



Approaching Your API Strategy

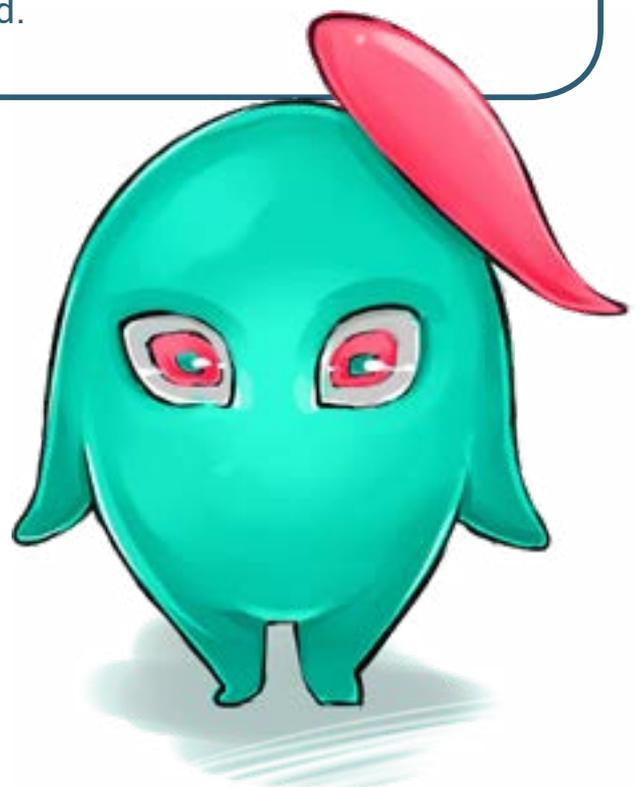
About the Author

James Higginbotham

James Higginbotham is an Executive API Consultant with experience in API strategy, software architecture, and training teams in API and microservice design. James guides enterprises through their digital transformation journey to deliver a great customer experience. James has published several titles, including his latest book "A Practical Approach to API Design". You can learn more at <https://launchany.com/services>.

Powered by Tyk

Tyk is a leading Open Source API Gateway and Management Platform, featuring an API gateway, analytics, developer portal and dashboard. We power millions of transactions per day, for thousands of innovative organisations including AXA, Cisco, Starbucks, and the Financial Times. Fast, flexible, and highly-performant, Tyk is available to install on-premises, as a cloud service, or hybrid.



Contents

Executive Summary	4
Discipline 1: Define your API Strategy	5
Executive-Driven	5
Clear Objectives	6
Communicated Often	7
Case Study: Lack of Executive Support	7
Case Study: Executive-Driven API Initiative	8
Discipline 2: Create Organisational Alignment	9
Product-Centric Delivery Model	9
Cross-Functional Team Ownership	12
Training Support	13
Discipline 3: Manage Your API Programme	15
API Governance	15
Portfolio Management	17
API Lifecycle Management	18
Case Study: Successful API Portfolio Management	19
Discipline 4: Focus on API Adoption	20
Quick API Discovery	20
Clear Onboarding Process	21
Case Study: Successful API Portfolio Management	23
Discipline 5: Accelerate your API Programme	25
Engage All Cross-Functional Roles	25
Support Solution Teams	26
Migrate to Direct-to-Customer Support Model	26
Getting Started with Your API Strategy	28

Executive Summary

APIs exist at the intersection of business, product, and technology. It isn't enough to leave your API strategy up to your technology leaders—it requires a holistic approach across the entire organisation.

In this report, we will examine the five key disciplines that lead to a successful API programme:

1. Define Your API Strategy
2. Create Organisational Alignment
3. Manage Your API Programme
4. Focus on API Adoption
5. Accelerate Your API Programme

Each of these disciplines build upon one another and are supported by 12 principles, which we will learn about throughout the report.

