

Document Title: How to measure different aspects of an API Strategy

Version 1.0

Existing Documents and Content

<https://docs-snaplogic.atlassian.net/wiki/spaces/SD/pages/1402142725/SnapLogic+API+Management>

Authors: [Guy Murphy - SnapLogic Enterprise Architecture Team](#)

How to measure the success of an API strategy?

When measuring the effectiveness and success of an API strategy, several Key Performance Indicators (KPIs) can be used to provide valuable insights into its performance. Historically integration KPI's have been overwhelmingly technical in nature, for API Strategies there needs to be a wider set of measurements. Business Impact, API usage and adoption metrics to broader business goals. For example, you could assess how API usage affects customer acquisition, retention, or revenue generation. This for many Enterprises API strategies are aspects of a wider strategy such as new digital products and services or internal IT strategies such as Composable Enterprise, Data Mesh and/or Data as a Service. Another concept is that of the Community that is the developers that use an API.

The specific KPIs you choose may vary depending on your organization's strategy and the types of services that the API's strategy supports.

Technical KPI's

API Usage Metrics: Keep track of the number of API calls made over a specific period. This metric can give you an overall idea of how much your APIs are being utilized by developers or consumers.

API Re-Use: Measure how many API's are supporting multiple use cases, this can correlate a reduction of p2p patterns.

User Engagement: Measure the number of unique users or applications accessing your APIs. Understanding who is using your APIs can help you target your efforts better.

Response Time and Latency: Monitor the average response time and latency of your APIs. Faster response times usually lead to higher user satisfaction.

API SLA: While one of the most basic metrics, API Service Level Agreements (SLA) notably measure when SLA's are not achieved.

Total pass and error rates: Measuring how often APIs trigger HTTP error (non-200) status codes helps you understand how error-prone your APIs may be. This aggregate measure provides information to help judge the overall quality of the APIs

API Traffic by Source: Analyse the sources of API traffic, such as mobile apps, web applications, or partner integrations. This information can help you prioritize support and improvements for integrations.

API usage growth – This metric also measures API adoption and is often the preferred metric for doing so. Ideally, API traffic grows monthly as the number of applications and developers using them also increases

API Version Adoption: Keep an eye on the adoption rate of new API versions. Encouraging developers to use the latest version can help manage technical debt and ensure better functionality.

Security and Compliance Metrics: Keep track of security-related metrics, such as the number of security incidents, security vulnerabilities found, and API access logs to ensure data privacy and compliance.

Cost Efficiency: Analyse the cost of maintaining and operating your APIs in relation to the value they bring to your organization. Assess whether the investment in API development aligns with the expected returns.

User/Community Support

One of the critical aspects of an API strategy is the ability for projects and developers to self-serve to API in the appropriate manner.

User Engagement: Measure the number of unique users or applications accessing your APIs. Understanding who is using your APIs can help you target your efforts better.

Developer Onboarding Time: Measure how long it takes for developers to start using your APIs after registration. A shorter onboarding time implies developer-friendly documentation and an intuitive API design.

API Documentation Quality: Evaluate the clarity, completeness, and ease of understanding of your API documentation. High-quality documentation can enhance developer experience and attract more users.

Community Feedback: this indicates active support and interest in using API's

Community Response Rate: How quickly are community questions resolved.

Product Metrics

For API's that are digital services that are partner or customer facing a different set of KPI's should be considered to those previously described

Direct and indirect revenue: These metrics target the different ways APIs contribute to revenue. While some APIs are directly monetized, others support integrations with business partners or are third-party integrations valued by customers. As with the adoption rate for your APIs, tracking indirect revenue helps developers build revenue-generating apps for partners.

Applications per API: APIs need to be reusable. This metric measures how many applications integrate with an API to see which APIs provide the most value.

Number of partners: APIs often enable business relationships. Tracking the number of partner API integrations helps drive adoption and demonstrate value to other business units.

Partner-Developer churn: This can be an indicator that the service, support or business case of an API service should be reviewed in depth.

Service Failure Rate: Track the rate of errors and failures occurring in API calls this should include the incorrect or poor data quality of the service.

Conclusion

There are several different dimensions to measuring the success of an API strategy, these will often be aimed at different personas within the organization, these are some common examples of such measurements but are not exhaustive especially as measurement of a product is and should be highly reflective of the service exposed by the API(s) and its impact on the process/business.