

Le manuel d'utilisation des API

7 meilleures pratiques des équipes qui réussissent
à exploiter les API

Manfred Bortenschlager

Directeur, développement commercial des solutions d'intégration basées sur des API et de la gestion des API, Red Hat

Andrew Mackenzie

Directeur de la gestion des API pour l'ingénierie logicielle, Red Hat

Jaya Baskaran

Responsable principale du marketing technique, Red Hat

Greg Pack

Responsable senior du marketing produit Services d'applications et solutions de gestion des API, Red Hat



Sommaire

Synthèse	3
Introduction	3
Pourquoi voulons-nous implémenter des API ?	4
À qui ces API sont-elles destinées ?	4
Quels résultats concrets attendons-nous de ces API ?	4
Comment prévoyons-nous de mettre en œuvre notre programme d'API pour y parvenir ?	5
L'équipe API	5
Meilleure pratique n° 1 : Axer tous les efforts sur la valeur de l'API	6
Quelles sont les implications pour l'API ?	6
Aspects à prendre en compte pour définir la valeur du programme d'API	7
Meilleure pratique n° 2 : Clarifier d'emblée votre modèle économique	7
Comprendre un modèle économique d'API	7
Aspects à prendre en compte pour votre modèle économique d'API	8
Meilleure pratique n° 3 : Réaliser une conception et une mise en œuvre orientées utilisateur	9
Simplicité	9
Flexibilité	9
Aspects à prendre en compte pour concevoir le protocole d'API	9
Intégration de produits	9
Infrastructure de mise en œuvre et déploiement des API	10
Meilleure pratique n° 4 : Faire de l'exploitation des API une priorité	10
Schéma explicatif de l'exploitation des API	10
Composants essentiels de la gestion des API	11
Aspects à prendre en compte pour l'exploitation des API	11
Meilleure pratique n° 5 : Soigner l'expérience de développement	12
Une conception de qualité ne suffit pas pour correctement exploiter l'API	12
Aspects à prendre en compte pour évaluer l'expérience de développement	13
Meilleure pratique n° 6 : Définir une stratégie marketing efficace	13
Aspects à prendre en compte pour le marketing des API	13
Meilleure pratique n° 7 : Gérer les changements et le retrait des API	14
Aspects à prendre en compte pour augmenter la longévité des API	14
Pérenniser votre stratégie d'API	15
Créer un programme d'API durable	15
Les API ont besoin d'une base solide	15
Red Hat, un partenaire de choix pour la création et la gestion des API	16
Conclusion	16

Synthèse

Les progrès en matière de développement d'applications modernes, de connectivité et d'infrastructure, tels que la 5G et l'edge computing, continuent de transformer la manière dont se déroulent les processus métier. Si ces avancées accélèrent l'exploitation, seules des interfaces de programmation d'application (API) modernes permettent d'en retirer tous les avantages.

Les API sont aujourd'hui devenues le cœur numérique des entreprises modernes, avec les nouvelles capacités qu'elles ajoutent à tous les niveaux : exploitation, produits, stratégies de partenariat. Que vous disposiez déjà d'une stratégie d'API ou que vous vous apprêtiez à en définir une, nous avons créé ce livre numérique pour vous proposer sept meilleures pratiques qui vous aideront à réussir la mise en œuvre de programmes d'API.

Introduction

Avant de consulter ce livre numérique, définissez les principaux objectifs de votre programme d'API. Certaines questions abordées par la suite peuvent vous aider à revoir, valider ou préciser vos objectifs.

Pour être efficace, un programme d'API doit s'appuyer sur la stratégie globale de l'entreprise et contribuer à la réalisation de ses objectifs. Vous saurez que votre stratégie est solide si vous pouvez répondre aux quatre questions suivantes :

1. Pourquoi voulons-nous implémenter des API ?
2. À qui ces API sont-elles destinées ?
3. Quels résultats concrets attendons-nous de ces API ?
4. Comment prévoyons-nous de mettre en œuvre notre programme d'API pour y parvenir ?

Pourquoi voulons-nous implémenter des API ?

Lorsque vous envisagez de déployer des API, il est essentiel de bien comprendre leurs avantages et leur finalité, et de s'éloigner des idées reçues à leur sujet pour maximiser leur valeur métier. On croit souvent à tort qu'une API ne peut avoir de valeur que si ses utilisateurs acceptent de payer pour y avoir accès. Ceci n'est vrai que si l'API en soi constitue le produit. Or, de nombreux projets d'API sont réservés à un usage interne et servent à générer des indicateurs de mesure (délai de mise sur le marché, réutilisation, ventes, notoriété de la marque, parrainage d'entreprises affiliées, etc.).

Voici les cinq cas d'utilisation les plus courants associés aux API :

1. **Création d'interactions multicanal** : amélioration de l'accessibilité grâce à divers canaux tels que les applications mobiles, l'Internet des objets, etc.
2. **Développement de l'écosystème** : élargissement des écosystèmes de clients ou de partenaires.
3. **Expansion de l'audience** : création de réseaux étendus pour les transactions ou la distribution de contenu.
4. **Promotion de nouveaux modèles économiques** : renforcement de l'innovation.
5. **Accroissement de l'innovation interne** : soutien à l'innovation au sein de l'entreprise.

Beaucoup de fournisseurs de technologies, dont Amazon, Salesforce et Twilio, ont réussi le déploiement de leur stratégie d'API. Toutefois, ces leaders ne sont pas les seuls à savoir exploiter les API dans le cadre d'un projet de monétisation, d'innovation, d'amélioration de l'agilité ou de développement des partenariats. Les entreprises historiques utilisent souvent des systèmes traditionnels, qu'elles ne comptent pas abandonner de sitôt (si ce n'est jamais). Les API connectent tous ces systèmes, facilitent leur accès dans l'entreprise ou en dehors, et les associent à d'autres services afin de générer de la valeur. Toute entreprise qui a entamé sa transformation numérique peut bénéficier des avantages offerts par les API grâce à la mise en œuvre d'une stratégie structurée.