Applying these disciplines will result in an API programme with:

 <p>A clearly defined strategy</p>	 <p>Well-designed, discoverable, and properly documented APIs</p>
 <p>A clear path for developers to get started quickly</p>	 <p>Increased API adoption and reuse</p>

Discipline 1: Define your API strategy

APIs are more than a set of technologies. They power customer experience, business relationships, and internal innovation.

A clear API strategy transforms your data and processes into digital capabilities with a programmable interface.

Effective API strategies are executive-driven, have clear objectives, and are communicated often.

Executive-Driven

Some API programmes begin from the bottom up, when IT staff are tasked with integrating existing systems and supporting mobile applications. Whilst often an effective way to launch an API programme initially, programmes that begin within IT must quickly obtain executive buy-in to realise longevity and maximum impact.

Unless equipped with sufficient budget and leadership, these programmes can become stagnant or fragmented.

API programmes that are not driven at the executive level will be limited in budget, resources, and overall effectiveness.

Clear Objectives

An effective API strategy must first and foremost define the objectives or the entire API programme.

Common API programme objectives include:

- **Accelerate mobile strategy** by making data and services more accessible.
- **Adapt to changing customer relationships** that go beyond web and mobile to new devices and experiences (omnichannel).
- **Transform partner integrations** by improving efficiency and freeing up resources.
- **Foster technical and business innovation** by reducing technical barriers to delivery of new solutions.
- **Break down silos** to facilitate easier and more consistent data sharing among internal teams and systems.
- **Develop scale by** increasing reuse, and reducing lines of code written.
- **Convert competitors into partners** by allowing them to build products on top of your API.
- **Empower users** by enabling new workflows for doing things that weren't previously available.
- **Increase revenue directly or indirectly** by reducing customer churn through deeper integration.

While teams may design and deploy internal and external APIs on their own, without clearly defined objectives they will not align with the organisation's goals.

Seek to clearly define the objectives of your organisation's API programme first, as they will keep teams focused on delivering business value that meet these objectives.

Communicated Often

After establishing your API programme objectives, clearly document them and share them throughout the organisation.

Identify key performance indicators (KPIs) that measure success against your objectives. Track your KPIs over time and adjust your execution accordingly to meet these objectives. Communicate the objectives as often as possible across the organization.

Review them in executive and leadership meetings on a monthly, quarterly, and yearly basis to demonstrate programme progress and to ensure continued alignment with market needs.



Case Study: Lack of Executive Support

An education content provider received multiple requests to allow for custom API-based integrations with school districts around the US. Driven by partner demand for an API platform to provide more robust integrations, the provider began to deliver APIs to solve specific integration needs. However, their API programme remained focused within the IT department and did not receive wide support from the executive level.

Due to the lack of executive support, the API platform stagnated, with limited resources and budget. The IT staff delivered APIs that solved some specific integration needs, but they failed to deliver a robust API platform. Their API programme created minimal impact to both the organisation and their partners. Integration initiatives still require custom development efforts on a per-customer and per-partner basis.



Case Study: Executive-Driven API Initiative

A US-based banking institution with more than 10,000 developers rolled out a multi-year, CEO-driven API programme. This programme required transforming the culture from database-centric software design and toward an API-centric approach.

The executive team championed developing a federated API governance model, migrating outdated job roles to support an API-centric approach, and creating tools to support the processes required for such a large development focus.

The organisation produced thousands of API endpoints across multiple lines of business that drive internal innovation and new customer experiences, including voice and chat support.

SUMMARY

*Establish clear reasons why the API strategy exists. Make sure the strategy is easy to understand and applies to all job roles.
Communicate the API strategy often, and throughout the organisation.
Achieve executive team support to maximise API programme success.
The result is a more cohesive portfolio of APIs that deliver value to the organisation, its partners, and its customers.*

Discipline 2: Create Organisational Alignment

With objectives now clearly defined, the next step is to ensure we align the organisation to these objectives in order to enable API programme success.

Organisational alignment involves three key elements: a product-centric delivery model, cross-functional team ownership, and training support.

Product-Centric Delivery Model

IT includes web and mobile app development to drive a variety of customer engagement channels, along with the traditional system integration and internal solution development.

