



API MANAGEMENT. THE ULTIMATE GUIDE.

UNLOCK YOUR FULL POTENTIAL WITH LOBSTER.

CONTENTS

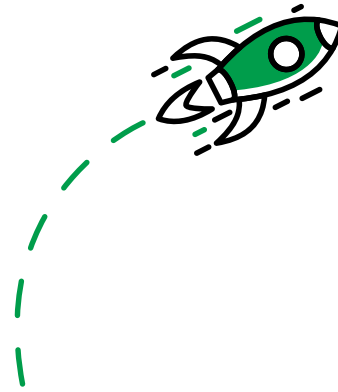
API BASICS. _____	2
Key terms for getting started. _____	2
THE 5 STEPS TO YOUR API MANAGEMENT. _____	3
Step 1: Business plan _____	3
Step 2: API design _____	5
Step 3: Development & publication _____	8
Step 4: Monitoring & reporting _____	10
Step 5: Versioning & retirement _____	10
HOW TO DESIGN YOUR API ECONOMY WITH LOBSTER. _____	12



Lobster

API BASICS.

KEY TERMS FOR GETTING STARTED.



API ecosystems harbour incredible potential for businesses: efficiency gains through electronic data interchange, data monetisation and access to innovative business models. But to establish successful API strategies, companies need to first understand their existing options, get to grips with the technical basics and master API management. This is where Lobster comes in. With the right tool for each step of the way. So you can hit the ground running and drive for digital.

API

API stands for Application Programming Interface. It gives developers access to a toolbox of ready-made, standard commands for implementing software applications. Where they don't need to program new code. An API is a publicly available, standardised software intermediary which collects information from or transfers it to applications.

API publishing

API publishing is the process of making previously proprietary data systems accessible, so it can be consumed both internally or externally and one-way (e.g. shipping status requests) or two-way (e.g. exporting a route to a navigation system and sending the vehicle speed back to the navigation system).

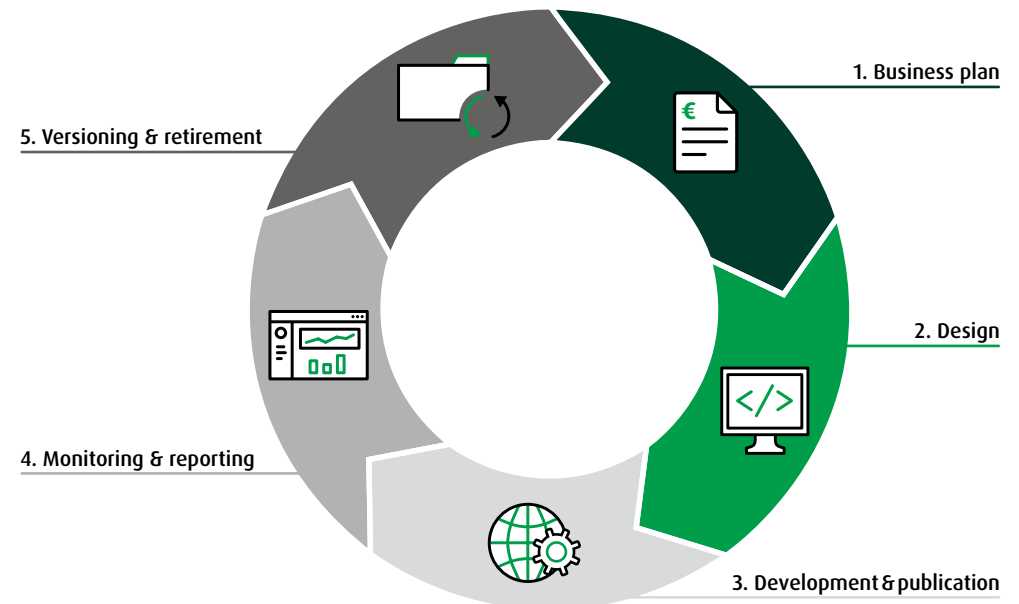
API strategy

If companies give each other access to their data via APIs, then this is known as an API-based, digital ecosystem or an API economy. Sharing information and skills within an API ecosystem drives growth and fosters innovation for all involved. An API strategy is where you take a reflected approach to creating an API system, while API management is implementing that strategy in order to develop, monetise, publish, document, analyse, version, and retire APIs throughout their entire life cycle – hence the use of the term API life cycle management.

An API strategy is both technically and economically effective. On the one hand, a considered and well-delivered API strategy can cut costs by streamlining and automating processes ('API-led integration'). On the other, it also acts as a growth driver and unlocks innovative business models ('API-led connectivity'), cultivates new revenue streams and increases business value across the entire ecosystem.

A successful API strategy is the product of careful planning. Which is why it is so important to understand the necessary steps of each phase and relevant factors during its implementation. The best way to do this is by considering the entire API life cycle.

Fig. 1: The 5 phases of the API lifecycle



Lobsterpedia. Lobster's handy dictionary for your digital transformation.

THE 5 STEPS TO YOUR API MANAGEMENT.

STEP 1: BUSINESS PLAN.

APIs are scalable tools for integrating internal/external data and skills for adding value. The first step is therefore defining the objective of an API strategy. This will identify the most important data areas and determine relevant customer cohorts internally and externally. Both the business departments and IT should take an active part in this process to clarify, e.g. whether existing APIs already meet current requirements. This avoids redundancy and prevents overcomplication of the API landscape.

At this stage, it is important to focus on the most important and time-intensive business processes to determine whether APIs can help to improve these workflows. First up – taking stock of existing customers: What information/skill requirements could be addressed by introducing the relevant API systems? What information should be made available from which systems and in what format?

To ensure the API strategy is a win-win for all, it can be helpful to engage in an in-depth conversation and create a proof-of-concept process with key accounts. Taking a look at what competitors are doing to learn how similar businesses are profiting from the benefits of an API strategy is also beneficial here.

Once workable use cases have been identified, it's important to weigh up their possible costs and revenue potential. This will take the project from a vague concept to a definitive business plan. Which monetisation model you go for will depend on the target group, the value-add and the business goals of the company. At this stage a forecast of API-related costs, e.g. for the required computing performance and security, is vital.