Les avantages doivent être suffisamment intéressants pour qu'une entreprise décide spontanément d'investir dans des API.

À qui ces API sont-elles destinées ?

Pour mieux définir les indicateurs de mesure de la réussite de votre programme d'API, il est nécessaire d'en identifier les utilisateurs finaux.

- **Utilisateurs finaux internes** : les équipes internes peuvent se servir d'API pour accéder à des informations d'entreprise en vue d'atteindre un ou plusieurs objectifs métier.
- **Utilisateurs finaux externes** : les mêmes API ou un sous-ensemble de celles-ci peuvent être utilisés par des équipes de développement tierces dans leurs applications.

Les avantages doivent être suffisamment intéressants pour qu'une entreprise décide spontanément d'investir dans des API.

Quels résultats concrets attendons-nous de ces API ?

Pour identifier des résultats concrets générés par les API, prenez en compte les visions interne et externe de l'entreprise.

- **Vision interne** : ressources spécifiques et utiles dont vous disposez. Une entreprise qui possède des données uniques, telle que Meta avec ses données sur la fonction « J'aime », peut proposer des API sources de valeur.
- **Vision externe** : dynamique du marché, tendances, concurrence et comportement des clients. Les forces externes façonnent les stratégies métier et influencent les fonctionnalités des API.

La dynamique du marché affecte profondément la mise en correspondance des API. Les entreprises qui effectuaient historiquement cette tâche ne tenaient pas compte des exigences en temps réel, ce qui a permis à des start-up telles que Waze de se développer. Google a ensuite acquis Waze et a intégré sa technologie à une API efficace. Twitter, Reddit et d'autres grandes entreprises ont également adopté les API pour stimuler l'innovation et la croissance de leur écosystème.

Une stratégie d'API adaptée associe souvent des ressources internes à des données externes sur le marché.

Comment prévoyons-nous de mettre en œuvre notre programme d'API pour y parvenir ?

La dernière question est « Comment concevoir le programme d'API pour atteindre l'objectif désiré ? » et porte sur l'implémentation et l'exécution, ce qui nécessite une planification attentive des aspects suivants.

1. **Choix technologiques** : déterminer la pile technologique pour concevoir les API.
2. **Conception et maintenance** : définir les stratégies de création et maintenance des API.
3. **Promotion et marketing** : promouvoir les API au sein de l'entreprise et en assurer le marketing à l'extérieur.
4. **Allocation des ressources** : allouer les ressources nécessaires au développement et à la maintenance des API.
5. **Composition de l'équipe** : constituer une équipe capable d'assurer le développement et la gestion des API.
6. **Communauté de développement** : créer et gérer un plan de communication dédié pour les équipes de développement internes et externes.
7. **Indicateurs de mesure de réussite** : définir des méthodes pour suivre la réalisation des objectifs métier.

En déterminant tous les tenants et les aboutissants des API, les entreprises peuvent stimuler l'innovation et la croissance au sein d'une économie des API en pleine évolution. Pour ce faire, elles devront procéder différemment en fonction de leurs objectifs, de la stratégie déployée et, surtout, de la composante la plus importante : l'équipe API.

L'équipe API

Une équipe API est généralement structurée comme n'importe quelle autre équipe produit. Que vos clients soient internes ou externes, cette équipe est responsable de la création, du déploiement, de l'exploitation et de l'optimisation de l'infrastructure sur laquelle d'autres équipes se reposent.

À l'instar des équipes produit, les équipes API peuvent aussi être très diverses, mais elles doivent comprendre un responsable produit qui veille au respect de la stratégie et des objectifs, des membres chargés de la conception qui appliquent les meilleures pratiques en matière de conception d'API, des ingénieurs qui codent les technologies API, et des collaborateurs chargés de l'exploitation qui exécutent l'API. Plus tard, d'autres membres peuvent être appelés à intervenir, par exemple des personnes chargées de l'assistance et de la communauté, des spécialistes de l'API, des spécialistes de la sécurité, etc. Votre équipe API étendue peut également inclure des membres de votre communauté de développement.

Si cette équipe peut intégrer de nombreuses personnes, certaines d'entre elles peuvent y jouer plusieurs rôles, notamment au sein des entreprises plus petites. Il est important de veiller à ce que les opinions de toutes les parties prenantes soient représentées, même si cela implique de prendre en compte des avis plus réservés.

L'équipe API est souvent créée pour une courte durée et peut appartenir à plusieurs divisions gérées par différents responsables. Cette structure peut notamment compliquer la définition d'une vision commune pour l'API. Les grands programmes d'API peuvent nécessiter la collaboration de plusieurs équipes API.

Quelle que soit la taille de votre entreprise, les sept meilleures pratiques décrites dans ce livre numérique vous aideront à constituer une équipe API efficace et potentiellement plus nombreuse qu'attendu.

Meilleure pratique n° 1 : Axer tous les efforts sur la valeur de l'API

Les programmes d'API doivent accorder la priorité à leur principal objectif qui consiste à générer de la valeur tout en évitant la complexité. La proposition de valeur détermine l'utilisabilité, un facteur de réussite essentiel de l'API, c'est pourquoi elle doit être convaincante pour réaliser un marketing efficace. Pour une stratégie durable, il est important que cette proposition soit en phase avec les objectifs métier, ce qui permettra aux entreprises établies d'optimiser leurs offres grâce aux API.

Le modèle de valeur d'API conçu par Alex Osterwalder fait correspondre les avantages et les besoins utilisateur avec les fonctions et la valeur des API¹. Il est donc essentiel de répondre aux besoins, de résoudre les difficultés et de générer de la valeur.

Quelles sont les implications pour l'API ?

Dans le cadre de ce processus itératif, la première étape consiste à décrire les tâches que vos utilisateurs souhaitent réaliser (par exemple, envoyer automatiquement des messages en cas d'urgence, sauvegarder des fichiers importants ou recueillir des échantillons de données pour détecter certains événements).