Traditionally, teams manage the delivery of projects that have a fixed timeline and budget rather than products that continue to grow and mature over time. This is changing as organisations adopt a product-centric delivery model.

Let's review the differences between projects and products.

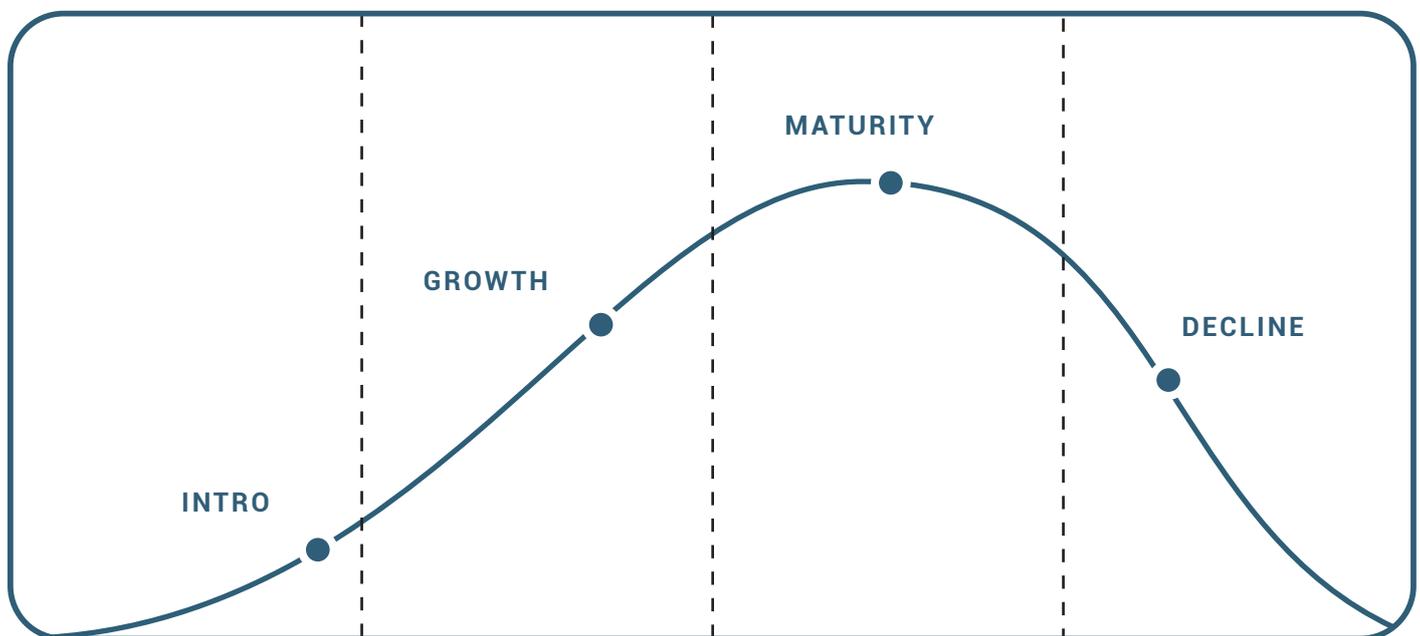
PROJECTS ARE:
Fixed-budget
Output-based
One-off solution
Internally focused
Date-driven
No continuous improvement
Ended with maintenance
Rarely marketed

PRODUCTS ARE:
Variable-budget
Outcome-based
Repeatable and reusable
Customer focused
Results-driven
Continuously released and improved
Ended when the need ends
Evangelistically marketed

An effective API strategy considers APIs as *products*, not projects. These API products are digital assets (Fig. 1.1) that belong to a growing and evolvable portfolio of digital capabilities.

APIs offer contracts between both internal developers and partners. These contracts are more than a technical concern—they require product ownership that seeks to grow and mature the APIs over time.

(Fig. 1.1) APIs don't end with version 1.0 - They grow and mature over time.