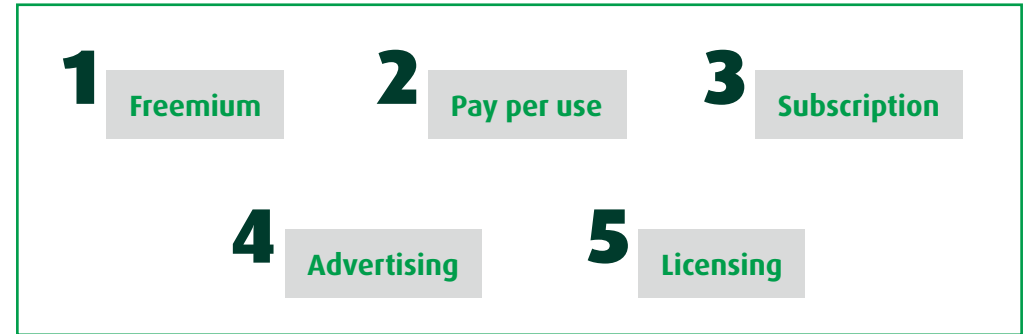
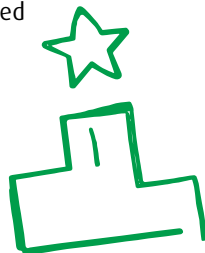


Fig. 2: API monetisation models

Examples of common API monetisation models:

- **Freemium:** This is where the company offers a free basic version of its API. Charges are only applied when premium features are used, or if the API is used frequently. So the weather channel can use an API to let users access temperature information and daylight hours for free but can also impose a fee for hourly weather forecasts or historical weather data.
- **Pay per use:** With this option, a company charges its customers for the number of times they access the API and also applies a fee for each login, per MB, with or without volume bandwidths, via an account where credit can be added. This would allow an e-commerce platform to charge for every product search using its API or for every sale processed via its API. The pay-per-use model is most common when the value of the API is inherently linked to its usage.
- **Subscription:** With a subscription model, a recurring fee is paid in order to access an API – often with various increasing levels of more sophisticated features and uses (e.g. Basic, Pro, Expert).
- **Advertising:** Here, the API is free. Revenue is generated by showing adverts when the API is accessed or by integrating advertising platforms and their APIs. This monetisation model is often used in the mobile-app sphere.
- **Licensing:** Here, the API becomes part of a product portfolio, which is covered by a licensing fee or is in itself a licensable product.

HOW LOBSTER SUPPORTS YOUR BUSINESS PLAN.

- Accompanying the entire API life cycle with advanced security and maximum performance.
- An all-in-one tool including connectors and pre-programmed data conversion features. Customers that use Lobster_data only need one solution for API and EDI, EAI and IoT.
- Exceptional user-friendliness through configurable, no-code design so no programming skills needed. IT specialists deliver high-performative results faster. Employees in business departments create their own solutions as Citizen Developers.
- Completely hybrid tool that is not only cloud compatible but can also be operated on-premises, or independently in your own private cloud. No more external or outbound calls, Lobster_data also runs on in-house customer servers, VMs or Docker containers in the internal network.
- Condensed training at a 3-day Lobster_data foundation course with additional further training options at the Lobster Academy in groups or on-demand.
- In-depth online documentation, interactive user support, 'help' mode with videos and tutorials for all Lobster_data Modules, tools and features.

WHAT CAN THE API MANAGEMENT SOLUTION LOBSTER_DATA BE USED FOR?

- All industries from logistics, e-commerce and manufacturing to food and textiles (industry-agnostic).
- Internal system integration or building API-based ecosystems with external partners.
- Creating REST APIs and web services as simultaneously usable as server and client.
- Connecting all end points and breaking down data silos thanks to outstanding functional depth.

MONETISING YOUR API STRATEGY WITH LOBSTER_DATA.

- E2E functional transparency and analyses via the Lobster_data dashboard plus configurable reporting data pipelines.
- Seamless transmission of usage statistics to third-party systems for implementing all monetisation models!

