LA GESTION DE VOS API EN 5 ETAPES.

RÉALISER SON PLEIN POTENTIEL AVEC LOBSTER.

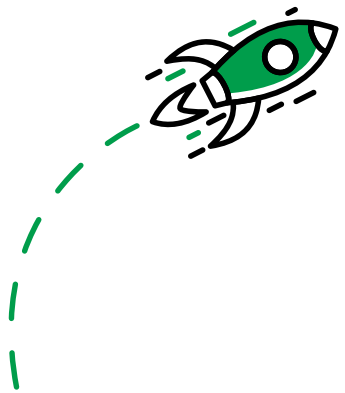
Lobster

Lobster

SOMMAIRE

LES API. PRINCIPES DE BASE.

CE QUE VOUS DEVEZ SAVOIR POUR RÉUSSIR VOTRE PLANIFICATION.



Les écosystèmes d'API apportent des atouts considérables aux entreprises : gain d'efficacité grâce à l'échange électronique de données, monétisation des données et développement de nouveaux modèles commerciaux. Pour réussir sa stratégie API, il faut disposer d'une bonne connaissance des options existantes, des bases techniques et de la gestion des API. Pour chaque étape, Lobster vous propose les bons outils pour prendre votre envol numérique.

API

API signifie Application Programming Interface. Une API est une boîte à outils offrant aux développeurs des commandes standard prédéfinies pour l'implémentation d'applications logicielles. Elle leur évite d'avoir à produire du code à partir de zéro. Il s'agit donc d'un point d'ancrage standardisé et public, par lequel des informations sont récupérées depuis d'autres applications ou leur sont transmises.

Publication d'API

La publication d'API (« API publishing ») désigne la mise à disposition publique de systèmes de données auparavant privés. Elle peut être interne ou externe à l'entreprise, unilatérale (par exemple dans le cas de requêtes de statut d'un envoi) ou bidirectionnelle (par exemple, envoi d'un itinéraire à un système de navigation et renvoi de la vitesse de déplacement en retour).

Stratégie API

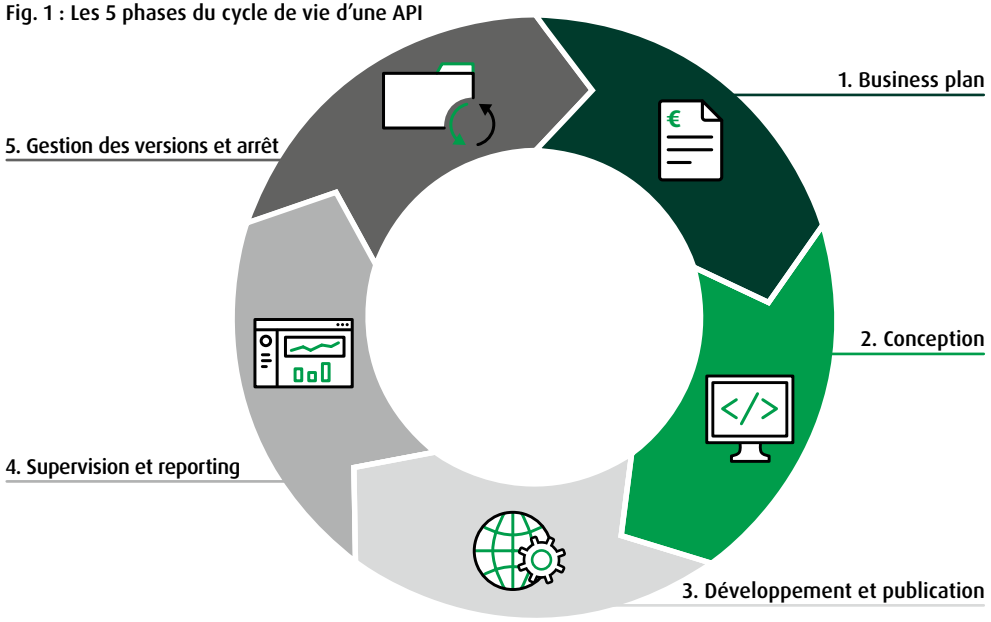
Lorsque des entreprises se donnent mutuellement accès à leurs données via des API, on parle d'écosystème numérique basé API ou d'économie API. En partageant leurs entrées et leurs compétences au sein d'un écosystème API, tous les partenaires renforcent leur croissance et leur capacité d'innovation. La conception planifiée d'un système d'API est appelée stratégie API et sa mise en œuvre est appelée gestion des API (« API management »). Celle-ci a pour objectif le développement, la monétisation, la publication, la documentation, l'analyse, la gestion des versions et la désactivation des API tout au long de leur cycle de vie, c'est pourquoi on utilise ici le terme de gestion du cycle de vie des API.

Lobsterpédia. Le glossaire Lobster des mots-clés pour votre transformation numérique.

L'introduction d'une stratégie API n'est pas seulement pertinente du point de vue technique, mais elle est aussi essentielle du point de vue commercial. Une stratégie API réfléchie et correctement mise en œuvre permet d'une part de réduire les coûts en rationalisant et en automatisant les processus (« API-led integration »), et agit d'autre part comme un moteur de croissance et d'innovation pour tout modèle commercial (« API-led connectivity »), en générant de nouvelles sources de revenu et en augmentant la valeur commerciale dans l'ensemble de l'écosystème.

Réussir la mise en œuvre d'une stratégie API requiert une planification minutieuse. Pour chaque phase de ce processus de mise en œuvre, il est donc important de bien saisir les étapes et les éléments à prendre à compte. Le moyen le plus simple d'y parvenir est d'étudier le cycle de vie d'une API.

Fig. 1 : Les 5 phases du cycle de vie d'une API



5 ÉTAPES POUR VOTRE GESTION API.

ÉTAPE 1 : BUSINESS PLAN.

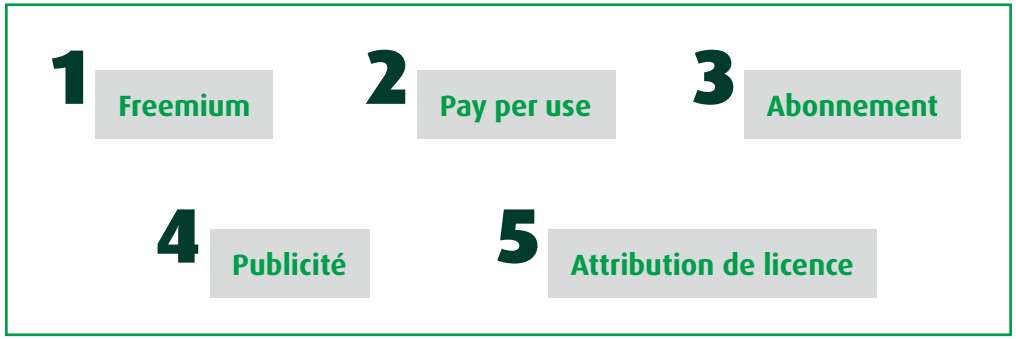


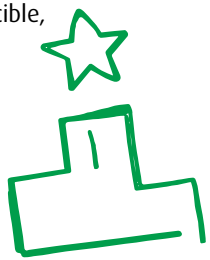
Fig. 2 : Modèles de monétisation des API

Les API servent à intégrer de manière modulaire des données et des capacités internes et externes afin de générer de la valeur ajoutée. La première étape consiste donc à définir les objectifs de la stratégie API afin d'identifier les données les plus importantes ainsi que les groupes-cible pertinents en interne et en externe. Ce processus devrait être porté conjointement par l'équipe métier concerné et le service informatique, pour clarifier ensemble, par exemple, si les API existantes couvrent déjà les exigences nouvellement formulées, afin d'éviter les doublons et ainsi ne pas complexifier inutilement le paysage API.

