# Comparison of IDEs For Exploring and Testing APIs

## Postman vs. Insomnia vs. Bruno

Patrik Valentiny

---

## Project Overview

**Goal:** Evaluate and compare Integrated Development Environments (IDEs) for API testing, focusing on performance, data privacy, and automation.

**The Contenders:**

- **Postman:** The Industry Standard.
- **Insomnia:** The Extensible Alternative.
- **Bruno:** The Local-First Option.

---

## Methodology (Quality Metrics)

The comparison evaluates the tools across five key dimensions:

1. **Performance:** Startup time, request execution speed, and resource usage.
2. **Maintainability:** File formats, Git version control integration, and organization.
3. **Security:** Local vs. Cloud storage of credentials and environment variables.
4. **Testing Capabilities:** Assumption languages, scripting, and reporting.
5. **CI/CD Integration:** CLI availability and pipeline ease-of-use.

---

## Architecture: The Local-First Approach (Bruno)

Unlike Postman and Insomnia's proprietary formats, **Bruno** uses a custom plain-text markup language (`.bru`).

- **Benefit:** Enables "Infrastructure-as-Code" for API collections.
- **Storage:** Direct Git integration without export steps.

**Example `.bru` File:**

```
meta {
  name: Refresh Token
  type: http
  seq: 2
}
post {
  url: {{base_url}}/auth/refresh
  body: json
  auth: inherit
}
```

---

## Testing Capabilities

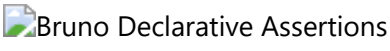
All three tools support the **Chai Assertion Library** allowing for standardized test scripts.

**Test Script Example (Postman/JavaScript):**

```
// Standard Chai Syntax
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});

pm.test("Response has correct product title", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.title).to.eql("iPhone 9");
});
```

*Bruno also supports declarative assertions (no-code).*


Bruno Declarative Assertions

---

## CI/CD Pipeline Integration

Automation is critical. The tools differ significantly in execution strategy.

1. **Postman CLI:** Requires syncing collections to Postman Cloud first.
2. **Insomnia CLI (Inso):** Works with Git, but is currently in an experimental state.
3. **Bruno CLI:** Runs directly from the filesystem.

**GitHub Actions (Bruno):**

```
    - name: Setup Node.js
      uses: actions/setup-node@v4
      with:
        node-version: '20'
    - name: Install Bruno CLI
      run: npm install -g @usebruno/cli
    - name: Run API Tests
      run: cd DummyJson && bru run
```

---

## Comparison Summary

### Postman (The Enterprise Suite)

- *Strengths:* Rich feature set, Mock Servers, Documentation hosting.

- *Weaknesses:* Bloated, Slow, Forced Cloud Sync.

### Insomnia (The Middle Ground)

- *Strengths:* Design, Plugin Ecosystem.
- *Weaknesses:* Recent shift to cloud-first, unstable CLI.

### Bruno (The Developer Tool)

- *Strengths:* **Speed**, **Offline-only**, **Git-Native**, Privacy.
- *Weaknesses:* Newer ecosystem, fewer "cloud" collaboration features.

---

## Recommendation

**Orchestration Strategy:**

- **Choose Postman if:** Your team requires non-technical collaboration, hosted documentation, or enterprise-grade mock servers.
- **Choose Insomnia if:** You prefer a polished UI and plugin extensions, but can tolerate some CLI instability.
- **Choose Bruno if:** You prioritize performance, strict data privacy, and want to manage API collections like code (Git) within your CI/CD pipelines.

**Final Verdict:** *Bruno is the superior choice for developer-centric workflows.*