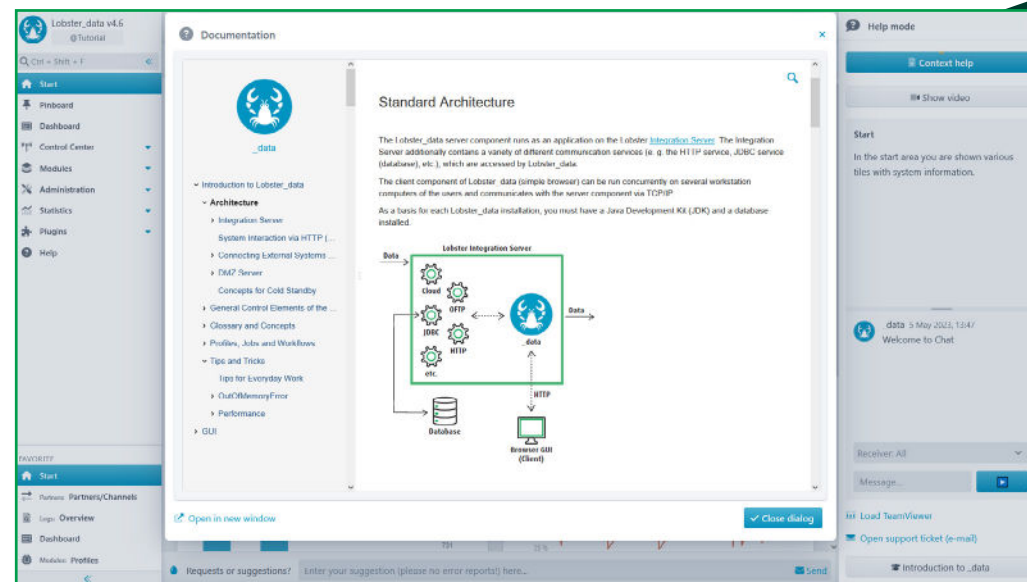


Fig. 3: Lobster_data help mode

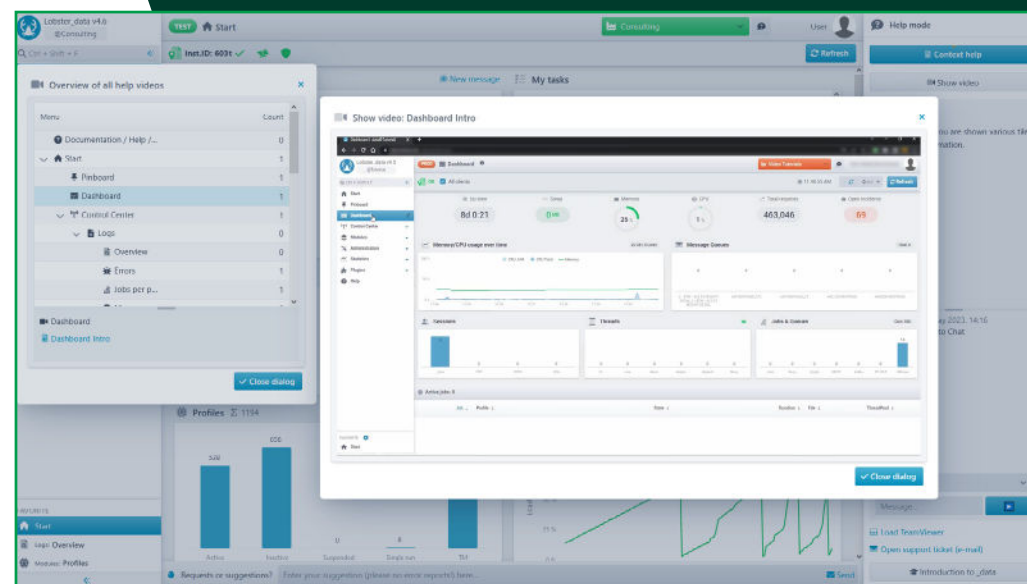
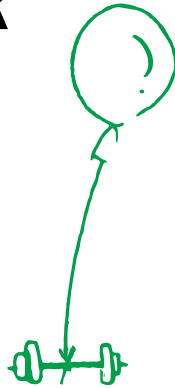


Fig. 4: Lobster_data help video

THE 5 STEPS TO YOUR API MANAGEMENT.

STEP 2: API DESIGN.



Designing an API is an important phase of any strategy as it has a significant impact on the functionality and usability of the API. Good design is only possible if the various parties (product managers/process owners, software developers and stakeholders) collaborate.

Companies should pay particular attention to functional scope, user friendliness, scalability/performance and security. Choosing a tech stack, defining the data structure and data flows and specifying error resolution processes. In addition, the API's user interface should be designed to allow for rapid integration of third parties – making simplicity and consistency paramount as fundamental design considerations.

Ensuring connectivity with the necessary systems is another part of API design. Depending on the complexity of the IT landscape, old and new systems need to be connected via different input and output channels. In heterogeneous systems, messaging formats can vary widely. So, exchanging them via a middleware is the easiest way to homogenise them.

API performance is incredibly important. APIs must be designed to run efficiently, even when processing high data volumes. To ensure the necessary performance, a well-designed, scalable infrastructure, e.g. via a cloud provider, is key. It reduces API latency and increases reliability. Caching common data and introducing load balancing can help optimally balance query load and reduce downtime. Finally, API standards such as http, JSON and REST can greatly improve API performance.

No matter how many elements have already been introduced to ensure optimal API performance: You will still need to continually optimise your APIs after they have been implemented, and recalculate your TCOS. But the most important criterium is ensuring the API always meets customers' requirements and generates revenue.

Finally, security is a key concern when designing an API, as it offers an access point to business systems and data. Advanced APIs can even provide automated contracts with authentications, authorisations, guidelines and by supporting blockchain technology. The potential data pitfalls for customer and company are therefore considerable, which is why server systems should always be secured using the following methods:

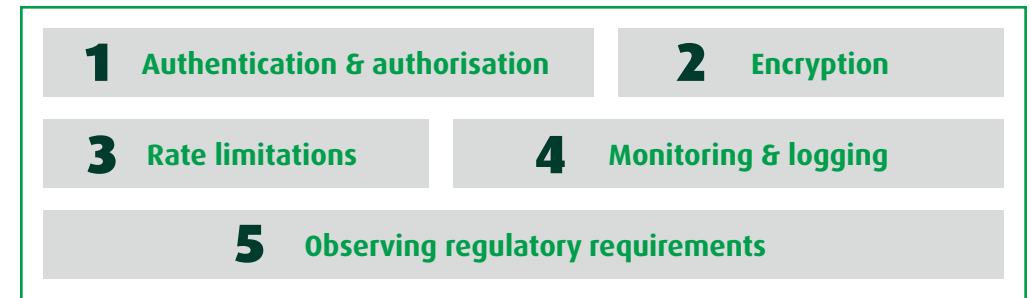


Fig. 5: How to secure a server system

- Authentication & authorisation processes via OAuth and OpenID Connect ensure that only authorised users and systems can access APIs.
- Encryption protects data during transmission and when stored.
- Rate limitations prevent malicious parties flooding the API's network traffic with queries (DDoS attacks).
- Dedicated monitoring and logging routines allow providers to examine API usage, identify potential security risks and respond promptly in case of security breaches.
- Observing industry standards and requirements such as HIPAA, SOC2 and PCI-DSS ensures regulatory compliance with all security requirements.

As with API performance, API security doesn't simply end once suitable measures have been implemented, but has to be reviewed regularly, and should continually inform the cost analyses of your API strategy.

DESIGN AND MOCK-UP OF APIS MIT LOBSTER_DATA.

- Easy creation of multi-layer API systems for integrating data from all internal (EAI), external (EDI) and machine systems (IoT).
- True multi-tenant environment and client compatibility – both for public clouds and on-premises. With lots of options for regulating visibility and API access.
- Ability to create REST and SOAP APIs to receive, send or convert data.
- Connectivity with entire system landscapes. Be they new or old, specialised or generalised systems, excel or EDIFACT. Lobster's 6-Phase Model also unlocks every data silo (incoming channels, parsing, mapping, DB cooperation, outgoing files, out-going channels), creating identically structured, low-maintenance APIs and data pipelines. Connectivity is no longer dependent on the employee, incoming channel, data format or outgoing channel.
- Extensive directory with over 450 calculation functions for processing, consolidating and combining data.
- Support for API calls for invoking an API with another API (API mashup).
- Dynamic routing and ability to define context-specific interface structures.
- Specifying API schemes and publishing them via Swagger/OpenAPI interface descriptions for automated documentation, coding and test case generation.
- Graphical modelling of complex workflows with a workflow designer.
- Accompanying the entire API life cycle with add-ons for every challenge (validating, enriching or decision-making processes with workflows).
- Universal HTTP-API web service support via 'http' input agent when receiving a http request. Once the service has been published, Lobster_data generates all elements for building a SOAP or REST web interface.