La deuxième étape consiste à identifier les difficultés que rencontrent les utilisateurs avant, pendant ou après les tâches qu'ils ont à accomplir, comme assurer la fiabilité des envois par tentatives répétées, détecter les échecs, gérer des messages multiples, réussir à intégrer différents systèmes de messagerie en fonction de l'emplacement de l'utilisateur, assurer la bonne réception des fichiers en utilisant le moins de bande passante possible et corrélérer de grands volumes de données en temps réel.

La troisième étape de la création d'un profil d'utilisateur implique de lister les avantages potentiels, comme envoyer d'autres types de notifications visant à dégager des opportunités plutôt qu'à signaler des risques, abandonner certains équipements de stockage en cas de fiabilité suffisante, et automatiser des actions déclenchées par des événements.

Reste ensuite à établir une « value map » qui met en correspondance les capacités, fonctions et services de l'API, en se concentrant sur la résolution de problèmes et la création d'avantages. Le processus peut alors aboutir à des résultats concrets : par exemple, une API de messagerie qui distribue les messages, une API de synchronisation du stockage qui assure les mises à jour et une API capable de rassembler différentes sources de données sous forme de flux configurables.

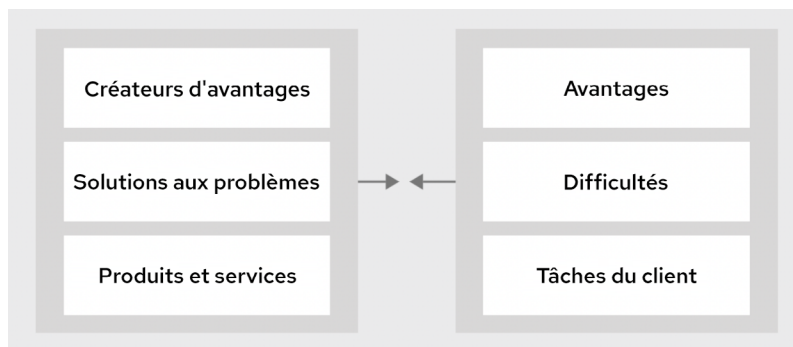


Figure 1 : schéma de proposition de valeur

Pour terminer, l'équipe API doit se livrer à un exercice de clarification consistant à rédiger plusieurs affirmations qui démontrent l'utilité de l'API pour le profil d'utilisateur visé. Lorsque vous synthétisez tous vos arguments sous la forme d'un message général, vous obtenez la proposition de valeur de vos API. Ces arguments renforcent davantage la correspondance entre l'API et l'utilisateur et reflètent la proposition de valeur. Dans le cas de l'API de messagerie, cette proposition peut prendre la forme suivante :

« La citation du client se place ici. Veillez à la longueur du texte lors de l'ajout de contenu dans la barre latérale. Certaines langues comme l'allemand prennent plus de place que l'anglais. Pour pouvoir accueillir les traductions, laissez au moins 25 % d'espace libre dans la barre latérale. »

Vous pouvez avoir l'impression que ce processus est trop lourd, surtout si vous travaillez sur une API limitée à un usage interne. Cette réaction est naturelle, mais il est essentiel d'accorder une attention particulière à la valeur, même dans des cas d'utilisation en interne. Une proposition de valeur mal définie ne vous permettra pas de convaincre facilement et rapidement les autres équipes d'adopter votre API. Si elle est bien définie, au contraire, elle permettra à votre API de devenir un atout majeur de votre entreprise.

¹ Strategyzer, « [Value Proposition Canvas](#) », consulté en septembre 2023

Aspects à prendre en compte pour définir la valeur du programme d'API

Pour définir la valeur de votre programme d'API, tenez compte des aspects suivants :

1. **Identification des utilisateurs** : identifiez les utilisateurs sur la base de leurs relations (clients, partenaires, développeurs), leurs rôles et leurs préférences.
2. **Gestion des difficultés et des avantages** : le schéma de proposition de valeur (Figure 1) vous permet de déterminer les difficultés, les avantages et les besoins essentiels. Mesurez les améliorations au niveau des indicateurs (rapidité, chiffre d'affaires, coût) et les nouvelles opportunités potentielles.
3. **Cas d'utilisation pris en charge** : utilisez le schéma de proposition de valeur pour identifier les solutions aux problèmes et les éléments créateurs d'avantages. Votre API doit répondre à ces cas d'utilisation spécifiques.
4. **Accroissement futur de la valeur** : préparez votre proposition de valeur en envisageant l'avenir. Anticipez les grandes étapes, tendances ou innovations technologiques à venir pour faire croître la valeur à long terme.
5. **Valeur organisationnelle interne** : évaluez les avantages de l'API au sein de l'entreprise et sa valeur potentielle pour les autres équipes.

En gardant ces aspects à l'esprit, votre équipe API pourra établir une proposition de valeur claire qui répond aux besoins des utilisateurs ainsi qu'aux objectifs de l'entreprise.

Meilleure pratique n° 2 : Clarifier d'emblée votre modèle économique

Pour créer une API efficace, il ne suffit pas de définir une proposition de valeur. Il faut aussi respecter un modèle économique précis. S'il est important de reconnaître et de transmettre la valeur de l'API, les coûts qu'elle engendre doivent également être évalués au regard des avantages tangibles ou financiers qu'elle apporte. Dans son livre collaboratif « Business Model Generation », Alex Osterwalder définit le modèle économique d'une entreprise comme le processus selon lequel celle-ci propose, crée, fournit et capture de la valeur.