APIs that are treated like projects do *not* grow and mature over time. After they are delivered, support is limited to maintenance developers assigned to the project. Support offered to integrating teams is limited to the maintenance team's knowledge rather, than the original team that built the API.

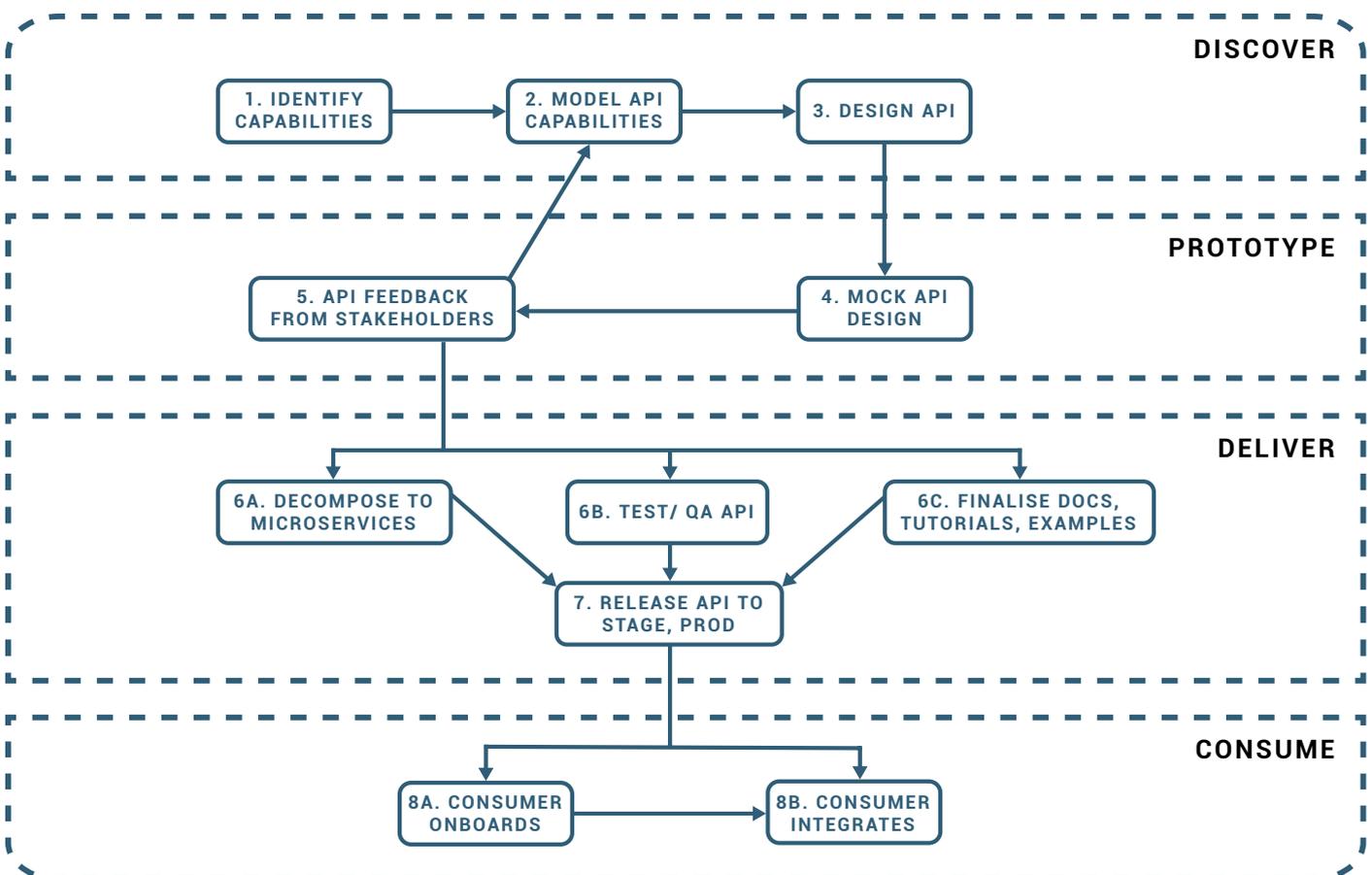
Product ownership is essential to increased API adoption. It moves teams beyond thinking about technology and raw data to solving business problems. API product ownership focuses on identifying stakeholders, continually identifying their needs, and delivering the API capabilities that meet those needs.