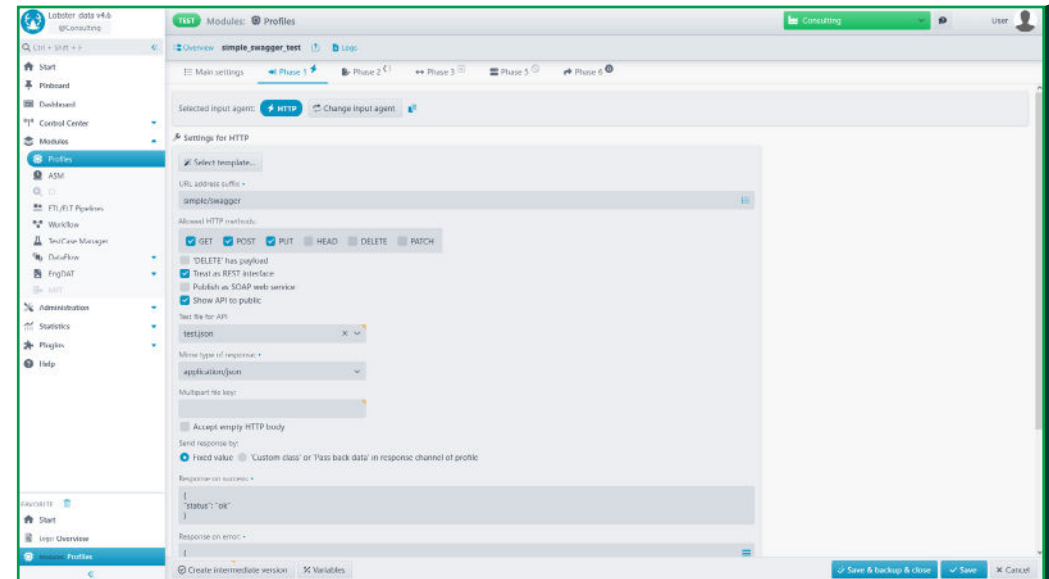
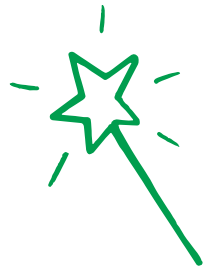


Fig. 6: Lobster's 6 phase model for standardised access to every data silo.

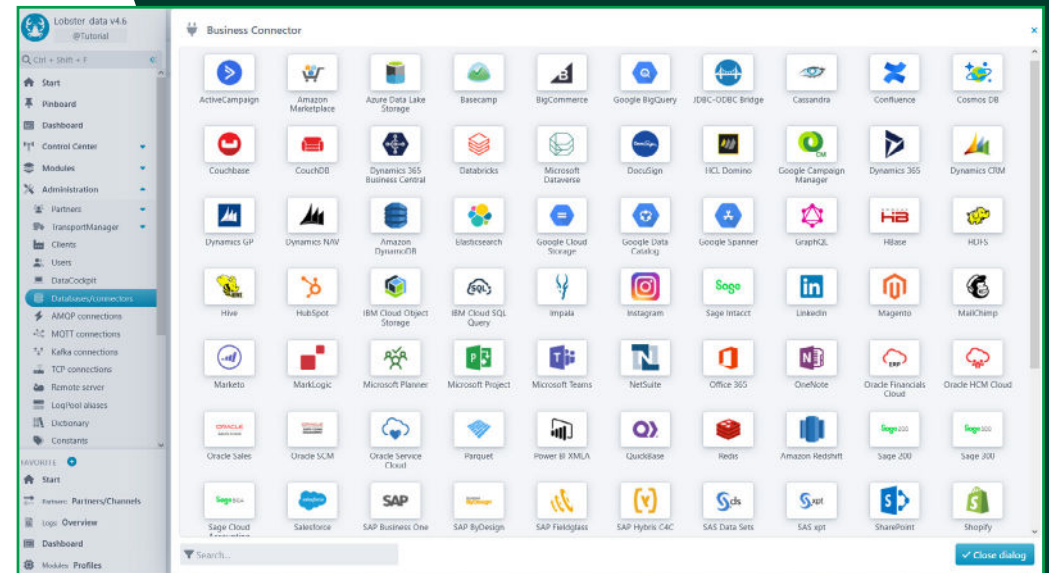


Fig. 7: Ready-made connectors for connecting legacy systems

USER-FRIENDLY APIS WITH LOBSTER_DATA.

- Internal interface configuration via no-code GUI and intuitive design process for REST API clients and servers in just a few clicks – no programming or in-depth knowledge about HTTP/HTTPS/HTTP authentications needed.
- Straightforward and compact interface featuring a dashboard for all APIs.
- Multi-user capability for independent teams working in parallel.
- Swagger generation functions & automatic OpenAPI interface documentation.

SCALABLE & DYNAMIC FOR ALL PERFORMANCE REQUIREMENTS.

- Unlimited service scalability implemented either via self-service, consulting or Lobster's full-service delivery.
- Scalable deployment on-premise, in the Lobster cloud, your private cloud or a hybrid cloud.
- Load balancing and high availability for dynamic traffic spikes with multiple working nodes within a regionally distributed infrastructure (server, VM, docker container).
- Patching and updates without downtime, with rolling processes and successive updating of working nodes during API operation.
- Efficient management of workload. Intelligent proxy compression allowing for less bandwidth and reduced latency.
- Streaming APIs via Apache Kafka for high-performance, real-time data pipelines and data integration.

MAXIMUM API SECURITY.

- Modern authentication methods such as OAuth2.0, SSO, 2FA, MFA, LDAP. HTTP/HTTPS connections for both receiving and transmitting.
- Clear proof of quality through Lobster certification for ISO 27001 Information Security and ISO 27018 Data Privacy Cloud Solutions.
- Central management of all communication partner's certificates for signing and encrypting.
- Automatic acceptance of HTTP authentication – both for basic authentication and Digest Access Authentication via the server.
- User-specific import of Java classes and access to further resources such as encryption directories, schema validation tools, data validation directories for advanced security requirements.