Comprendre un modèle économique d'API

Le schéma du modèle économique détaille les principales composantes d'un tel modèle² :

1. **Proposition de valeur** : valeur unique de l'API.
2. **Sources de revenus** : manière dont l'API génère des revenus.
3. **Structure de coûts** : coûts associés à la gestion de l'API.
4. **Segments de clients** : groupes d'utilisateurs cibles.
5. **Relations clients** : comment l'entreprise crée de l'engagement avec les utilisateurs.
6. **Canaux** : méthodes de distribution pour toucher les utilisateurs.
7. **Partenaires clés** : collaborateurs importants pour l'exploitation réussie de l'API.
8. **Activités clés** : tâches essentielles à l'exploitation de l'API.
9. **Ressources clés** : ressources indispensables à la mise en œuvre de l'API.

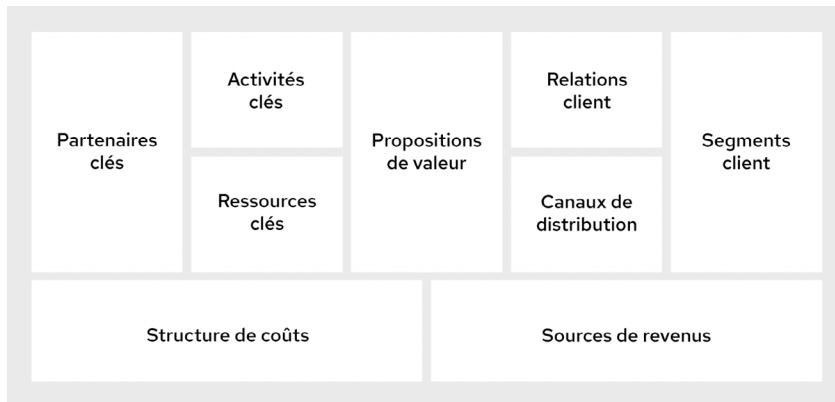


Figure 2 : schéma du modèle économique

Une API peut générer de nouvelles opportunités à partir des ressources existantes. Bien qu'il soit important de reconnaître la valeur qu'elle produit, les coûts qu'elle engendre doivent être maîtrisés. En effet, un modèle inadapté peut les faire augmenter malgré la valeur créée.

Aspects à prendre en compte pour votre modèle économique d'API

Pour que votre modèle économique soit en phase avec l'utilisation qui est faite de l'API, tenez compte des cinq aspects essentiels suivants :

- 1. Valeur de l'API pour l'entreprise :** sans vous limiter à la valeur monétaire, évaluez les différentes manières dont l'API permet à votre entreprise d'étendre sa portée ou de renforcer l'innovation.
- 2. Capture de la valeur :** déterminez le meilleur moyen de capturer toute la valeur identifiée, en éliminant le plus d'obstacles possible.
- 3. Absorption des coûts :** identifiez les coûts liés à l'API au-delà de l'équipe dédiée (ingénierie ou marketing, par exemple) et trouvez des solutions pour les absorber.
- 4. Engagement à long terme :** au-delà de l'investissement initial, gardez à l'esprit que les API nécessitent un engagement à long terme en matière d'exploitation et de maintenance.
- 5. Partenariats stratégiques :** identifiez les partenariats essentiels, et utilisez les offres complémentaires de partenaires et fournisseurs lors du développement des API et de la mise sur le marché.

Meilleure pratique n° 3 : Réaliser une conception et une mise en œuvre orientées utilisateur

La conception d'une API implique de respecter des principes de base afin d'aboutir à une expérience utilisateur cohérente. L'objectif est de parvenir à une conception d'API « prête à l'emploi » et suffisamment intuitive pour les spécialistes du développement. Cela passe par deux aspects clés : la simplicité et la flexibilité.

Simplicité

Cet aspect de la conception d'une API dépend du contexte, car la complexité varie selon les cas d'utilisation. Il est donc important de trouver un juste équilibre au niveau des méthodes de l'API. Pour ce faire, envisagez la simplicité à différents niveaux :

1. **Format des données** : prise en charge XML, JSON, de formats propriétaires ou d'une combinaison de ces formats.
2. **Structure de la méthode** : les méthodes peuvent être génériques ou très spécifiques, et sont souvent appelées selon une certaine séquence pour répondre à des cas d'utilisation particuliers.
3. **Modèle de données** : le modèle de données visible peut différer du modèle sous-jacent. Ce choix a un impact sur la facilité d'utilisation et de maintenance.
4. **Authentification** : les différents mécanismes d'authentification ont chacun leurs forces et leurs faiblesses. Faites votre choix en fonction du contexte.
5. **Politiques d'utilisation** : les autorisations et les quotas imposés aux équipes de développement doivent être faciles à comprendre et à gérer.

Flexibilité

Il est impératif de trouver un équilibre entre simplicité et flexibilité. Une API trop simpliste risque de ne convenir que pour des cas d'utilisation très spécifiques, ce qui limitera son adaptabilité. Pour lui donner une certaine flexibilité, définissez l'espace des opérations potentiel, y compris les systèmes et modèles de données sous-jacents, puis déterminez quel sous-ensemble de ces opérations est réalisable et utile. Voici comment trouver le juste équilibre entre simplicité et flexibilité :

1. **Identifiez les opérations atomiques** : combinez différentes opérations atomiques pour couvrir tout l'espace des opérations.
2. **Identifiez les cas d'utilisation les plus répandus et utiles** : créez une deuxième couche de méta-opérations qui combine différentes opérations atomiques pour répondre à ces cas d'utilisation.

Aspects à prendre en compte pour concevoir le protocole d'API

Bien que le type d'architecture REST (Representational State Transfer) demeure dominant pour le développement d'API, les protocoles d'API deviennent de plus en plus variés. Si d'autres concepts ont émergé récemment, comme les API de streaming ou WebSocket, les API de service web classiques vont continuer d'exister. Idéalement, le principe consiste à choisir le protocole le plus pertinent pour l'utilisateur, puis à trouver le bon équilibre entre simplicité et flexibilité. L'infrastructure permettant de distribuer des API REST continue d'évoluer.