Cross-Functional Team Ownership

Many large-scale organisations have adopted the “you build it, you own it” model for software delivery made popular by the DevOps movement. Under this model, software delivery is no longer a serialised process across the silos of design, development, QA, and deployment.

Instead, teams consist of cross-functional roles that address all aspects of the delivery lifecycle. Team members cooperatively (Fig. 1.2) design APIs and execute necessary work in parallel. Cross-functional teams use OpenAPI Specification (OAS) definitions to communicate the API design to the entire team, shortening communication time as well as API delivery time. Product owners work with stakeholders to gather downstream needs and validate that the API design meets their solution needs.

(Fig. 1.2) API first design applies cross-functional team ownership, not just developers.



Moving to cross-functional team ownership and a product-centric delivery model requires shifting from the traditional brokered model used for service-oriented architecture (SOA) to a direct API consumption model.

In the direct API consumption model, teams release new APIs for solution teams to discover and consume. Solution teams discover these APIs, request access, and integrate them with little or no human involvement. If solution teams need help, they communicate directly with the API team rather than an intermediary team.

Training Support

It is essential to deliver a training programme that prepares the way for an API-centric organisation. Starting from the API programme's inception, it is critical to equip teams with awareness of the programme, the skills necessary to produce and consume APIs, and new processes and tools available to support the programme.

[An external organisation](#) or internal team may deliver the training. For large organisations, a combination of on-site training, video-based training, and coaching can accelerate an API programme at scale.

[Lessons from Training 1400+ People in Web API Design](#) details the following insights for building an API training program:

- **APIs involve everyone, including IT and business.** A variety of roles need to be trained, including developers, architects, QA teams, technical writers, product managers, product marketing, and scrum masters.
- **Include the fundamentals.** Do not assume that everyone knows what APIs are and how they work.
- **Focus on delivering value.** Training teams in product thinking leads to moments of clarity that deliver value and differentiate market leaders.
- **Train for the full API development lifecycle.** Training should discuss how various job roles are involved in the different stages of API development, including planning, design, development, testing, marketing, and management.
- **Focus on API ownership.** Many API training programmes spend a considerable amount of time describing the technical details of API design; instead, incorporate API ownership alongside design techniques to deliver high-value APIs.

SUMMARY

Shift toward product-centric thinking and cross-functional API ownership. Establish a consistent training programme that prepares teams for an API-centric approach.

Discipline 3: Manage Your API Programme

API programmes must be managed and maintained properly to ensure teams remain aligned on the strategy and objectives day-to-day. Otherwise, the API programme will stagnate or diverge from key objectives and ultimately fail.

Properly managing your API platform requires API governance, portfolio management, and API lifecycle management.

API Governance

SOA governance often brings to mind rigid, strict, and heavy-handed enforcement of rules and processes.

Many historical SOA governance programmes were *internally* focused to remove operational redundancies. Modern API programmes are *externally* focused to drive revenue and business value, which requires a new approach to API governance.

A healthy API governance initiative should encourage consistency across the organisation, mixed with flexibility to support changing requirements.

Effective API governance programmes include the following principles as part of their API governance:

- **Coach teams on API modeling and design techniques**, resulting in self-sufficient delivery teams.
- **Deliver educational material, training, and other resources** to communicate shared knowledge.
- **Empower solution teams** to discover and consume existing APIs.
- **Define clear API standards, protocols, and design patterns** supported by the organisation.
- **Create policies for onboarding, rate limiting, and access control.**
- **Craft flexible processes and practices** that encourage innovation.

Organisations with a large number of development teams or teams that are distributed across multiple regions or countries benefit from a federated governance model. In this model, API governance team representatives are part of a specific business unit or regional group that provide context-specific guidance and coaching.