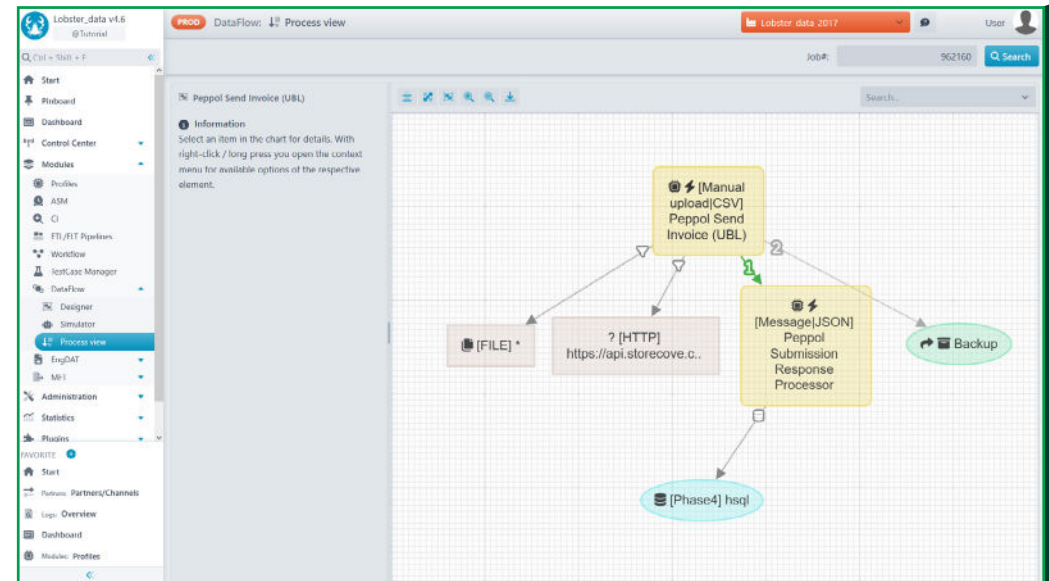


Fig. 8: API visualising via diagrams in the DataFlow Designer

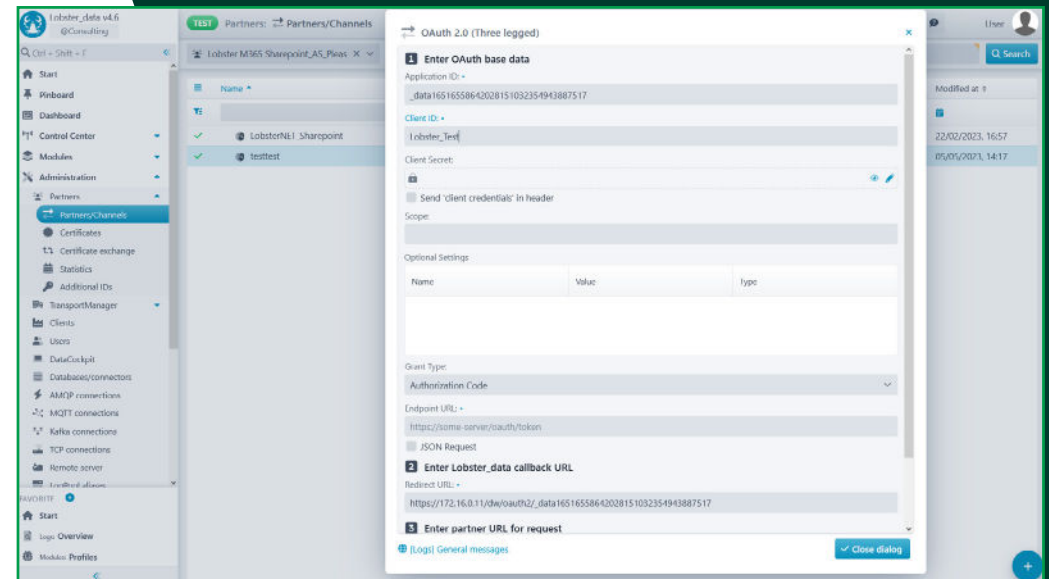


Fig. 9: User-friendly configuration of all authentication types

THE 5 STEPS TO YOUR API MANAGEMENT.

STEP 3: DEVELOPMENT & PUBLICATION.

Once the concept is clear, it's on to the development phase. After being built, an API must repeatedly undergo intense business logic and technical testing. It can only go live – i.e. be published – once four stages (development, test, pre-live, live) are successfully completed. Each stage has its own challenges.

The **development phase** is dedicated to building and implementing the API based on the agreed design specifications. Coding principles such as modularity, maintainability and reusability are key in this phase. Code checking and unit testing (module tests) ensure quality and stability of the underlying code.

Here, thorough API documentation becomes incredibly important. It should contain clear instructions about how to effectively use and integrate the API, examples for requests and responses, error codes. Usually, two versions of the documentation are generated:

- One for internal developers as a comprehensive and up-to-date reference manual, containing technical information such as end points, query and response formats as well as security logs
- And another for external users as a simplified and user-friendly documentation, highlighting the benefits and functionalities of the API in an easily understandable way.

A well-documented API therefore not only accelerates any further developments, streamlines maintenance processes and facilitates internal training. It also improves the end user experience and reduces the number of support tickets.