Pour concevoir votre API, tenez compte des aspects suivants :

1. **Alignement avec les cas d'utilisation** : adaptez votre API aux cas d'utilisation identifiés, en gardant suffisamment de flexibilité pour des projets novateurs et plus rares.
2. **Pertinence d'une API REST** : si ce type d'API est à la pointe de la technologie, déterminez s'il répond réellement à vos besoins. Des styles d'architecture différents peuvent davantage convenir à des cas d'utilisation spécifiques.
3. **Modèle de données séparé** : ajoutez une couche d'abstraction entre l'API et le modèle de données afin d'éviter les accès directs à la base de données, sauf si nécessaire.
4. **Paramètres géographiques** : analysez les aspects non fonctionnels tels que la latence et la disponibilité en plaçant les datacenters de manière stratégique à proximité des régions d'utilisation principales.
5. **Intégration de produits** : il est important d'harmoniser l'API avec les autres produits à travers un travail de coordination ou de dissociation, et d'indiquer clairement ce choix en interne comme en externe.

Infrastructure de mise en œuvre et déploiement des API

À mesure que les entreprises migrent vers des infrastructures cloud ou de cloud hybride pour les données et les applications, les infrastructures de mise en œuvre et déploiement des API se diversifient également. Il est possible de concevoir une API en utilisant des microservices et de la combiner à d'autres API basées sur des technologies différentes, comme des monolithes. Les API peuvent être déployées dans des conteneurs, des machines virtuelles (VM), des systèmes bare metal ou dans un cloud public.

Comme avec toute application, il est essentiel de disposer d'un pipeline automatisé d'intégration et de distribution continues (CI/CD) pour la gestion du cycle de vie de l'API. L'[adoption de GitOps](#) et d'Argo CD permet de bénéficier d'une gestion centralisée des configurations d'API et de déploiements continus automatisés, en plus de renforcer la collaboration au sein de l'équipe API, ce qui accélère le développement d'une API plus efficace.

Meilleure pratique n° 4 : Faire de l'exploitation des API une priorité

Une fois l'API déployée, l'équipe de plateforme d'API assure son accessibilité et son adéquation avec les attentes des équipes de développement. Face à l'éventail de solutions sur le marché, le choix d'une stratégie adaptée conditionne la réussite du projet. La gestion des API vise principalement deux objectifs :

1. Rationaliser les processus internes pour les rendre plus efficaces et réduire les coûts
2. Optimiser l'exploitation pour répondre aux attentes des équipes de développement externes

L'article de blog [Building great APIs, part I: The gold standard](#) et le schéma explicatif de l'exploitation des API (Figure 3) peuvent vous aider à atteindre ces objectifs.

Schéma explicatif de l'exploitation des API

Ce schéma peut être utilisé pour définir des méthodes d'exploitation afin de mener à bien la stratégie d'API d'une entreprise. À l'intérieur du cercle se trouvent les activités d'une entreprise et leurs effets internes, tandis que les éléments extérieurs au cercle représentent les effets externes.

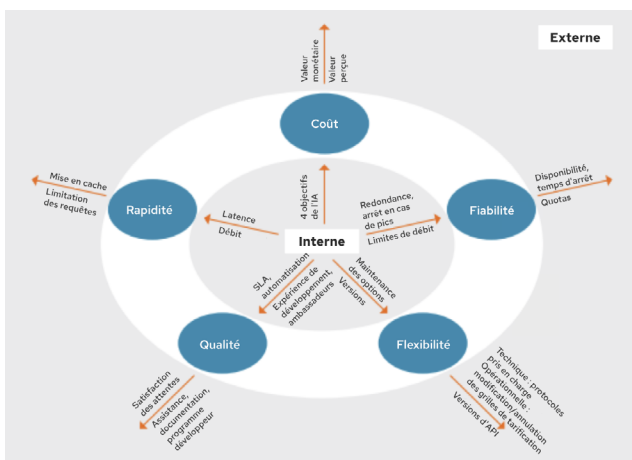


Figure 3 : schéma explicatif de l'exploitation des API

1. **Fiabilité** : veillez à la disponibilité de l'API grâce à la redondance ou aux quotas, ainsi qu'aux limites de débit. Ces aspects sont liés aux modèles économiques et évitent les temps d'arrêt.
2. **Flexibilité** : proposez des options d'adoption techniques et métier (modification des grilles de tarification ou annulations, par exemple). Gardez à l'esprit qu'une plus grande flexibilité peut entraîner une augmentation des efforts internes et des coûts.
3. **Qualité** : répondez de manière durable aux attentes des équipes de développement à l'aide de contrats de niveau de service (SLA) et de processus rationalisés.
4. **Rapidité** : atteignez un débit élevé et une latence faible grâce à des techniques telles que la limitation des requêtes ou la mise en cache.
5. **Coût** : optimisez la valeur pour les équipes de développement tout en réduisant les coûts internes, sans compromettre la qualité.

Plusieurs fournisseurs, dont Red Hat, proposent une infrastructure technique capable de relever nombre de ces difficultés liées à l'exploitation. Le fait de recourir à un fournisseur constitue souvent une solution économique. Cependant, cette démarche exige une stratégie clairement définie.

Composants essentiels de la gestion des API

La gestion d'un écosystème d'API requiert un ensemble unique de composants. Si ces composants varient en fonction de la stratégie d'API, trois d'entre eux s'imposent dans tous les cas :

1. **Contrôle des accès** : mettez en œuvre des systèmes d'authentification et d'autorisation pour permettre les accès et identifier le trafic entrant.
2. **Limites de débit et politiques d'utilisation** : imposez des quotas et des restrictions sur le trafic afin de prévoir la charge.
3. **Analyses** : capturez et analysez les modèles de trafic pour surveiller l'utilisation de l'API.

Aspects à prendre en compte pour l'exploitation des API

Il est utile de veiller à ce que votre stratégie d'exploitation des API soit en phase avec les stratégies globales de votre entreprise afin d'allouer plus facilement des ressources en fonction de l'importance de l'API.