The centralised API governance team works with federated API coaches to gain insight into emerging needs, improvements, and clarifications required to maintain consistency across the organisation.

This provides speed and flexibility while supporting the varying needs of different business units and regions.

For some organisations, API governance may be centralised and managed by a single team for the life of the API programme. For larger organisations, governance may be centralised at first but evolve to a more federated approach over time to scale the process.

Portfolio Management

It is essential to manage your API portfolio to ensure your API programme continues to meet stakeholders' needs. Common practices from successful organisations include:

- **Select a team to own the overall API portfolio**, whether it is the same team that manages API governance or a different team.
- **Define a clear process for adding new APIs into the portfolio**, and make sure to include a review stage to verify standards compliance and design consistency.
- **Manage URL paths under your API domain** (e.g., api.mycompany.com) to prevent scattered API endpoints and resources and confusing or conflicting names.
- **Routinely review analytics** to clarify API usage, areas for improvement, and customer and partner journeys. Receiving analytics also allow you to identify high-value candidates for future APIs.
- **Partition the portfolio into domain areas** (e.g., customer accounts, orders, and inventory for an eCommerce platform) and assign ownership of each domain to reduce centralised management.

With properly executed API portfolio management, teams across the organisation can contribute to the overall portfolio. Teams will also have confidence in terms of where they fit within the whole API portfolio and will not duplicate efforts.

API Lifecycle Management

Like all software, APIs have a lifecycle that starts at inception and extends to production and beyond. Successful organisations establish a clear API lifecycle that benefits stakeholders and accelerates delivery.

A typical API lifecycle includes the following seven stages:

1. **Discover:** Identify new API capabilities that stakeholders need.
2. **Design:** Refine needs with stakeholders, identify capability requirements, and design API to meet needs.
3. **Deliver:** Implement, test, and verify that new capabilities meet stakeholder expectations.
4. **Manage:** Deploy, monitor, and manage APIs at runtime.
5. **Market:** Inform existing consumers and build awareness with potential consumers.
6. **Enhance:** Incorporate consumer feedback and use proper versioning techniques to enhance the API without disrupting existing consumers.
7. **Deprecate:** Inform consumers of any plans to stop supporting an existing API, migrate consumers to a new API (if available), and retire the API.

Integrate your preferred software processes throughout the API lifecycle. Don't invent new processes unless they are necessary but be prepared to adjust processes to prevent negative effects for existing API consumers.



Case Study: Successful API Portfolio Management

A worldwide hotel chain began constructing a new hospitality platform to address the needs of the organisation, its property managers, and customers. Their API portfolio consisted of several domain areas, including account management, property management, and reservation booking.

By clearly defining and managing their API portfolio, they were able to focus on first delivering high-value capabilities with a small team. As the platform grew, they onboarded new teams to address additional areas of the portfolio in parallel.

With clear boundaries and oversight by key leaders and their enterprise architecture team, the platform grew from a few endpoints to hundreds of endpoints within the first year.

SUMMARY

Establish API governance early. Build in API portfolio management to encourage platform growth and quality. Apply an API lifecycle that encourages rapid delivery combined with stakeholder involvement.

Discipline 4: Focus on API Adoption

API consumers are an essential ingredient of any programme. However, many organisations focus on strategy, objectives, and governance to create APIs without addressing the need to continually grow platform adoption.

Focusing on API adoption helps organisations avoid spending time and resources to construct APIs that are rarely (or never) used by developers who are unaware of their existence.

To support API adoption, organisations should thoroughly document their APIs with a developer portal that encourages quick API discovery and establishes a clear onboarding process.

Quick API Discovery

For most teams, API documentation is an afterthought, with serious consequences for API discovery and adoption.

Your documentation is the first encounter most developers will have with your APIs, so providing great documentation is essential to helping them understand what your API offers, how to use it, and what to do when they are ready to start integrating.