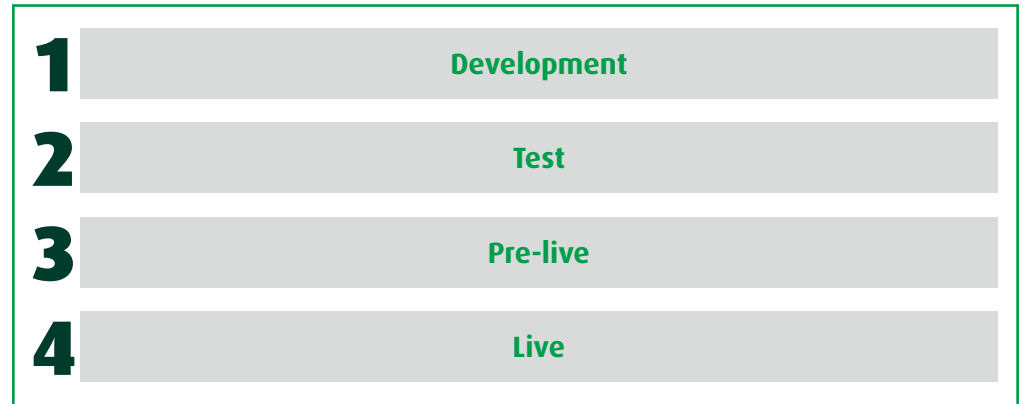


Fig. 10: The four phases of API development & publication

During the **test phase**, the API is thoroughly and systematically tested by developers and – where relevant – stakeholders for early detection of functional, performance-related or security issues. to resolve bugs, compatibility challenges and security gaps and carry out functional as well as non-functional tests.

Functional tests check whether the API maps the desired business logic, delivers the expected results and appropriately resolves errors and deviations. Non-functional tests check the performance, scalability, security and reliability of the API. This includes criteria such as response time, resilience, data protection and disaster recovery.

If the test phase was successful, then next up is the final API validation, which acts as a dress rehearsal, i.e. the **pre-live phase**. Here, the API is tested in an environment that is as similar as possible to the live version: Can the API communicate with other systems and services? Is the API documentation thorough enough? Is the API user-friendly?

Finally, in the **live phase**, the API and its specification (range of functions, call parameter, security functions, result, etc.) are published, meaning they can be accessed by end users. In this phase, it is important to monitor performance as closely as possible and continue to update and maintain the API at regular intervals to resolve errors, address security issues and tweak the interface, if necessary. It's also important to get feedback from users to keep improving the API over time and to ensure it continues to meet the requirements of the company and its stakeholders.

FROM START TO FINISH. THE ENTIRE API LIFE CYCLE WITH LOBSTER.

- Automatic development and documentation of APIs with OpenAPI (Swagger) interface descriptions, automatically generated by Lobster when information is added.
- Rapid creation of communication connections (REST, AS2, SFTP, SMTP, X.400, etc.) and numerous ready-made options for reading or creating different data formats (XML from XSDs, JSON, EDIFACT, CSV, etc.) thanks to rapid proto-typing of mock-up APIs.
- Risk-free checking of API design in test and live environment, as all Lobster_data Editions come with a test and a production installation as standard.
- Rapid, comprehensive transfer of changes from test to live via Lobster_data's Transport Manager, which securely transfers all relevant data from the test system to the target system.
- Transfer of APIs to the production infrastructure with optional extensions such as high availability and load balancing. Depending on API load behaviour: Defining scaling parameters for possible deployment of additional working nodes.

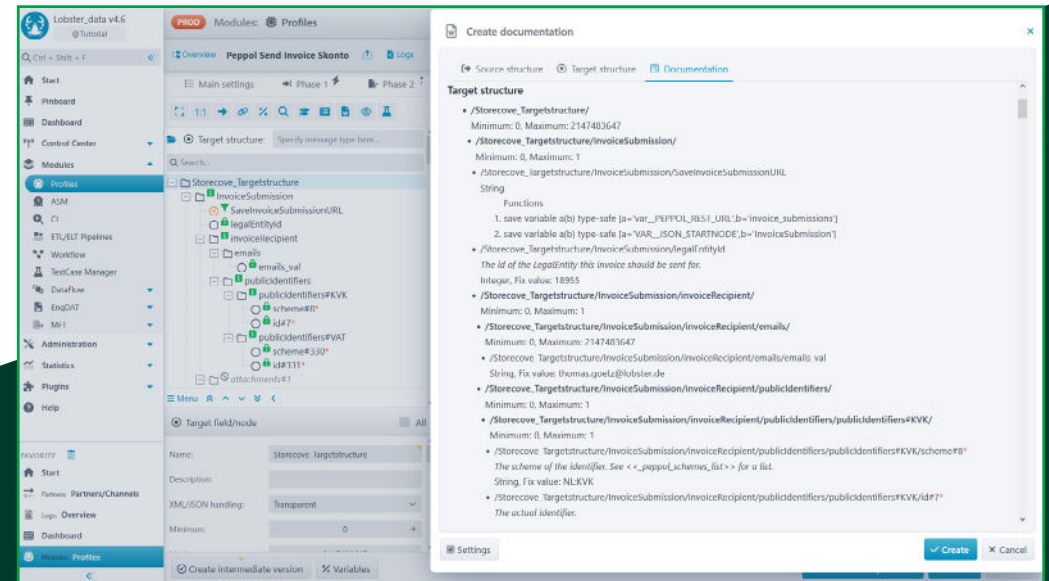


Fig. 11: Seamless transfer of information via automatically generated documentation