Pour cela, il est préférable de s'intéresser d'abord aux processus les plus importants et les plus laborieux au sein de l'entreprise, et de se demander en quoi les API peuvent aider à les améliorer. Puis, pour approfondir l'analyse, il faudrait se concentrer sur les parties prenantes : quels besoins en informations ou en compétences pourraient être satisfaits par l'introduction de nouveaux systèmes d'API ? Quelles informations doivent être mises à disposition, à partir de quels systèmes, et sous quelle forme ?

Dans cette première phase, il peut être utile de mener une discussion approfondie et de réaliser une preuve de concept avec quelques grands comptes, car une stratégie API doit être mutuellement bénéfique. Observer les concurrents peut aussi se révéler précieux. Que font-ils pour profiter des avantages d'une stratégie API ?

Enfin, une fois que les cas d'application pertinents ont été trouvés, il est temps de mettre en balance les coûts et les gains de chiffre d'affaires potentiels, afin de passer de la conception sommaire du projet à un business plan. Le choix du modèle de monétisation dépend du groupe cible, de la proposition de valeur et des objectifs commerciaux de l'entreprise. Il faut notamment prévoir tous les coûts inhérents à l'API, comme, par exemple, ceux liés aux exigences en matière de puissance de calcul et de sécurité.



Les modèles courants de monétisation des API sont les suivants :

- **Freemium :** dans ce modèle, l'entreprise met gratuitement à disposition une version de base de son API. Des frais ne sont prélevés que pour l'utilisation de fonctions premium ou en cas d'utilisation intensive. Par exemple, un service météorologique pourrait fournir gratuitement des informations générales de température et d'ensoleillement via l'API, mais facturer des frais pour obtenir des prévisions horaires ou pour accéder à des historiques de données.
- **Pay per Use (paiement à l'utilisation) :** dans ce modèle, une entreprise facture des frais à ses clients pour chaque accès à l'API, ou bien par volume de données (avec ou sans bande passante de volume), avec un compte rechargeable, etc. Une plateforme de commerce électronique peut ainsi demander de l'argent pour chaque recherche de produit ou pour chaque achat effectué via son API. Ce modèle est souvent utilisé lorsque la valeur de l'API est directement liée à son utilisation.
- **Abonnement :** dans ce modèle, l'accès à l'API est soumis à des frais périodiques, souvent échelonnés sur différents niveaux correspondant à des fonctionnalités et des variantes étendues (par exemple basic, pro, expert).
- **Publicité :** dans ce modèle, l'API est proposée gratuitement. Le chiffre d'affaires est généré par la publicité diffusée lors de l'utilisation de l'API ou par l'intégration de plateformes publicitaires et de leurs API. Ce modèle de monétisation est souvent utilisé pour les applications mobiles.
- **Attribution de licences :** dans ce modèle, l'API fait par exemple partie d'un ensemble de produits facturé sous forme de droits de licence ou constitue elle-même un produit pouvant faire l'objet d'une licence.

COMMENT LOBSTER SOUTIENT VOTRE BUSINESS PLAN.

- Accompagnement tout au long du cycle de vie des API, avec un haut niveau de sécurité et de performance.
- Fourniture d'un logiciel tout-en-un, Lobster_data, avec connecteurs et fonctions de conversion de données préprogrammées, de sorte que les clients n'aient besoin que d'une seule plateforme pour les API, l'EDI, l'EAI et l'IoT.
- Grande convivialité grâce à la configuration no-code, sans connaissances en programmation. Permettant aux développeurs professionnels de parvenir plus rapidement à des résultats excellents et aux équipes métier de se convertir en « citizen developers » pour configurer elles-mêmes leurs applications logicielles.
- Outil totalement hybride, à la fois compatible avec le cloud et utilisable sur site ou de manière autonome dans le cloud privé du client. Les appels sortants appartiennent au passé. Lobster_data fonctionne également sur les serveurs, les machines virtuelles ou les conteneurs Docker du client, sur le réseau interne.
- Maîtrise rapide de Lobster_data grâce à une formation de base de 3 jours, complétée par des stages additionnels au sein de la Lobster Academy, en groupes ou sur demande.
- Documentation complète en ligne, assistance interactive pour utilisateurs, mode « Aide » avec vidéos et tutoriels pour tous les modules, outils et fonctions de Lobster_data.

OÙ UTILISER LA SOLUTION DE GESTION API LOBSTER_DATA ?

- Utilisation indépendante du secteur (« industry agnostic ») – de la logistique au textile, en passant par l'e-commerce, et l'agro-alimentaire, par exemple.
- Intégration interne des systèmes ou construction d'écosystèmes basés API regroupant des partenaires externes.
- Conception d'API REST comme des services web, car Lobster_data est utilisable simultanément en tant que serveur et client.
- Connexion de tous les points terminaux et suppression des silos de données grâce à une grande profondeur fonctionnelle.

COMMENT LOBSTER_DATA SOUTIENT LA MONÉTISATION DE VOTRE STRATÉGIE API.

- Transparence et analyse approfondies quant à l'utilisation d'une API grâce au tableau de bord Lobster_data et à des pipelines configurables intégrant les données de rapport.
- Transfert fluide des statistiques d'utilisation à des systèmes tiers pour la mise en œuvre de tous les modèles de monétisation.