Follow these five steps to achieve API documentation success:

1. Deliver comprehensive API reference documentation using the OpenAPI Specification (OAS) or a similar standard: Without reference documentation, developers cannot use your API.
2. Launch a developer portal that addresses the needs of both business and developers, and which drives marketing and recruiting opportunities. The developer portal is your communication hub for the community that grows around your API.
3. Integrate documentation into your delivery process. If it isn't documented, your API isn't done.
4. Refine the portal to make sure it includes the right mix of artefacts. Include reference documentation, a straightforward guide for getting started, case studies or walkthroughs, and other resources to accelerate developer success.
5. Add content for non-technical decision-makers, including marketing and sales, the executive team, and product/programme managers. Help every role understand how the API addresses their needs.

Clear onboarding process

Developers who are new to your API platform do not have an easy journey. In fact, development teams go through several phases as they evaluate and integrate your API, as you can see in the table overleaf (page 21).

CONSUMPTION PHASE	GOAL
Onboarding	Register for portal and API access
Discovery	Identify API capabilities
Mapping	Map solution to Platform API capabilities using reference documentation
Exploration	Prototype consumption ("Try-it-out")
Integration	Consume via code
Certification	Obtain approval for production API access
Usage Monitoring	Production access monitoring and throttling for compliance
Platform Improvement	Request platform API enhancements to meet the needs of the solution
Platform Updates	Update notifications for new API endpoints, enhancements, case studies

To ensure developers can quickly get started with using your API, define a clear onboarding process.

Set the expectations for the path from discovery to mapping and integrating their solution to your API. Don't stop with getting developers to use your API initially—stay in contact with them through newsletters or distribution lists. Announce new and upcoming improvements, success stories, and common use cases in your developer communications.

Highlight the teams that are responsible for building and supporting your APIs to demonstrate your commitment to meeting developers' needs (and build team ownership of APIs at the same time).



Case Study: Successful API Portfolio Management

An API programme initiative for a large enterprise IT group started with just a few key people. After a year of investment, the team had produced several APIs that offered a number of high-value capabilities to the business. However, the team produced only reference documentation—no developer portal. As a result, information about how to start using the API wasn't readily available. With help, the team expanded the reference documentation into a complete developer portal.

Their revised developer portal guides developers through an introduction to the API's structure and capabilities, onboarding in a sandbox environment for integration, and production access through a lightweight certification programme.

Influential executives use the developer portal to evangelise the API programme throughout the organisation, resulting in increased demand for adopting APIs. The developer portal now serves as a communication tool and a method of promoting the programme to technical and non-technical teams.

SUMMARY

Don't assume that reference documentation is sufficient. Deliver a comprehensive developer portal that covers the full end-to-end process for discovering, onboarding, and consuming your APIs. Make sure that developers can get started quickly and encourage continual communication for those who want to stay informed.

Discipline 5: Accelerate Your API Programme

After their API programme is underway, organisations use a few techniques to accelerate it. These include engaging all cross-functional roles, supporting solution teams throughout the app development lifecycle, and migrating to a direct-to-consumer support model.

Engage All Cross-Functional Roles

An organisation's API governance and portfolio management groups are not the only teams that need to be involved in the day-to-day tasks of an API programme. Other teams that should be involved include:

- **Security Team:** Review API designs for personally identifiable information (PII) and nonpublic information (NPI) concerns, identify risks, and approve or decline APIs and apps that consume sensitive data.
- **Infrastructure/DevOps:** Ensure network, server, container platforms, message brokers, and other necessary resources are available for the teams that are building and consuming APIs.
- **Engineering:** Coach teams on common practices, patterns for API consumption, and delivery.
- **Programme Management Office:** Help with the overall platform process and implement change management as needed.

Support Solution Teams

High-performing organisations take solution teams into consideration when designing their API programme. These organisations:

- **Establish clear processes and roles** for quickly onboarding consumers.
- **Provide coaches or solution architects** who help teams successfully build APIs.
- **Provide a developer relations (DevRel) team** to coach solution teams on consuming APIs, including proper error handling and retry strategies

A common problem for organisations that are struggling to grow their API programme is failing to provide the necessary support, training, and coaching to the teams that are consuming the APIs. Lack of solution team support will limit programme growth, hinder adoption, and cause teams to build their own APIs rather than consume existing APIs.