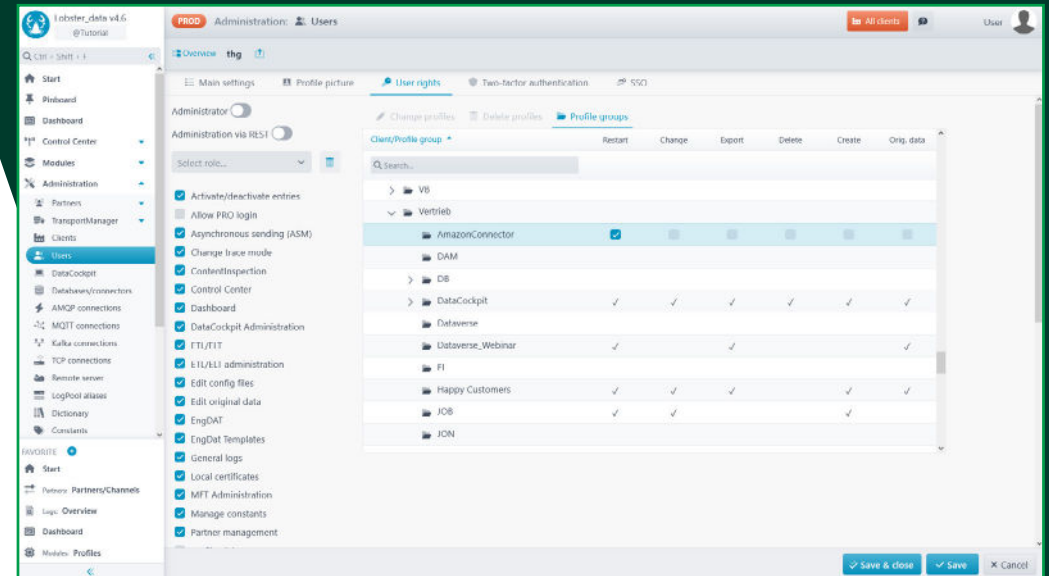
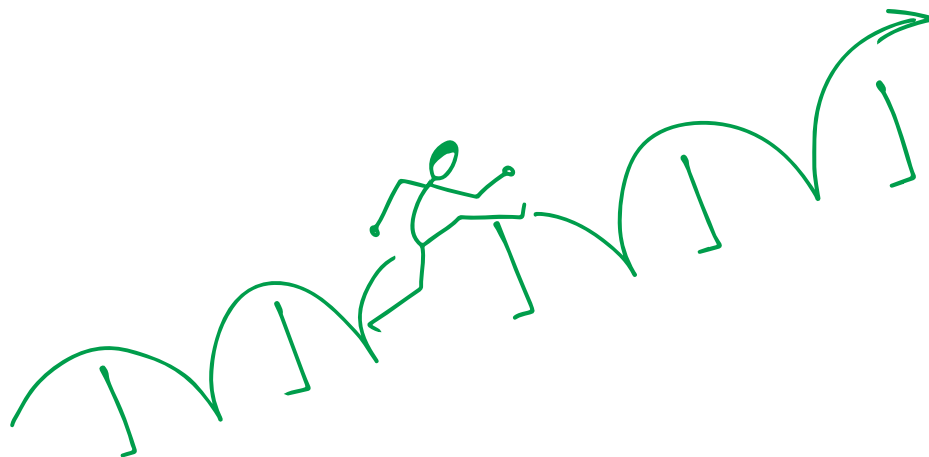


Fig. 12: Easy transition to live phase thanks to user-friendly role distribution

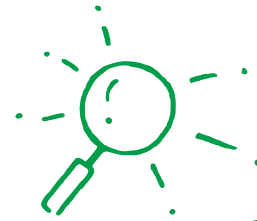


THE 5 STEPS TO YOUR API MANAGEMENT.

STEP 4: MONITORING & REPORTING.

Once an API has gone live, it's important to keep monitoring its performance, usage, security and commercial success. Each monitoring process has its own unique challenges:

- Monitoring API performance means assessing how quickly and efficiently the API can react to queries and deliver data. Monitoring response times, error rates and server load will deliver high-quality data for this purpose.
- How well an API performs is closely connected to who uses it, how it is used and at what intensity. Recognising these usage patterns is important for regulating manager attention, helping to determine which API features are most valuable to which customer segments and allowing for adequate responses to unexpected query spikes.
- API security is safeguarded through regular vulnerability scans and penetration tests, whilst log data is checked for unusual activities. Security protocols are to be continually updated.
- Last but not least, it is a good idea to keep an eye on income generated by the API as well as any associated outgoings. On the one hand, this means monitoring user growth and the number of their queries within the chosen monetisation model. On the other hand, it's also important that running costs for API management, the underlying infrastructure and support are tracked to get a clear picture of the TCO and cost-benefit ratio.



STEP 5: VERSIONING & RETIREMENT.

During operation, it is also possible to create new versions of the API, e.g. by changing the output information or the request or output format such as integer value to floating point number. This new API version must also be carefully documented and is typically made available via a different URI or an adapted request header.

Sooner or later, every API reaches the end of its product life cycle. Pinpointing the right time for retirement is key because shutting down an API prematurely can result in missed revenue. If, on the contrary, the API retires too late, this can result in unnecessary costs. But since retiring an API is an important step for the entire ecosystem, it should be well thought through to minimise the impact to connected systems and the user experience overall.

You should think about shutting down your API if:

- The usage rate is declining although operating costs are not going down
- The features offered are past their prime, which is impacting performance or usability
- The business environment has changed, making the API obsolete
- The API system represents a security risk to connected companies

It is up to the API management team to carefully follow the API's life cycle and proactively plan for retirement. Just as when launching the API, it is important to prepare the API shut-down properly (who, why, what, when, how), communicate the change internally and externally in good time and set a reliable schedule. The aim here is to again offer users a smooth transition with minimal impact to business operations and allow time to potentially make alternative APIs available to users. Key in all of this is giving the user enough notice so they can make the necessary adjustments to their systems. Finally, the API team should update all documentation and remove any references to the retired API.

MONITORING IN LOBSTER_DATA. IN THE DRIVING SEAT.

- Comprehensive monitoring via a dashboard with general and analytical reports.
- Real-time tracking of performance data for every application and API message type.
- API usage rates for determining maximum size and response time.
- API statistics data and their compacted or uncompact calculation and presentation.
- Debugging tools and alerts in case of deviations.
- Detection of data injection attacks via the customer's virus scanner.

REPORTING IN LOBSTER_DATA.

- Transparency features for intuitive involvement of all API stakeholders.
- Asynchronous data collection for smooth operation of ongoing data traffic.
- Creation of user-defined (ad-hoc) reports – preconfigured or on-demand.
- Easy export of analysis data to data warehouse systems or data lakes.
- Real-time trend analyses by checking the profiles themselves, the dashboard or the admin console.
- Real-time notifications for exceptions (log entries, email, ticket system).

VERSIONING & RETIREMENT.

- Scalable life cycle management with e2e transparency and control.
- Integrated version maintenance with allocation of a unique ID and version number, mandatory creation of a new version in case of changes and creation of a version back-up.
- Automatic creation of a Swagger/OpenAPI documentation for every API version.
- Version and variant management for creating, importing, exporting, replicating, deactivating or updating APIs.
- Machine-readable retirement announcement with communication of a response header X-API-Warn.