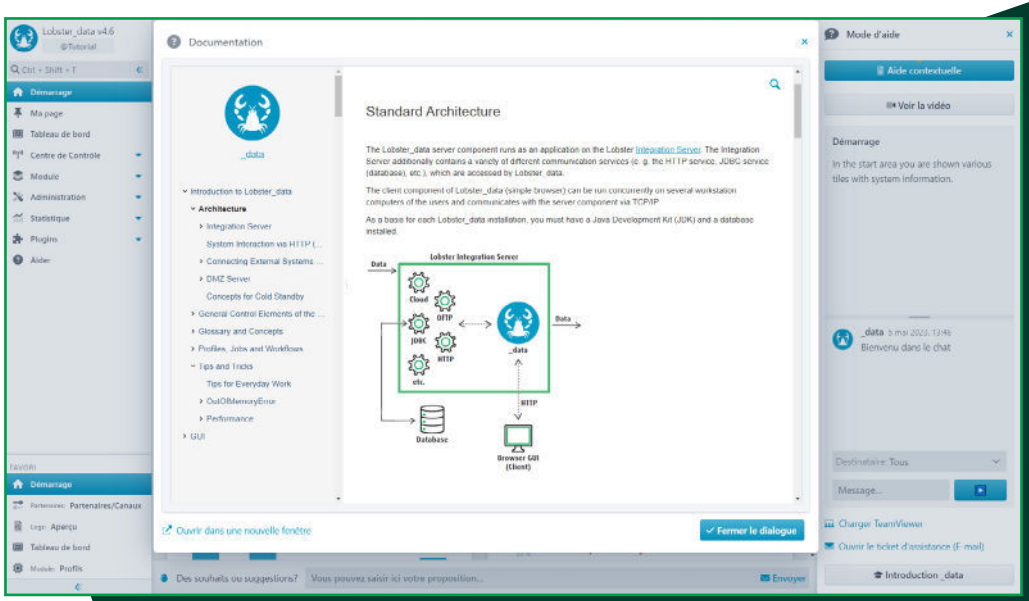


Fig. 3 : Aide Lobster_data

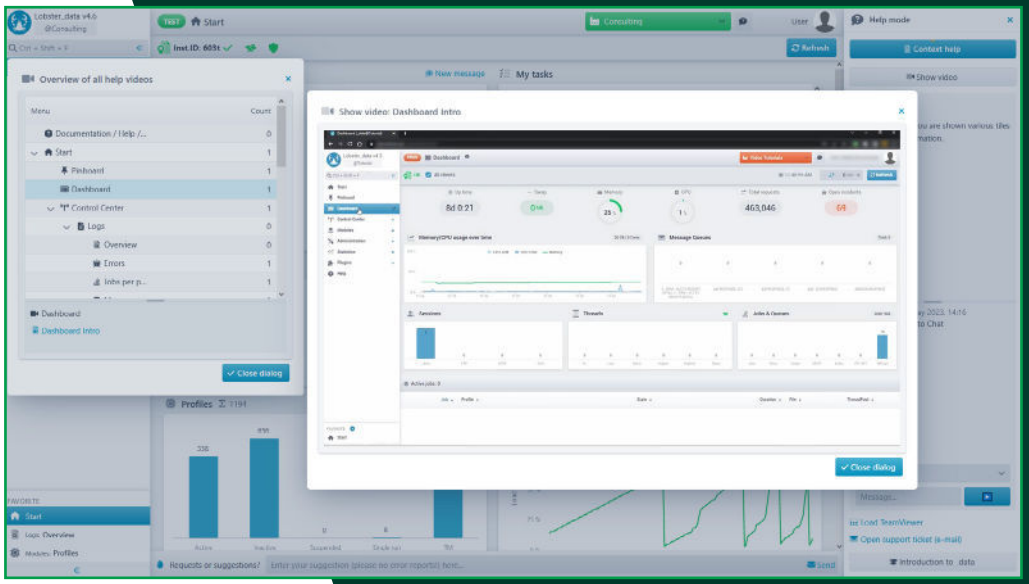
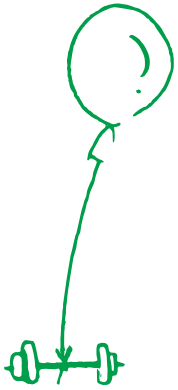


Fig. 4 : Vidéo d'aide Lobster_data

5 ÉTAPES POUR VOTRE GESTION API.

ÉTAPE 2 : CONCEPTION D'UNE API.



La conception d'une API revêt une grande importance pour sa gestion API, car elle détermine fortement la fonctionnalité et la facilité d'utilisation de celle-ci. Une bonne conception est donc le résultat d'une étroite collaboration entre les différents acteurs, tels que le responsable produit/process, le développeur logiciel et autres parties prenantes.

Le processus doit par conséquent mettre l'accent sur l'étendue des fonctionnalités, la convivialité, l'évolutivité/les performances et la sécurité – en plus de la sélection de la pile technologique et de la définition de la structure et du flux de données ainsi que d'une procédure explicite de traitement des erreurs. L'interface utilisateur de l'API doit en outre permettre aux externes de s'intégrer rapidement : la simplicité et la cohérence doivent donc être des principes directeurs de toute conception.

La connexion des systèmes requis est une autre composante de la conception d'une API. Selon la complexité du paysage informatique, il est nécessaire de lier entre eux les anciens et les nouveaux systèmes via différents canaux d'entrée et de sortie. Dans les systèmes hétérogènes, les formats des messages échangés diffèrent souvent fortement les uns des autres, de sorte que la solution la plus simple consiste à utiliser un middleware.

La puissance de l'API est un autre critère essentiel. Les API doivent être conçues de manière à pouvoir traiter efficacement un volume important de requêtes. Pour répondre à cette exigence de performance, il faut disposer d'une infrastructure bien structurée et évolutive, par exemple un fournisseur de cloud. Une telle infrastructure réduit la latence de l'API et augmente sa fiabilité. La mise en cache des données fréquemment utilisées et la mise en œuvre d'une fonction d'équilibrage des charges peuvent en outre contribuer à répartir de manière optimale la charge des requêtes et à réduire les temps d'arrêt. Enfin, l'utilisation de normes d'API telles que http, JSON et REST permet également d'améliorer les performances d'une API.

Peu importe le nombre de modules déjà intégrés pour garantir la haute performance d'une API : même après la mise en service, il faut procéder continuellement à des améliorations, lesquelles doivent être prises en compte dans le coût total de possession (« TCO »). La première des priorités est toutefois que l'API réponde à tout moment aux exigences des clients et génère des recettes.

Enfin, un aspect central d'une API réside dans sa sécurité, car les API constituent un point d'entrée dans les systèmes et les données des entreprises. Les API les plus perfectionnées, en s'appuyant sur l'authentification, l'autorisation, la réglementation et la technologie de la blockchain, sont même en mesure de générer des contrats de manière automatique. Il faut donc prendre en compte que toute panne de données expose tant l'entreprise que ses clients à des dommages considérables, et sécuriser le système de serveur grâce aux mesures suivantes :

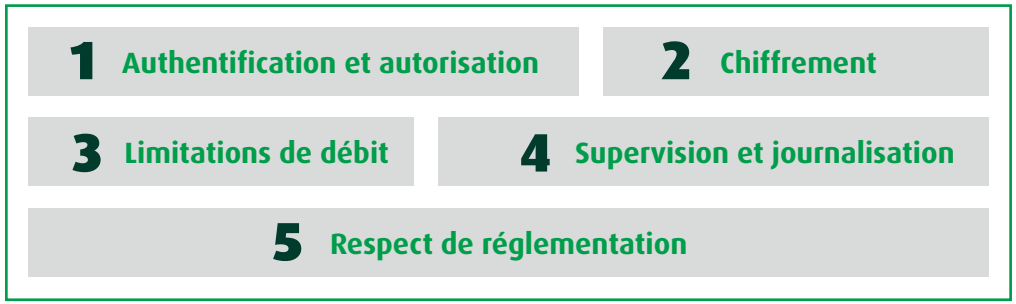


Fig. 5 : Mesures de sécurisation du système de serveur

- Les processus d'authentification et d'autorisation via OAuth et OpenID Connect garantissent que seuls les utilisateurs et les systèmes autorisés aient accès aux API.
- Le chiffrement assure la protection des données, lors de leur transmission et au repos.
- Les limitations de débit empêchent d'éventuels acteurs malveillants de surcharger de requêtes le trafic réseau vers l'API (attaques DDoS).
- La supervision et la journalisation au moyen de routines spécialisées permettent au fournisseur d'observer l'utilisation de l'API, d'identifier d'éventuelles menaces de sécurité et d'intervenir rapidement en cas d'incident.
- Le respect des normes et des réglementations, telles que HIPAA, SOC2 et PCI-DSS, garantit que toutes les exigences de sécurité sont appliquées conformément aux prescriptions légales.

La remarque formulée pour la performance d'une API s'applique également à sa sécurité : celle-ci n'est pas réglée par la mise en place de mesures appropriées, mais doit faire l'objet de contrôles réguliers – lesquels doivent être pris en compte dans l'évaluation du coût d'une stratégie API.

CONCEPTION ET MAQUETTES D'API AVEC LOBSTER_DATA.

