APIFORTRESS

Load Testing with API Fortress

Problem

API quality is imperative to any organizations moving towards microservice architectures. That quality includes functional and end-to-end tests, but should not stop there. Performance is another key component to delivering APIs successfully to customers, and load testing is a major part of that. Waiting until an API is in production to evaluate how it works with thousands of concurrent customers is a dangerous risk that leads to lost sales and customer reputation.

API programs are more than endpoints, but a collection of endpoints that must interact with each other properly. That is why it's important to go beyond just hitting the API, but reproducing actual user flows to verify load and functionality.

Solution

API Fortress allows test engineers and developers to reuse existing functional and end-to-end tests as load tests. Not only does this save time by eliminating the need to rewrite complicated tests, it ensures that load tests simulate real user behavior. Gain accurate insight as to how your APIs react to actual concurrent users, and if every object and piece of functionality of your APIs is delivering as expected in terms of accuracy and performance.

Key Benefits

Easy Setup

- Generate tests from payloads or spec files
- Use existing tests
- Eliminate learning curves with a simple UI

Detailed Reporting

- See the fetch and latency metrics as users ramp up
- Learn if your APIs respond quickly & correctly under load
- Functional and load test results in a single report

Simplified Collaboration

- Centralize tests and reports on a single platform
- Manage user roles via a web-based platform
- Collaborate seamlessly across departments

Scalability and Monitoring

You can deploy API Fortress on-premises for unlimited load testing on any number of load agents or virtual users. No disruptions to infrastructure or workflows are needed to run load tests that simulate real user scenarios. Run the load tests adhoc, or as part of your CI/CD pipeline thanks to APIs and command-line tools.

Load testing is important, but it shouldn't be complicated. Use your existing tests to truly validate an API under load.