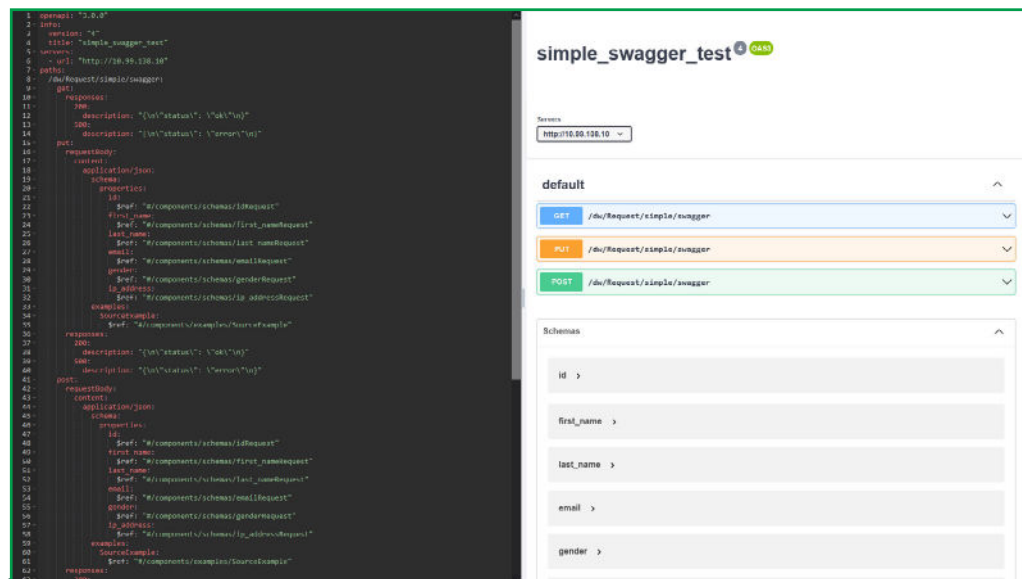


Fig. 13: Automatic Swagger/OpenAPI documentation

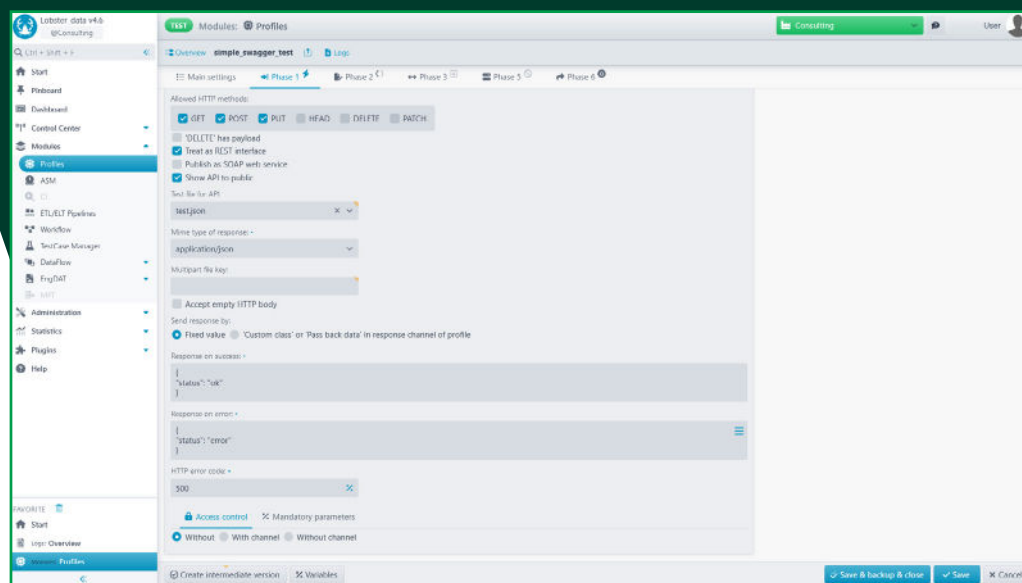


Fig. 14: Activation of Swagger/OpenAPI documentation in Lobster_data

HOW TO DESIGN YOUR API ECONOMY WITH LOBSTER.

FROM INTERFACE TO INTERACTION.

Companies looking to stay ahead of the competition should definitely consider an API strategy. A well-planned and delivered API strategy saves money, drives innovation and growth whilst also freeing up capacity to focus on core business operations. APIs are therefore a key component of any future-oriented, digital-centric corporate strategy no matter the industry. Where could APIs add value in your company?

Whether you already know the answer to this question or are still finding your feet – thanks to Lobster, implementing an API strategy is no longer a hurdle. So why wait? With conventional IT ecosystems getting slow and expensive, external service providers adding their own challenges not to mention the growing skills shortage – there's no time like the present to try a new approach.

Lobster's innovative no-code technology is here to change that: You manage the implementation, remain independent of external service providers and stay agile. Lobster is paving the way to your API economy with its rapid delivery and delivery timelines, no-code configuration and a platform that treats data and processes as one. Like a universal adapter – Lobster connects all internal and external systems, from the cloud to machines all the way to IoT devices. So you really can leverage your API strategy and unlock digital opportunities for your business.

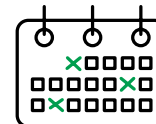
Lobster offers the perfect toolbox for your API economy. Technically delivered via REST API. With automatic human- and machine-readable Swagger/OpenAPI interface descriptions. With configurable, no-code connections to old and new systems. With ready-made connectors. With countless features for maximum security and scalability, ensuring your API strategy is sustainable. No matter whether JSON, YAML, or XML.

And Lobster can do so much more than API. EAI, EDI, ETL/ELT, IoT, Industry 4.0 are just a few of the strings to its bow. All applications are available on one platform. So you can build connections to all systems no matter the source or format. You want to learn more? Then book a meeting with one of our experts. We're happy to prepare a custom PoC for your business, highlighting the benefits of our solution for your business.

Why Lobster?

- 1 Lobster is one of the most popular integrators with a net promoter score of 99%.
- 2 Our incredible 98.3 % customer retention rate proves just how happy Lobster customers are.
- 3 More than 80 % of all customers cover all their use cases using Lobster_data.
- 4 Over 90 % of all Lobster customers fully integrated Lobster_data in less than 24 days.
- 5 Over 97 % of Lobster_data customers quickly see a positive ROI.

We're happy to help with your questions.
Schedule a meeting or drop us an email today.



information@lobster.de



Lobster

Lobster DATA GmbH

Bräuhäusstraße 1

82327 Tutzing

T: +49 8158 4529 300

F: +49 8158 4529 301

information@lobster.de

lobster-world.com



Lobster

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