Comparison of IDEs For Exploring and Testing APIs

Postman vs. Insomnia vs. Bruno

Patrik Valentiny

Project Overview

Evaluate and compare Integrated Development Environments (IDEs) for API testing.

The Contenders:

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

Tool comparison methodology

The comparison evaluates the tools across five key dimensions:

1. **Performance:** Overall tool responsiveness.
2. **Maintainability:** Organisation, environment setup, and Git/versioning support.
3. **Security:** Local vs. Cloud storage of credentials and environment variables.
4. **Testing Capabilities:** Assertions, scripting framework, and reporting.
5. **Extra Features:** Quality of Life Enhancements.

Performance

Fast development and test feedback loops are essential for Agile development.

- **Bruno:** Instant startup and local file storage allow rapid iteration.
- **Insomnia:** Cloud sync introduces latency, but generally responsive.
- **Postman:** Heavy resource usage and slow startup can impede rapid iteration.

Maintainability

Enable "Tests as Code" (Q1) to ensure reproducible and versioned suites.

- **Version Control: Bruno** uses plain text collections for superior Git integration, while **Postman's** cloud sync complicates versioning.
- **Environment Management:** All tools offer robust environment variables; **Bruno** uniquely adds request-level variables for deeper control.
- **Workflow & Migration:** All support dynamic chaining for complex **integration scenarios**, and both **Insomnia** and **Bruno** import Postman collections easily.

Security

Protecting test data and credentials is necessary in order to allow safe testing of production systems.

- **Bruno:** Offline-first design ensures sensitive production credentials never leave the machine without explicit user action.
- **Postman/Insomnia:** Cloud synchronization risks accidental exposure of secrets.

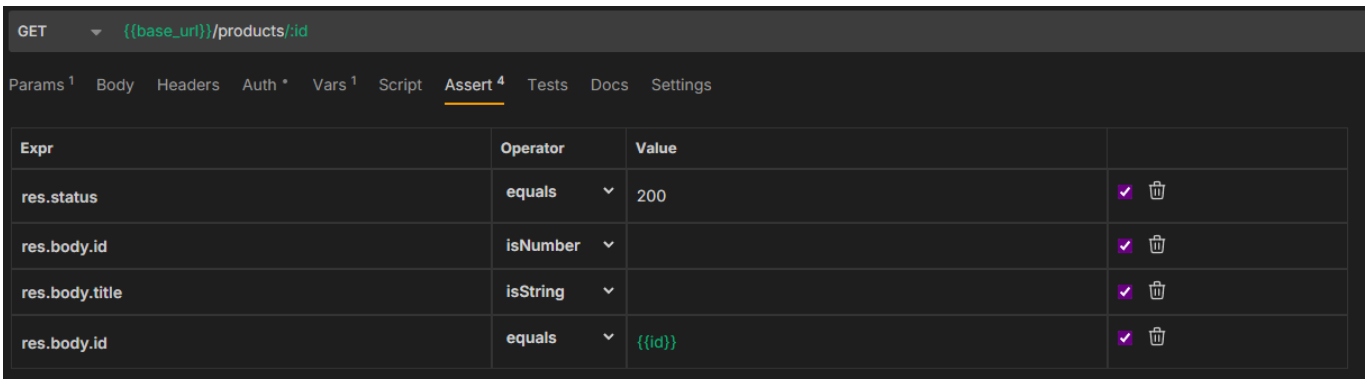
Testing Capabilities

Automated assertions drive Verification ("Building it right").

- **Assertions:** All three tools support the standard **Chai** library syntax.
- **Organization:**
  - **Bruno:** Separates tests from scripts for better readability and supports **no-code assertions** (accessible Q2 testing).
  - **Postman/Insomnia:** Tests are mixed within post-response scripts.
- **Performance Testing:** Only **Postman** offers built-in load testing suite.

Example: Writing Assertions

```
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});
pm.test("Response time is less than 200ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(200);
});
```



Extra Features

Advanced features like mock servers enable Shift-Left testing, allowing testing to start before implementation finishes.

- **Postman:** The most feature-rich
  - Mock Servers for simulating API responses.
  - AI-assisted test generation and code snippets.

    - Integrated collaboration tools for team environments.
- **Insomnia:** Strong plugin ecosystem for customization. Built-in design tools for API schema creation.
- **Bruno:** Minimal features, focusing on speed and privacy.

CI/CD

Continuous Integration enables automated API tests to run on every code change, ensuring early detection of issues.

- **Bruno:** Git-native approach allows seamless integration with existing CI/CD pipelines.
- **Postman:** Provides CI/CD integration via Postman CLI, but relies on cloud services.
- **Insomnia:** Experimental CI/CD tools, less stable.

```
steps:
  - uses: actions/checkout@v4
  - name: Install Postman CLI
    run: curl -o- "https://dl-cli.pstmn.io/install/linux64.sh" | sh
  - name: Login to Postman CLI
    run: postman login --with-api-key ${{ secrets.POSTMAN_API_KEY }}
  - name: Run API tests
    run: postman collection run "29167626-4746e42d-43a8-40d6-9d8a-d24b44531c54"
```

```
steps:
    - uses: actions/checkout@v4
    - 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:** Industry standard, Rich feature set, Mock Servers, AI assistance.
- **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, no cloud collaboration features.