- Facilité d'élaboration de systèmes d'API multicouches afin d'intégrer des données en provenance de tous les systèmes internes (EAI), externes (EDI) ou de machines (IoT).
- Véritable environnement mutualisé et prise en charge de clients, pour les clouds publics comme pour les solutions sur site. Nombreuses possibilités de réguler la visibilité et l'accès à l'API.
- Élaboration d'API REST et SOAP pour la réception, l'expédition et la conversion de données.
- Connexion de système entiers, qu'il s'agisse de systèmes nouveaux ou anciens, spécialisés ou généralistes, d'Excel ou d'EDIFACT. Grâce à son « modèle en 6 phases » Lobster_data permet en outre d'accéder à n'importe quel silo de données (canaux d'entrée, analyse, mapping, coopération base de données, fichiers de sortie, canaux de sortie), et de fournir ainsi des API et des pipelines de données structurées de manière identique et nécessitant peu de maintenance. Indépendamment des collaborateurs, du canal d'entrée, du format de données et du canal de sortie.
- Vaste bibliothèque, avec plus de 450 fonctions de calcul permettant de traiter, de consolider et de combiner des données.
- Prise en charge d'appels d'API imbriqués permettant l'appel d'une API par une autre (« API mashup »).
- Routage dynamique et définition de séquences d'interfaces spécifiques en fonction du contexte.
- Définition de schémas d'API et de leur publication via les descriptions d'interface Swagger/OpenAPI pour permettre la documentation et la génération de code et de cas de test automatisée.
- Modélisation graphique de processus complexes avec le Workflow Designer intégré dans Lobster_data.
- Accompagnement de l'ensemble du cycle de vie des API grâce à des extensions de tous niveaux de complexité (validation, enrichissement ou processus de décision avec flux de travail).
- Prise en charge de « Universal HTTP-API Webservice » via des agents d'entrée « http » à la réception d'une requête http. Lorsque le service est publié, Lobster_data génère tous les éléments pour la création d'une interface Web Soap ou REST.

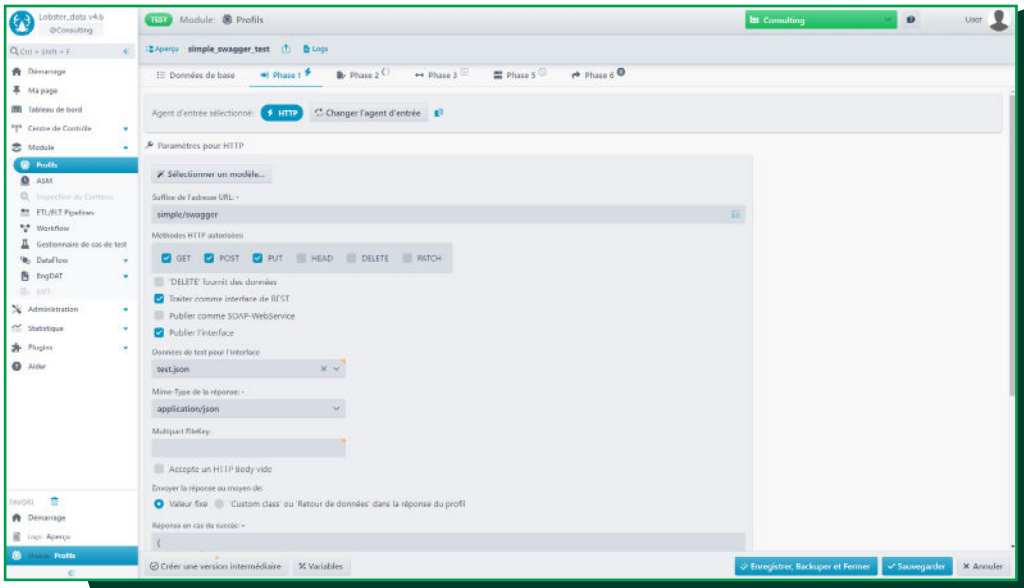


Fig. 6 : Le « modèle en 6 phases » de Lobster pour un accès normalisé à n'importe quel silo de données

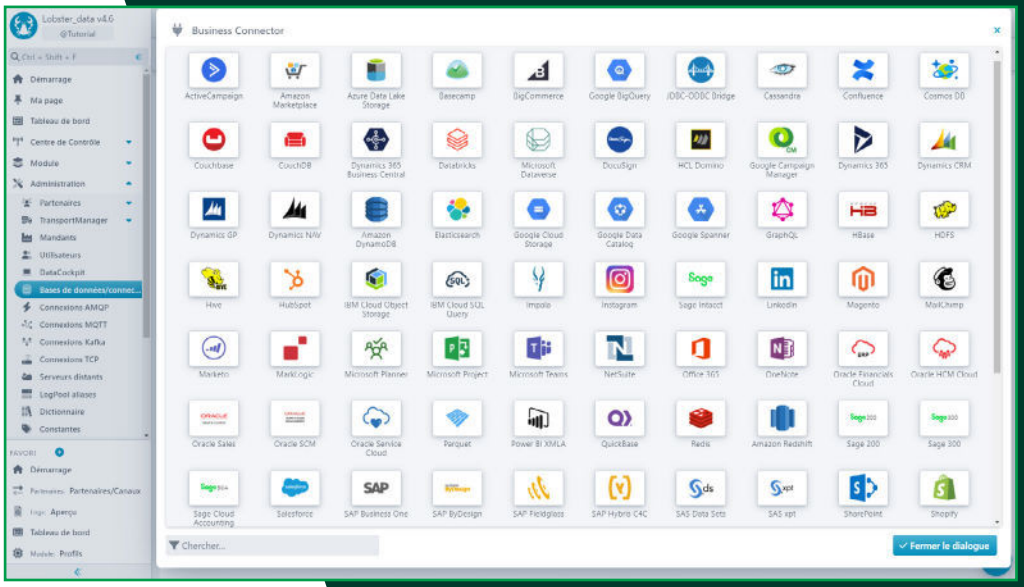


Fig. 7 : Connecteurs préconfigurés pour la connexion de systèmes hérités

CONVIVIALITÉ DES API AVEC LOBSTER_DATA.

- Configuration autonome des interfaces grâce à l'approche no-code et au paramétrage intuitif de clients et de serveurs API REST en quelques clics, sans programmation. Sans connaissances approfondies de HTTP/HTTPS ni des authentifications HTTP.
- Interface utilisateur claire et compacte, avec des tableaux de bord pour toutes les API.
- Capacité multi-utilisateur afin que plusieurs équipes puissent travailler en parallèle de manière indépendante.
- Fonctions de génération Swagger et documentation automatique d'interfaces OpenAPI.

ÉVOLUTIVITÉ ET DYNAMIQUE POUR SATISFAIRE TOUS LES BESOINS DE PERFORMANCE.

- Possibilité de faire évoluer les services à tout moment ; mise en œuvre en libre service, avec l'assistance de nos consultants ou avec prise en charge intégrale par Lobster.
- Déploiement évolutif sur site, dans le cloud Lobster, votre cloud privé ou une solution de cloud hybride.
- Équilibrage des charges et haute disponibilité pour les pics de trafic dynamiques grâce à l'utilisation de plusieurs nœuds de travail sur une infrastructure distribuée régionale (serveur, machine virtuelle, conteneur Docker).
- Application de correctifs et de mises à jour sans temps d'arrêt, par roulement ; mise à jour successive des nœuds de travail pendant le fonctionnement de l'API.
- Gestion efficace de la charge de travail. Compression intelligente par proxy avec gains supplémentaires en cas de bande passante limitée et de temps de latence critique.
- API de streaming via Apache Kafka pour une intégration de données et des pipelines de données en temps réel ultraperformantes.

