API Virtualization - Speed Software Delivery

Introduction

This white paper describes how teams can use service virtualization for APIs to accelerate their software delivery process by starting tasks earlier on in the product lifecycle.

Background

With the increasing trend of systems connected by web APIs, software teams looking to speed their delivery process via shorter Agile sprints have less time between development and testing phases. Traditionally, testing only begins when there is something to test, an early alpha build deployed to a testing environment. In waiting for development to produce an initial implementation, downstream members of the team waste valuable time and often fall out of cadence with sprint or goal timeframes.

Likewise, agile teams often work on multiple overlapping projects at once, some dependent on others to complete work required for the next project. Significant delays in delivery can cause multiple projects to be delayed due to a single project interruption, especially so with API projects as they are often a root dependency to front-end development and testing processes.

Solution

Service virtualization of specifically API dependencies (aka. "API virtualization") has become an increasing trend with the exponential rise of APIs as digital products and as middleware between consumer experiences in front-end applications and back-end systems. Providing control back to developers and testers over how to simulate various conditions in data, availability, and performance ensures that a team's quality strategy fits the project at hand.

Mainstream enterprise service virtualization vendors often assume the need to virtualize components that modern web and mobile developers don't directly interact with but through APIs, such as mainframes, databases, and ETL systems. Simply mimicking the necessary behavior of an API to build a testing strategy or complete a design process does not actually require a complex and costly full stack solution.

With API virtualization, front-end designers, testers, and system integrators can begin their work as soon as developers produce an initial specification (WSDL, RAML, Swagger, etc.) early on in the design process. Using virtual APIs:

- testing teams can immediately begin to build out test artifacts
- front-end designers can establish designs around realistic mock data
- system integrators can establish expected behavior characteristics between new systems

Many organizations have adopted aspects of Agile methodologies, specifically structuring teams based on project, not based on department. API projects often require one or more developers, testers, and operations personnel, and therein operate autonomously to expedite progress. While the larger organization may already have full stack service virtualization in place for other projects, API virtualization can still dramatically expedite time to delivery of API project-based teams by providing a streamlined and lightweight approach to simulating just the necessary technologies involved in an API project. Far fewer skills and much less knowledge about an enterprise service virtualization strategy is required for these teams to inherit the benefits of virtualizing one or more of their APIs.

API virtualization is a ground-up attitude to mimicking system behaviors and does not assume that an entire environment is required in order to simulate realistic behavior, just what is necessary. This approach minimizes unnecessary work and maintenance of systems, thereby adding to the time savings on projects to ultimately speed delivery.

Advertisement

Conclusion

The proliferation of APIs in web, mobile, desktop, and cloud solutions demands that teams look to optimize their capabilities around faster delivery. Lightweight virtual APIs overcome the burden of cost, skill, and implementation time associated with full stack service virtualization, and empower teams with the ability to begin parallel work earlier, exercise fine-grain control over service behavior, and share resources as necessary, to achieve fast results with minimal skills or cost.