Pour planifier l'exploitation de votre API, tenez compte des aspects suivants :

1. **Contrôle des accès** : déterminez qui dispose d'un accès, effectuez des actions et appliquez des limites pour garantir une utilisation sécurisée de l'API.
2. **Indicateurs de mesure et alerte** : gagnez en visibilité et optimisez les performances de l'API grâce aux analyses, à la mesure d'indicateurs personnalisés et à la mise en place d'alertes.
3. **Gestion des pics** : planifiez l'infrastructure à l'aide du contrôle des accès et des politiques, et mettez en place des mécanismes de basculement tels que l'arrêt en cas de pics ou la limitation des requêtes.
4. **Responsabilité relative à la disponibilité de l'API** : une identification claire de la responsabilité concernant la disponibilité de l'API est essentielle, car elle est directement liée à la génération et à la capture de valeur.
5. **Gestion des utilisations indésirables** : faites la distinction entre les utilisations souhaitées et les utilisations indésirables non souhaitées par le biais de l'exploitation proactive et des conditions générales.
6. **Communication** : établissez un plan de communication clair pour informer les équipes de développement internes et externes des temps d'arrêt, des interventions de maintenance et des modifications prévus en lien avec l'API.

Meilleure pratique n° 5 : Soigner l'expérience de développement

Si l'expérience de développement semble faire référence à la conception des API, il s'agit en réalité d'un concept beaucoup plus large qui englobe la mise en paquet et la distribution de l'API. Même parfaitement conçue, une API REST ou conteneurisée offrira une mauvaise expérience de développement s'il est difficile d'en obtenir un accès, de consulter sa documentation ou de réaliser des tests. Ces aspects peuvent donc nuire grandement à son exploitation réussie.

Une conception de qualité ne suffit pas pour correctement exploiter l'API

Une API simple et flexible a peu d'intérêt si elle n'est pas adoptée par les équipes de développement. De même, une API bien conçue a une influence majeure sur l'expérience de développement et profite d'une meilleure adoption. Or, l'adoption fait partie intégrante de cette même expérience.

John Musser, l'un des pionniers dans le domaine des API et de leur gestion, avance plusieurs solutions, toujours d'actualité, pour renforcer l'adoption :

1. Définir clairement l'objectif de l'API
2. Proposer une inscription facile et un accès gratuit
3. Communiquer de manière transparente sur les tarifs
4. Fournir une documentation complète

Le TTFHW (Time To First Hello World, ou délai avant le premier Hello World) est un indicateur clé pour mesurer l'interaction initiale avec les API, et ainsi améliorer leur conception et faciliter leur adoption. Le TTFPA (Time To First Profitable App, ou délai avant la première application rentable) permet de capturer un contexte plus large et met l'accent sur les programmes de développement. Cet indicateur est moins évident, car la définition de la rentabilité peut varier en fonction de votre API et de votre stratégie d'entreprise. Il est utile d'en tenir compte parce qu'il vous oblige à réfléchir à l'exploitation de l'API dans le cadre du programme d'API.

L'expérience de développement repose sur deux principes : la conception d'un produit ou service générateur de valeur et la facilité d'accès. Un programme efficace d'engagement des développeurs, incluant un portail, une communauté, des ambassadeurs, des événements et des mesures, permet d'optimiser l'engagement. Le portail développeur offre notamment des indicateurs de mesure tels que le nombre de visites de pages, le nombre d'inscriptions, le trafic API ou le nombre de demandes d'assistance. Les événements peuvent être mesurés par le nombre de participants, l'adoption de l'API lors de hackathons ou le nombre de prospects. Il est utile de créer des corrélations, pour savoir par exemple si une conversation lors d'un événement a généré davantage d'inscriptions pour l'API.

Un programme de développement doit inclure les éléments suivants :

- **Création d'une communauté** : des événements tels que les hackathons favorisent les interactions et promeuvent l'adoption.
- **Ambassadeurs de développement** : ils suscitent de l'engagement et mettent en avant les avantages des API, ce qui contribue grandement à la réussite de leur exploitation.
- **Premiers partenaires et études de cas** : les personnes qui adoptent en premier les API formulent des retours et créent des études de cas.
- **Partenaires de l'écosystème** : les partenariats amplifient l'adoption grâce à un travail conjoint.
- **Mesures** : les indicateurs sont en phase avec les objectifs des API et optimisent leur gestion.
- **Communication** : portails, newsletters, et serveurs Slack ou Discord privés.

La création d'un programme de développement adapté à l'audience cible améliore l'engagement et l'expérience.

Aspects à prendre en compte pour évaluer l'expérience de développement

Le fait de proposer une expérience réfléchie peut aider les équipes de développement à maximiser leur potentiel. Voici six critères essentiels qui permettent d'évaluer l'expérience de développement dans l'optique de créer des API qui libèrent la créativité et la productivité.

1. **Explication de la valeur** : élaborez un court argumentaire pour présenter la valeur de l'API aux équipes de développement.
2. **TTFHW et TTFPA** : évaluez ces deux indicateurs de mesure en tenant compte de tous les éléments relatifs à l'expérience de développement tels que les portails, et réduisez-les au maximum.
3. **Processus d'intégration** : gardez à l'esprit les cas d'utilisation de l'API pour préserver la simplicité et favoriser la réussite du développement.
4. **Provisionnement de valeur** : veillez à ce que l'API génère suffisamment de valeur pour attirer et retenir les développeurs.
5. **Assistance au développement** : privilégiez les ressources en libre-service (documentation, FAQ et forums), et proposez des mécanismes additionnels pour traiter les problèmes plus complexes.
6. **Utilisation inhabituelle** : proposez une aide et de la documentation aux équipes de développement faisant face à des cas d'utilisation non standard.