SÉCURITÉ MAXIMALE.

- Méthodes d'authentification modernes, telles que OAuth 2.0, SSO, 2FA, MFA, LDAP. Connexions HTTP/HTTPS pour la réception comme pour la transmission.
- Qualité attestée par la certification de Lobster conformément aux normes ISO 27001 « Information Security » et ISO 27018 « Data Privacy Cloud Solutions ».
- Gestion centralisée des certificats de tous les partenaires de communication pour la signature et le chiffrement.
- Prise en charge automatique des authentifications HTTP (authentifications de base et Digest Access) par le serveur.
- Importation individuelle de classes Java par l'utilisateur et appel à des ressources complémentaires telles que des bibliothèques de chiffrement, des outils de validation de schémas et des bibliothèques de validation de données pour les exigences de sécurité supérieures.

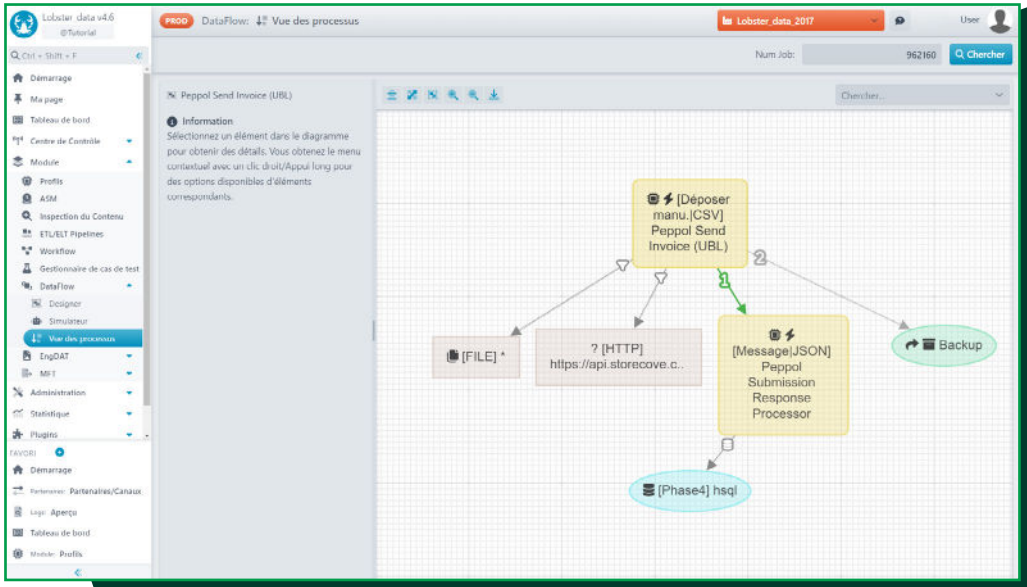


Fig. 8 : Visualisation des API sous forme de diagrammes dans le DataFlow Designer

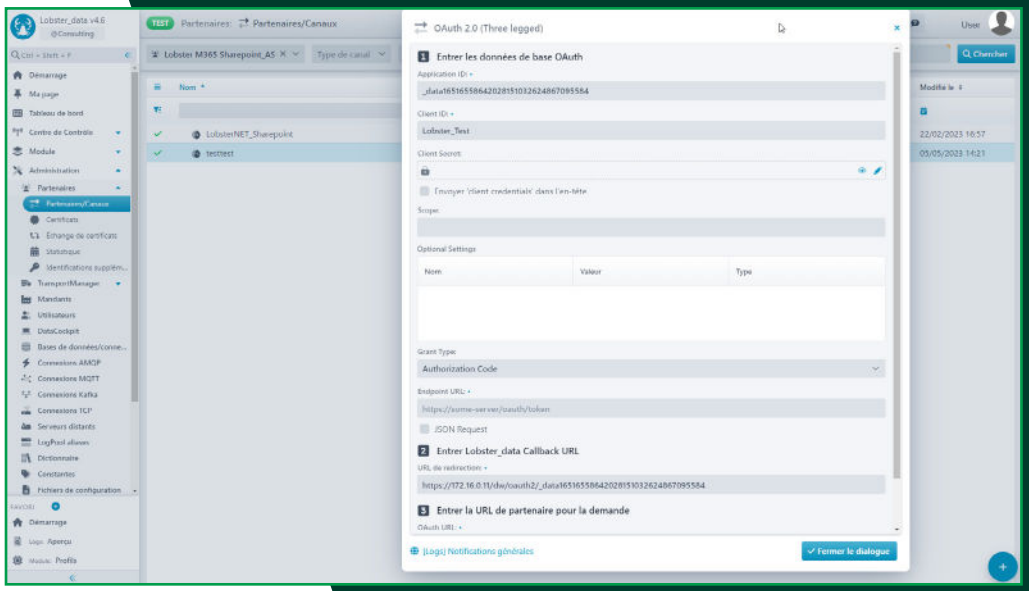


Fig. 9 : Configuration conviviale de tous types d'authentification

5 ÉTAPES POUR VOTRE GESTION API.

ÉTAPE 3 : DÉVELOPPEMENT ET PUBLICATION.

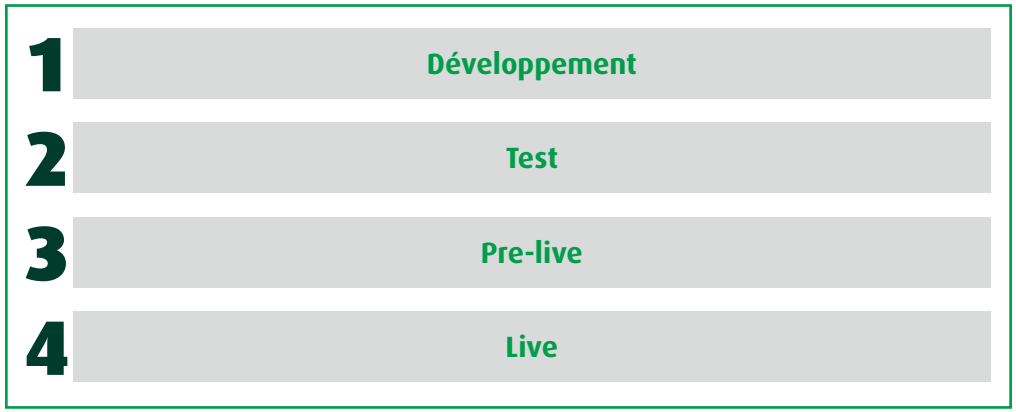


Fig. 10 : Quatre étapes pour le développement et la publication des API

Après la conception vient la phase de développement. L'API doit être créée et soumise à de nombreux tests pratiques et techniques. Elle doit passer par quatre étapes (développement, test, pre-live, live) avant d'être rendue publique. Chacune de ces étapes présente ses propres défis.

Le **développement** met l'accent sur la création et la mise en œuvre de l'API conformément aux spécifications définies lors de la conception. Durant cette phase, il est important de respecter certains principes de composition du code, comme la modularité, la maintenabilité et la réutilisabilité. Les révisions de code et les tests de module offrent une garantie de qualité et de stabilité du code de base.

Un soin particulier doit être apporté à la documentation de l'API pendant cette phase, afin de décrire succinctement le fonctionnement des interactions avec celle-ci. La documentation doit ainsi contenir des exemples de requêtes et de réponses, de codes d'erreur et les instructions générales pour l'utilisation de l'API. En général, deux versions sont produites :

- Pour les développeurs internes, un manuel de référence complet et actuel contenant des détails techniques tels que les points terminaux, les formats des requêtes et des réponses et les protocoles de sécurité.
- Pour les utilisateurs externes, une documentation simplifiée et conviviale, qui présente les avantages et les possibilités de l'API sous une forme abordable.