Migrate to Direct-to-Consumer Support Model

As part of an overall digital transformation effort, organisations are moving to dedicated product development teams rather than following a project-based approach. Developers remain on the team after the initial API version rather than passing the API to a maintenance team.

Each team has a product owner who listens to stakeholders, monitors KPIs, and adjusts the API roadmap to meet stakeholders' needs. This shift from a project-centric approach to a product-centric approach is an extension of the "you build it, you own it" mantra common with the DevOps movement.

Without a product-driven approach, APIs become stale and less useful, resulting in frustrated teams that are unable to find an API to solve their problems. Project-driven APIs are also often duplicated, resulting in reproduced data sources and wasted efforts.

SUMMARY

Engage the other roles within your organisation as part of your API programme. Support solution teams to ensure they have the knowledge and resources necessary to successfully find and integrate APIs. Ideally, shift to a product-centric approach to API ownership, with solution teams working directly with API owners to encourage a “you build it, you own it” mindset.

Getting Started with Your API Strategy

In this report, we have covered the five disciplines of a successful API programme:

1. Define Your API Strategy
2. Create Organisational Alignment
3. Manage Your API Programme
4. Focus on API Adoption
5. Accelerate Your API Programme

Each of these disciplines build upon one another and are supported by 12 principles, summarised below:

1. **Establish why the API strategy exists.** Ensure it is easy to understand and apply to all job roles.
2. **Communicate your API strategy often across the organisation.** Obtain executive team support to maximise API programme success.
3. **Establish a consistent training programme** that prepares teams for an API-centric approach.
4. **Establish a lightweight API governance model** that ensures consistent API design and standards compliance.
5. **As your API programme grows, implement a federated governance approach** to support large development teams distributed across business lines and varying regions
6. **Build in API portfolio management** to encourage platform growth and quality.

7. **Apply an API lifecycle that encourages rapid delivery** combined with stakeholder involvement.
8. **Deliver a comprehensive developer portal** that covers the full end-to-end process for discovering, onboarding, and consuming your APIs.
9. **Make it quick to get started with your APIs** and encourage continual communication for those who want to stay informed.
10. **Engage other roles within your organisation** in your API programme.
11. **Support solution teams** to ensure they have the knowledge and resources necessary to successfully find and integrate your APIs.
12. **Shift to a product-centric approach to API ownership**, with solution teams working directly with API owners to encourage a “you build it, you own it” approach.

While it may be challenging at times to launch and sustain an API programme, it is ultimately rewarding as you see your organisation shift from a set of adhoc APIs to a healthy programme. Apply these 12 principles discussed throughout this report to achieve a success API programme.

Find out more

If you have any questions or comments regarding this whitepaper, please contact James Higginbotham, LaunchAny, <https://launchany.com/api-strategy/>

For more information on how to implement Tyk where you are, please contact your local Tyk consultant. A list of all our Tyk consultants can be found at <https://tyk.io/about/partners/>

About Tyk

Tyk is the fastest-growing and most exciting API Gateway & Management platform on the market, comprising an open source Gateway, coupled with a proprietary management dashboard. We power millions of transactions per day, for thousands of innovative organisations including AXA, Cisco, Starbucks, and the Financial Times.

API analytics, out-of-the-box developer portal and multi-cloud capability are some of the most popular features of the Tyk Management Platform, but only just touch the surface of what this powerful product can do. Fast, flexible, and highly-performant, Tyk is available to install on-premises, as a cloud service, or hybrid. Get started at <https://tyk.io> or contact us for more information.

TYK LONDON
31 Charlotte Road, London EC2A 3PB, United Kingdom

TYK SINGAPORE
Collision 8, High Street Centre, 1 North Bridge Road,
#08-08, Singapore 179094