Meilleure pratique n° 6 : Définir une stratégie marketing efficace

Le déploiement d'une stratégie marketing en matière d'API auprès des équipes de développement peut être difficile, notamment si la valeur présentée ne correspond pas aux besoins métier ou techniques desdites équipes. Comme n'importe quel autre produit, les API doivent faire l'objet d'une promotion, et elles doivent répondre aux critères de segmentation, ciblage et positionnement (SCP). Une stratégie marketing efficace consiste à présenter l'API adaptée aux équipes pertinentes, en s'appuyant sur la valeur créée et les principes SCP.

1. **Segmentation** : classez les clients par catégorie (utilisateurs internes, partenaires, utilisateurs finaux et développeurs externes). Pour un meilleur engagement, segmentez l'audience des développeurs, particulièrement vaste, en fonction des tâches qu'ils ont à accomplir (méthode « Job-To-Be-Done »).
2. **Ciblage** : évaluez l'attractivité de chaque segment sur des critères d'accessibilité, de substantialité et de différenciation. Sélectionnez les segments prometteurs et adaptez vos stratégies marketing en conséquence.
3. **Positionnement** : démarquez-vous avec une API qui répond à des besoins spécifiques, résout des difficultés particulières et apporte des avantages pour les segments de développeurs retenus.

Il est impératif d'axer les efforts sur la promotion de l'API. Des stratégies telles que les ambassadeurs de développement, les portails développeur, les hackathons et autres événements peuvent permettre de nouer des relations solides avec les équipes de développement.

Aspects à prendre en compte pour le marketing des API

1. **Audience cible** : déroulez chaque étape du processus en donnant la priorité aux groupes clés (utilisateurs internes, partenaires, clients ou utilisateurs finaux).
2. **Sélection de spécialistes** : pour une promotion efficace, choisissez des spécialistes en phase avec la proposition de valeur de votre API, et tenez compte des aspects comme l'ingénierie, l'assistance, les ventes et la gestion produit.
3. **Stratégie d'événements** : sélectionnez les types d'événements (horizontaux ou verticaux, mondiaux ou locaux, sous forme de conférence ou de hackathon) adaptés aux objectifs de l'API.
4. **Pertinence des hackathons** : déterminez si des hackathons seraient adaptés (thématiques possibles : inscriptions, téléchargements de SDK, applications, recrutement, marque) et planifiez-les en conséquence.
5. **Marketing interne** : garantisiez une assistance au sein des départements, apportez des clarifications avec le service marketing et communiquez à propos des avantages auprès des équipes produit et des clients.

Meilleure pratique n° 7 : Gérer les changements et le retrait des API

En général, les conseils liés aux API visent les phases de conception, de création et d'exploitation. C'est ainsi qu'on néglige une étape primordiale du cycle de vie de l'API, qui intervient de nombreux mois après le lancement et l'exploitation, à savoir la gestion des mises à jour de l'API et son retrait.

Les perturbations liées à des changements brutaux peuvent nuire à la confiance et engendrer des coûts importants, notamment dans le cas de développeurs peu connus, d'approbations d'applications mobiles ou d'appareils incapables de procéder à des mises à jour.

Les changements apportés aux API sont généralement classés comme étant mineurs ou majeurs. Les changements mineurs, liés aux numéros de versions mineurs, incluent les nouvelles méthodes et améliorations. Les changements majeurs, liés aux numéros de versions majeurs et aux migrations, impliquent une suppression, des modifications ou un retrait total. Dans la mesure où il peut être difficile d'empêcher des perturbations au niveau de certaines applications, nous recommandons vivement d'appréhender comme suit chaque changement, même mineur :

- Fournir un point de terminaison de test avec la nouvelle version avant son lancement
- Informer les équipes de développement du changement à venir, avec le calendrier de déploiement et tous les détails, par e-mail ou un autre canal de communication

Une communication efficace et des contrats transparents sont primordiaux. Expliquez les problèmes en détail, respectez vos engagements et présentez la durée de prise en charge des versions. Par exemple, le retrait d'une API nécessite une approche structurée, qui comprend une notification préalable, la gestion de la couverture média, des plans de migration et des outils d'export de données, le cas échéant.

Le plan de migration, qui facilite les mises à jour des API, se décompose comme suit :

1. Présentation d'une nouvelle version en phase de test
2. Notification du retrait de l'ancienne version
3. Assistance pendant la transition
4. Retrait définitif de l'ancienne version

Le processus complet de gestion des API implique d'anticiper les mises à jour et les retraits définitifs, de communiquer de manière efficace et de préserver la confiance en faisant preuve de transparence.

Aspects à prendre en compte pour augmenter la longévité des API

- 1. Engagement envers votre garantie de service :** il s'agit probablement du point le plus important pour votre programme d'API : quel niveau de stabilité de service vous engagez-vous à proposer à vos clients ? Définissez cet engagement avec soin, car il a une incidence sur l'adoption de l'API.
- 2. Processus de changements mineurs et majeurs :** définissez les procédures liées aux versions pour tenir votre engagement. Identifiez les parties concernées ainsi que les étapes d'approbation.
- 3. Signalement des changements :** détectez, documentez et déployez les changements à l'aide de formats de définition des API. Garantissez la compatibilité et veillez à une communication claire.
- 4. Gestion des versions :** surveillez l'utilisation des anciennes versions à l'aide d'identifiants spécifiques pour les développeurs et les utilisateurs. Élaborez un processus de retrait définitif pour éviter tout problème éventuel.
- 5. Alignement produit :** coordonnez l'évolution de l'API avec les changements de produit associés, en tenant compte des engagements pris envers les clients et en gérant toutes les adaptations nécessaires.

Pérenniser votre stratégie d'API

Les meilleures pratiques décrites ici sont conçues pour vous aider à définir, mettre en œuvre et améliorer votre stratégie d'API. L'objectif est de vous permettre d'adapter votre stratégie et de générer de nouvelles opportunités en matière d'API. Vous devrez certainement répondre de manière précise à de nombreuses questions posées dans les sections précédentes, et de nouveaux risques ou opportunités sont susceptibles de survenir avec le temps. Par exemple :

- Existe-t-il des moyens d'augmenter la valeur de l'API pour le client ?
- Le provisionnement est-il suffisant pour l'avenir ?
- La proposition de valeur est-elle suffisamment intéressante pour susciter l'engagement des équipes de développement ?