Une API bien documentée rend non seulement par la suite le développement, la maintenance et la formation interne plus rapides, mais elle améliore aussi l'expérience utilisateur des clients finaux et réduit ainsi le nombre de demandes d'assistance.

Durant la **phase de test**, l'API est soumise à des tests approfondis et systématiques par les développeurs et éventuellement les parties prenantes, afin d'identifier en amont les problèmes de fonctionnement, de performance et de sécurité, d'éliminer les bugs, les problèmes de compatibilité et les failles de sécurité et d'effectuer des tests fonctionnels et non fonctionnels.

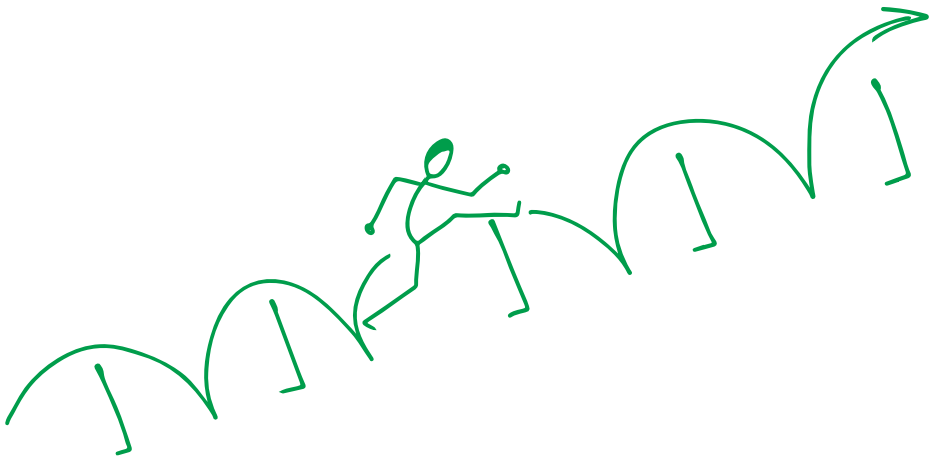
Les tests fonctionnels visent à vérifier que l'API reproduit la logique métier, fournit les résultats attendus et traite les erreurs et les exceptions de manière adéquate. Les tests non fonctionnels contrôlent la performance, l'évolutivité, la sécurité et la fiabilité de l'API. Les critères évalués sont notamment le temps de réponse, la capacité de charge, la protection des données et la récupération après sinistre.

Lorsque la phase de test s'est déroulée avec succès, il convient de réaliser un dernier cycle de validation de l'API, appelé « **pre-live** » : une sorte de répétition générale. Au cours de cette phase, l'API est testée dans un environnement qui ressemble le plus possible à la phase « live » : la communication avec les autres systèmes et services fonctionne-t-elle ? La documentation de l'API est-elle complète ? L'API est-elle conviviale ?

Enfin, durant **la phase « live »,** l'API et ses spécifications (fonctionnalités, paramètres d'appel, fonctions de sécurité, résultats, etc.) sont publiées, et donc rendues accessibles à l'utilisateur final. AAu cours de cette phase de publication, il est important d'observer de près l'utilisation et les performances de l'API, de la mettre à jour et d'assurer une maintenance régulière, afin d'éliminer les défauts, de résoudre les problèmes de sécurité et d'améliorer l'interface si nécessaire. Il convient en outre de recueillir le feed-back des utilisateurs pour être en mesure d'améliorer continuellement l'API au fil du temps et s'assurer qu'elle continue de répondre aux exigences de l'entreprise et des parties prenantes.

DE A JUSQU'À Z. L'ENSEMBLE DU CYCLE DE VIE API AVEC LOBSTER.

- Développement et documentation automatiques des API au moyen de descriptions d'interface OpenAPI (Swagger) que Lobster_data génère automatiquement lorsque les informations requises sont disponibles.
- Établissement rapide de liaisons de communication (REST, AS2, SFTP, SMTP, X.400, etc.) et nombreuses options prédéfinies pour la lecture et la création de différents formats de données (XML à partir de XSD, JSON, EDIFACT, CSV, etc.) grâce au prototypage rapide de maquettes d'API.
- Vérification sans risque de l'API créée en environnement de test et en environnement « live », car toutes les Éditions dans Lobster_data sont livrées avec une installation de test et une installation productive.
- Transfert rapide et exhaustif de toutes les modifications du test vers le « live » avec le gestionnaire de transfert de Lobster_data, qui transmet toutes les données pertinentes du système de test vers le système cible.
- Migration des API vers l'infrastructure de production, avec des fonctions avancées en option comme la haute disponibilité et l'équilibrage des charges. En fonction du comportement de charge de l'API : définition de paramètres d'évolutivité pour une éventuelle intégration de nœuds de travail supplémentaires.



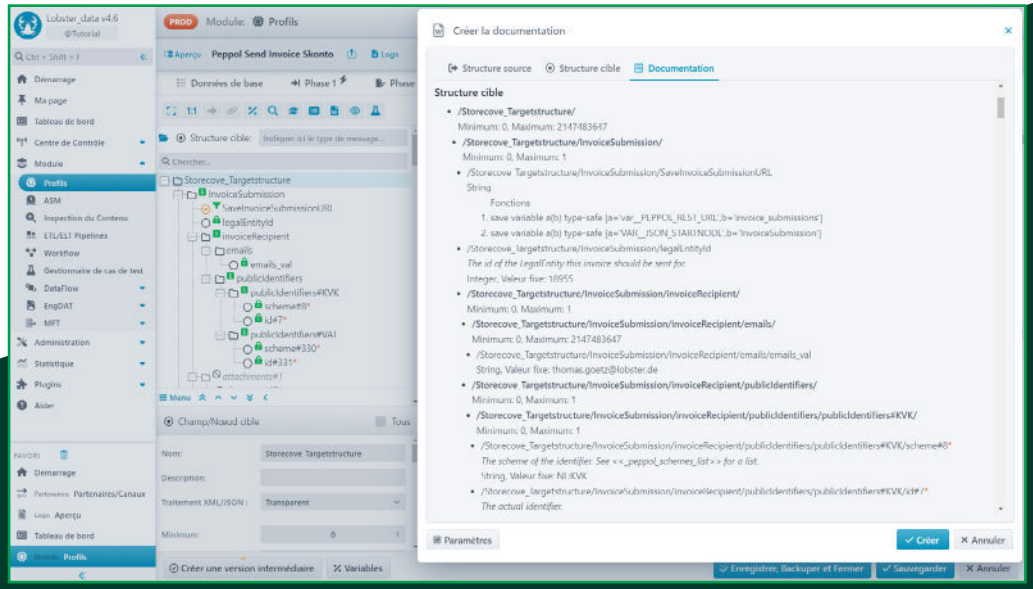


Fig. 11 : Transmission exhaustive des informations grâce à la création automatique de documentation

