift to cloud-first, subscription-based model with restrictive free tier limits. Solo developers and small teams hit paywalls quickly (25 collection runs/month, 3 users max, 1K API calls), face slow performance with large collections, and are forced into vendor lock-in with cloud sync. The problem affects millions of developers who need API testing but don't want enterprise pricing or forced cloud dependency.

## 2. Evidence of Demand

---

- **Direct user complaints:** "Tired of Postman's workspace limits, rising prices, slow performance" — multiple sources, Jan 2026
- **HN validation (TODAY):** Show HN: Voiden — "Every other API-testing tool has evolved into bloated, cloud-first, subscription-based service" (18 pts, 4h ago)
- **Competitor traction:** Bruno (open-source, Git-native) rapidly gaining adoption; Thunder Client has 3M+ VS Code downloads
- **Market validation:** 5+ "Postman alternatives" articles updated in Jan 2026 alone, showing active dissatisfaction
- **Anti-vendor-lock-in sentiment:** "My concern was to be trapped with \$160m VC-funded company when the inevitable rug gets pulled" (re: similar tools)
- **Pricing pain:** Postman team plans: \$12-49/user/month. Free tier strictly limited.

## 3. The Solution

---

Build a lightweight, offline-first, Git-native API testing tool that gives developers unlimited local usage without paywalls. Focus on speed, simplicity, and no vendor lock-in. Collections stored as plain files (JSON/YAML), sync via Git, zero cloud dependency by default. Optional paid tier adds team collaboration features (real-time sync, shared workspaces) at fair pricing.

## 4. MVP Scope

---

- HTTP request builder (GET, POST, PUT, DELETE, PATCH, etc.)
- Environment variables and secrets management
- Request collections stored as JSON/YAML files
- Basic JavaScript pre/post-request scripting
- Response viewer (JSON, XML, HTML, headers)
- Collection runner for automated testing
- VS Code extension OR lightweight Electron desktop app
- Git-native: collections are files, use normal Git workflows

**What's NOT in MVP:** GraphQL/WebSocket support, advanced testing frameworks, cloud sync, team features, import from Postman (v2 feature).

## 5. Build Estimate

---

**3-5 days** for functional MVP (VS Code extension path, faster than Electron).

- Day 1: HTTP client core, request builder UI
- Day 2: Collections (YAML storage), environment variables
- Day 3: Response viewer, basic scripting (eval JS snippets)
- Day 4: Collection runner, polish UI/UX
- Day 5: Testing, docs, publish to VS Code marketplace

*Tech stack: TypeScript, VS Code Extension API, axios/fetch for HTTP, YAML for storage.*

## 6. Monetization

---

**Freemium model:**

- **Free tier:** Unlimited local usage, unlimited collections, Git-native sync (100% free forever)
- **Pro tier (\$8/user/month or \$60/year):** Real-time team sync (WebSocket-based), shared workspaces, encrypted secret sharing, priority support

**Alternative:** One-time license (\$29-49) for lifetime access to Pro features (reduces ongoing billing complexity).

## 7. Path to \$400/month

---

### Subscription Path:

50 Pro users @ \$8/month = \$400

*Achievable? YES — Bruno has 20K+ GitHub stars in <2 years. If we convert 0.25% of interested users → 50 paying.*

### One-Time License Path:

14 licenses/month @ \$29 = \$406

*Lower friction, easier to hit early. Scale requires steady new user flow.*

**Distribution:** VS Code marketplace (built-in discovery), Product Hunt launch, dev.to/HN posts, GitHub README targeting "Postman alternative" searches.

## 8. Risks / Unknowns

---

- **Crowded space:** Bruno, Insomnia, Thunder Client, Testfully already exist. Differentiation: We're focused on Git-native + simplicity, avoid feature bloat.
- **Free tier cannibalization:** If free tier is too good, users won't upgrade. Mitigation: Team sync is clear value-add for small teams.
- **Build complexity:** HTTP client + scripting can get complex. Mitigation: Start minimal, use proven libraries (axios), limit scripting scope in MVP.
- **Discovery:** Getting initial users is always hard. Mitigation: Launch on Product Hunt, post to r/webdev, HN Show HN, target "frustrated Postman users" explicitly.
- **Postman could improve:** If Postman drops prices or improves free tier, market shrinks. Counter: Anti-vendor-lock-in sentiment is strong and growing.

## 9. Competitive Landscape

---

**Bruno:** Open-source, offline-first, Git-native. 20K+ stars. Free forever. No paid tier (yet). Opportunity: We can monetize better.

**Thunder Client:** VS Code extension, 3M+ downloads. Lightweight but lacks Git-native workflows. \$5-20/user pricing.

**Insomnia:** Kong-owned, shifted to cloud-first. Users migrating away (same complaints as Postman).

**Testfully:** SOC2/HIPAA compliant, targets enterprises. Different market segment.

## 10. Recommendation

---

### **STRONG OPPORTUNITY — BUILD IT.**

Clear pain, validated demand, proven market (competitors thriving), fast MVP build (3-5 days), viable path to \$400/month. Risk is manageable. This is exactly the kind of opportunity we're looking for: developers with budget, clear frustration, willingness to pay for solutions.