Votre API et les offres associées doivent évoluer en même temps que le paysage informatique. Le changement est notamment favorisé par les quatre facteurs suivants :

1. **Forces du secteur :** de nouveaux concurrents ou services peuvent remplacer votre API.
2. **Forces du marché :** les exigences des utilisateurs ou les conditions du segment de marché sont susceptibles d'évoluer.
3. **Forces macro-économiques :** les transformations du marché mondial influent sur les budgets des utilisateurs.
4. **Tendances :** les technologies et les exigences réglementaires subissent des évolutions.

Créer un programme d'API durable

En adaptant le programme d'API à ses principaux cas d'utilisation et à ses clients potentiels, l'entreprise est en mesure de le lancer avec efficacité, de le développer et de le faire évoluer. Un tel programme, basé sur la flexibilité des API et la gestion proactive des comportements négatifs à travers la conception et l'exploitation, libère l'innovation. Par ailleurs, les conditions générales liées à l'API doivent permettre à l'entreprise d'agir contre tout comportement inattendu.

La stratégie d'API sera à réévaluer si :

- l'innovation fortuite nuit à la valeur proposée, car une infrastructure robuste est impérative pour mener à bien cette approche ;
- l'absence de cas d'utilisation concrets pour vous et vos utilisateurs indiquent la nécessité d'adopter une autre approche ;
- des doutes subsistent en interne concernant des comportements négatifs, ce qui indique que les mesures prises ou la communication sont insuffisantes ;
- la fréquence des compromissions ou des mauvaises utilisations de l'API reflète un décalage entre la valeur attendue et la valeur réelle.

Les API ont besoin d'une base solide

Le paysage technologique, en évolution constante, a connu l'arrivée de puissants outils de développement d'applications cloud-native, parmi lesquels Kubernetes, Red Hat® OpenShift® et Red Hat OpenShift Service Mesh, qui s'associent pour optimiser la connectivité et accélérer le développement d'applications de qualité. En dépit de ces avancées, les principes de base d'une stratégie d'API robuste et la nécessité d'un système de gestion d'API efficace demeurent. Dans le cadre de votre parcours, choisissez une solution de gestion des API qui soit en phase avec votre stratégie globale et qui démontre votre compréhension de ces plateformes ou votre intégration cohérente avec ces dernières. Vous pourrez ainsi faciliter la mise en œuvre de votre vision stratégique.

Red Hat, un partenaire de choix pour la création et la gestion des API

Chez Red Hat, nous préconisons une approche orientée conception afin de garantir la réussite des programmes d'API. Cette approche englobe l'intégralité du cycle de vie de l'API, de la planification initiale à la conception, qui comprend la mise en correspondance des ressources et la modélisation des scénarios métier, en passant par la gestion des API qui inclut le contrôle des accès, l'accès aux API et l'analyse de l'utilisation.

Pour aider les équipes de développement à appliquer les meilleures pratiques de l'approche orientée conception décrite dans notre manuel d'utilisation des API, nous proposons la solution [Red Hat 3scale API Management](#) spécialement conçue pour la gestion des API dans les environnements de cloud hybride. Favorisant une approche collaborative axée sur la sécurité, cette solution distribuée, légère, cloud-native et basée sur des conteneurs prend en charge des charges de travail volumineuses. Grâce au partage et au contrôle des accès aux services, aux ressources, aux applications et aux systèmes au sein des environnements de cloud public et privé, votre entreprise est à même d'évoluer dans des écosystèmes complexes.

Comme nous l'avons déjà mentionné, les API ont toutefois besoin d'une base solide. La mise en œuvre de ces meilleures pratiques vous imposera de disposer d'une plateforme complète pour le développement d'applications cloud-native. La solution Red Hat 3scale API Management est incluse dans la gamme de produits [Red Hat Application Foundations](#), qui comprend des outils middleware avec des capacités d'intégration, de conception d'API, de gouvernance et gestion des registres d'API, ainsi que de streaming et communication compatibles avec [Red Hat OpenShift Container Platform](#). Grâce à l'association de Red Hat OpenShift et Red Hat Application Foundations, les entreprises disposent d'une plateforme d'applications cloud-native complète pour proposer de nouveaux logiciels de manière plus efficace et sécurisée, tout en générant des possibilités stratégiques et des avantages pour leur activité.

Renforcez votre approche en matière de développement d'applications cloud-native avec Red Hat 3scale API Management, Red Hat Application Foundations et Red Hat OpenShift, et faites l'expérience d'une conception rationalisée, efficace et collaborative des applications.

Conclusion

Nous espérons que ces meilleures pratiques vous aideront à guider vos choix ou à répondre aux questions que vous vous posez. La première étape de la définition de votre stratégie d'API consiste à développer une compréhension claire et commune de la valeur de votre API à l'échelle de l'entreprise. Néanmoins, cette valeur ainsi que les objectifs métier de votre API ne sont pas immuables. Votre stratégie doit donc s'adapter aux différentes évolutions.

D'après notre expérience auprès de nos clients et des observations que nous avons réalisées au cœur de l'économie des API, voici les principaux aspects d'un programme d'API générateur de valeur :

- Les utilisateurs adoptent une API, car elle leur apporte de la valeur. Elle résout des problèmes et offre des avantages.
- Une API génère en permanence de la valeur pour les utilisateurs ; la proposition de valeur et la stratégie associées évoluent en fonction de l'environnement.
- Une API est source de valeur au sein de l'entreprise. D'utilité majeure, elle contribue à la réalisation d'objectifs importants.
- La majorité, voire l'intégralité des parties prenantes sont satisfaites du programme.

Découvrez [Red Hat 3scale API Management](#) et explorez nos [ressources sur la gestion des API](#).