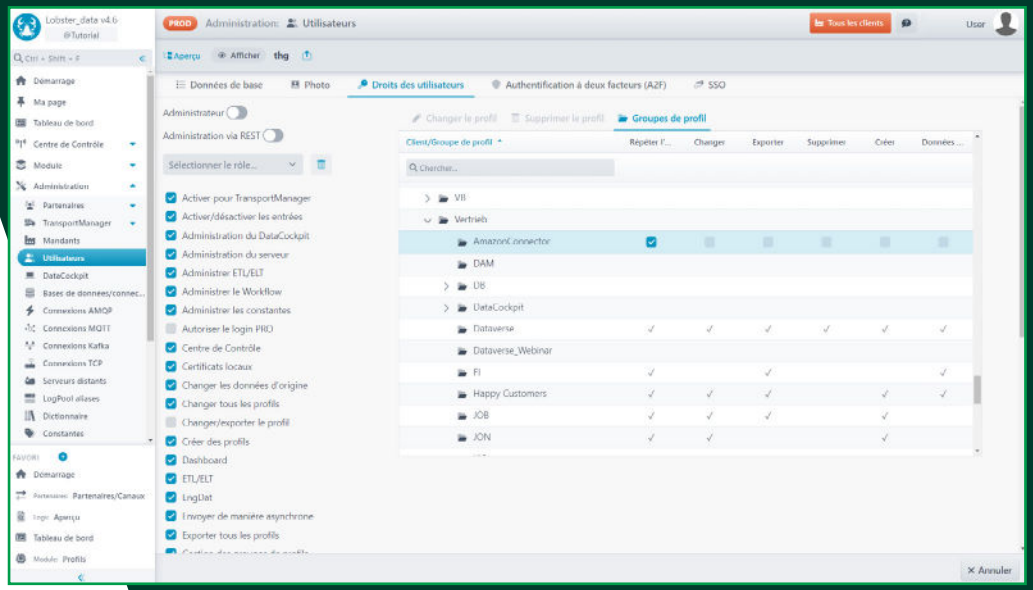


Fig. 12 : Entrée facile dans la phase « live » grâce à une répartition des rôles conviviale

5 ÉTAPES POUR VOTRE GESTION API.

ÉTAPE 4 : SUPERVISION ET REPORTING.

Après la publication d'une API, il est important de superviser ses performances, son utilisation, sa sécurité et son succès commercial. Chaque type de supervision présente des exigences bien spécifiques :

- Superviser les performances de l'API, c'est vérifier la vitesse et l'efficacité avec lesquelles l'API réagit aux requêtes et renvoie des données. Sur ce point, les valeurs à suivre sont avant tout les temps de réponse, le taux d'erreur et la charge du serveur.
- Les performances de l'API dépendent fortement de qui l'utilise, de quelle manière et avec quelle intensité. Il est important d'identifier de tels schémas d'utilisation, car ils peuvent permettre de guider l'attention des managers, de déterminer quelles fonctions de l'API sont les plus utiles pour les différents segments de clientèle, et de réagir de manière appropriée y compris à des pics de requêtes inattendus.
- La sécurité de l'API est garantie par la réalisation régulière de recherches de failles et de tests de pénétration, ainsi que l'analyse des fichiers journaux pour y détecter des activités inhabituelles. Les protocoles de sécurité doivent être en permanence tenus parfaitement à jour.
- Enfin, il est plus que judicieux de ne pas perdre de vue les recettes et les coûts générés par l'API. Cela implique d'une part d'observer l'évolution du nombre d'utilisateurs ainsi que la fréquence des requêtes dans le cadre du modèle de monétisation. Il convient d'autre part d'assurer un suivi des frais courants liés à la prise en charge de l'API, à l'infrastructure sous-jacente et au support technique, afin de disposer d'un tableau clair du coût total de possession (TCO) et du rapport coûts/avantages.

ÉTAPE 5 : GESTION DES VERSIONS ET ARRÊT.

Durant la phase d'exploitation, il est possible de créer de nouvelles versions de l'API, notamment via la modification des informations de sortie ou des formats des requêtes ou des sorties, en passant par exemple d'une valeur entière à une valeur décimale. La nouvelle version de l'API doit, elle aussi, être soigneusement documentée et faire l'objet d'une communication claire ; elle doit typiquement être accessible via une URI modifiée ou un en-tête de requête adapté.

Toute API atteint un jour la fin de son cycle de vie. Encore faut-il trouver le moment adéquat. Trop précoce, et vous perdez des recettes. Trop tardif, et vous supportez des coûts qui n'ont pas lieu d'être. L'annonce de l'arrêt d'une API est un événement déterminant pour la bonne santé de l'ensemble de l'écosystème, et il importe donc de bien le planifier pour limiter les répercussions sur les systèmes connectés à l'API et sur l'expérience utilisateur.

Les caractéristiques suivantes parlent en faveur de l'arrêt d'une API :

- Baisse du taux d'utilisation, pour un coût d'exploitation constant
- Obsolescence des fonctionnalités proposées, avec des effets négatifs sur la performance et la facilité d'utilisation
- Évolution du contexte économique, rendant l'offre de l'API obsolète
- Compromission de la sécurité des entreprises connectées par le système de l'API

Il incombe aux gestionnaires de l'API de suivre attentivement son cycle de vie et de planifier son arrêt avec anticipation. L'arrêt d'une API, au même titre que sa mise en service, doit être soigneusement préparé (quand ?, comment ?, etc.), faire l'objet d'une communication précoce en interne et en externe, ainsi que d'une planification fiable. L'objectif est, là encore, de faire en sorte que la transition soit fluide pour les utilisateurs, d'interrompre le moins possible l'activité et éventuellement de proposer des API de remplacement pour la base d'utilisateurs. Dans ce contexte, il est important d'informer les utilisateurs en temps utile, afin qu'ils puissent de leur côté procéder aux adaptations nécessaires de leurs systèmes. Pour finir, l'équipe compétente doit mettre la documentation à jour en supprimant tous les renvois à l'API mise hors service.

MONITORING DANS LOBSTER_DATA : NE PERDEZ AUCUNE INFORMATION.

- Monitoring complet grâce au tableau de bord, avec des rapports généraux et analytiques.
- Suivi des données de performance en temps réel pour chaque type de message des applications et de l'API.
- Définition du volume et du temps d'exécution maximaux pour les taux d'utilisation de l'API.
- Données statistiques de l'API, calcul et représentation sous forme condensée ou non.
- Outils de debug et envoi d'alertes en cas d'anomalies.
- Détection des attaques par injection de données grâce au scanner antivirus du client.

REPORTING DANS LOBSTER_DATA.

- Fonctions de transparence permettant à toutes les parties prenantes de s'impliquer de manière intuitive.
- Collecte des données asynchrone afin de pas perturber le trafic de données.
- Création de rapports (ad hoc) personnalisés – préconfigurés ou à la demande.
- Exportation facile des données d'analyse dans des systèmes d'entrepôt de données ou de data lakes.
- Analyses de tendances en temps réel par contrôle des profils eux-mêmes, avec le tableau de bord ou la console administrateur.
- Notification en temps réel en cas d'exception (entrées de journal, e-mail, système de ticket).

GESTION DES VERSIONS ET ARRÊT.

- Gestion évolutive du cycle de vie offrant une transparence et une maîtrise parfaites.
- Gestion de version intégrée avec attribution d'un identifiant et d'un numéro de version uniques, création impérative d'une nouvelle version en cas de modifications et de sauvegarde des versions.
- Création automatique d'une documentation Swagger/OpenAPI pour chaque version d'API.
- Gestion des versions et des variantes pour la création, l'importation, l'exportation, la réplication, la désactivation ou la mise à jour d'API.
- Annonce d'arrêt d'API sous forme lisible par une machine et indication d'un en-tête de réponse X-API-Warn.

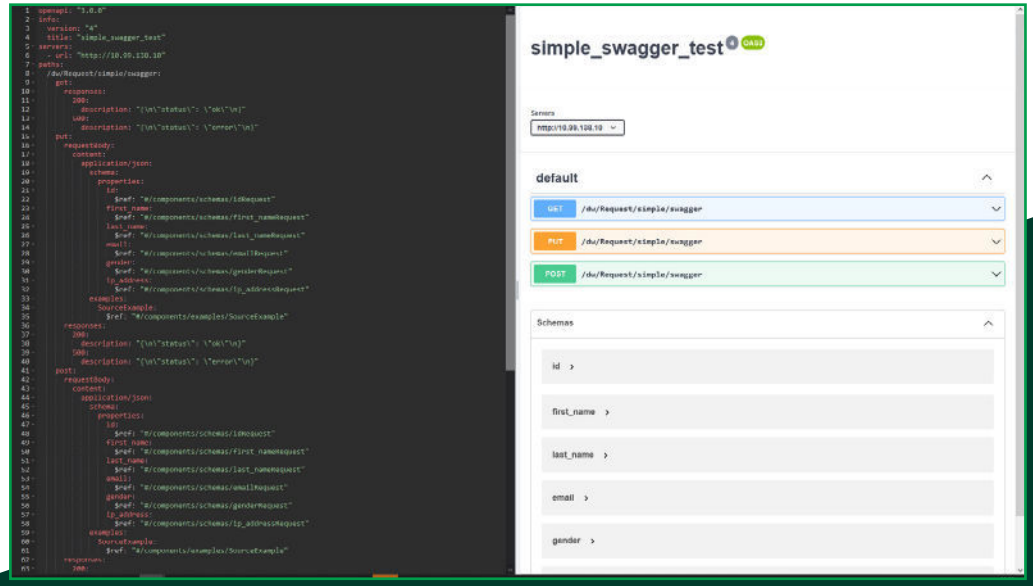


Fig. 13 : Documentation Swagger/OpenAPI automatique

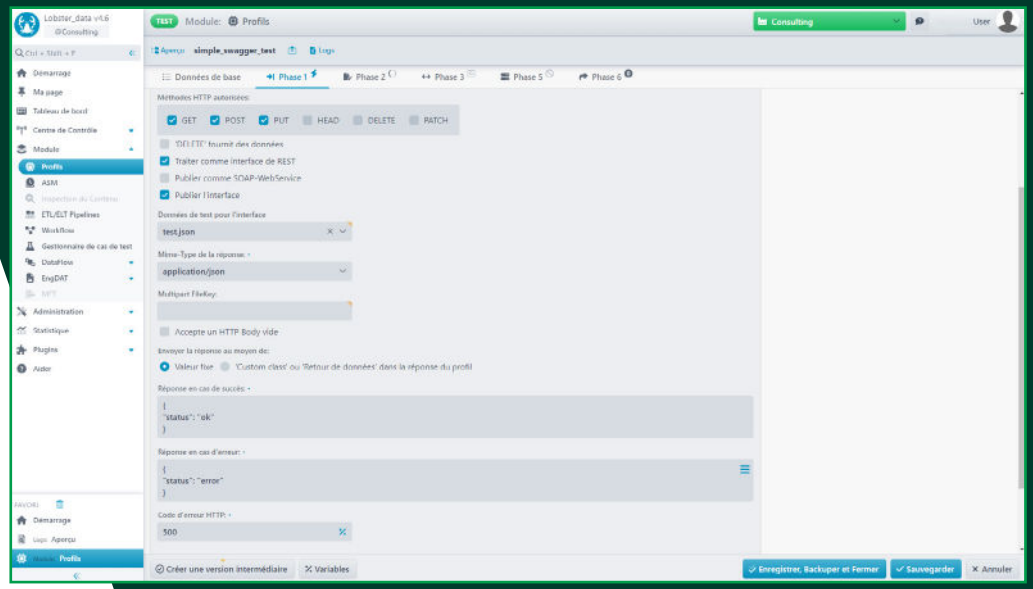


Fig. 14 : Activation de la documentation Swagger/OpenAPI dans Lobster_data

VOTRE ÉCOSYSTÈME API AVEC LOBSTER.

DE L'INTERFACE À L'INTERSECTION.

Disposer d'une stratégie API est important pour toute entreprise qui souhaite avoir une longueur d'avance sur ses concurrents. Une stratégie API bien planifiée et bien appliquée permet d'économiser des coûts, favorise l'innovation et la croissance et offre l'opportunité de se concentrer pleinement sur son cœur de métier. Les API ont donc leur place dans une stratégie tournée vers l'avenir et le numérique, quel que soit le secteur d'activité.

Que vous ayez déjà des réponses ou que vous n'en soyez qu'à la phase de recherche, grâce à Lobster, la mise en œuvre d'une stratégie API ne constitue plus un obstacle ! Et le jeu en vaut la chandelle, car l'ancien monde des écosystèmes informatiques est cher et lent. La dépendance aux prestataires externes et le manque de main-d'œuvre qualifiée compliquent encore les choses.

L'approche no-code innovante de Lobster change la donne : vous pilotez la mise en œuvre, vous ne dépendez pas de prestataires externes et vous êtes rapide. Lobster vous facilite l'accès à l'économie API grâce à une implémentation rapide, une configuration sans programmation et une plateforme qui traite données et processus comme un tout. À la manière d'une prise universelle, Lobster connecte tous les systèmes internes et externes, du cloud aux appareils IoT en passant par les machines. Afin que vous puissiez retirer tous les avantages d'une stratégie API pour votre entreprise.

Lobster vous fournit exactement la boîte à outils qu'il vous faut pour votre économie API. Avec la mise en œuvre technique par API REST et la description d'interface automatique Swagger/OpenAPI sous forme de mécanisme lisible par l'homme comme par la machine. Avec des liaisons configurables en no-code pour les systèmes anciens et nouveaux. Avec des connecteurs préconfigurés. Avec de nombreuses fonctionnalités offrant un haut niveau de sécurité et d'évolutivité, afin de garantir la longévité de votre stratégie API. Que vous utilisiez JSON, YAML ou XML.

Et l'expertise de Lobster ne se limite pas aux API, mais intègre également l'EAI, EDI, ETL/ELT, IoT, Industrie 4.0. Toutes ces applications sur une même plateforme. Lobster crée tout type de connexion vers tout type de système, indépendamment de la source et du format. Vous aimeriez en savoir plus ? Alors, prenez rendez-vous avec l'un de nos experts. Si vous le souhaitez, nous nous ferons un plaisir de préparer une preuve de concept sur mesure et de vous expliquer directement les avantages de notre solution pour votre entreprise.

Pourquoi Lobster?

1. Avec un taux de recommandation net de 99 %, Lobster compte parmi les intégrateurs les plus appréciés.
2. Notre taux de fidélisation de 98,3 % en dit long sur la satisfaction des clients de Lobster.
3. Plus de 80 % de nos clients traitent l'ensemble de leurs cas d'utilisation avec Lobster_data.
4. Plus de 90 % des clients de Lobster intègrent Lobster_data en moins de 24 jours.
5. Plus de 97 % des clients de Lobster obtiennent un retour sur investissement rapide.

Nous nous occupons de vos questions.
Prenez rendez-vous ou écrivez-nous dès maintenant.

info@lobster-france.com

Lobster

Lobster DATA GmbH
Bräuhausstraße 1
82327 Tutzing
T: +49 8158 4529 300
F: +49 8158 4529 301
lobster-world.com

Lobster – France
T: +33 1 74 54 66 27
info@lobster-france.com