Be sure. testo

Unlocking integration: API and webhooks

www.testo.com/solutions

Table of contents

Purpose of the whitepaper

This whitepaper provides a comprehensive overview of Application Programming Interfaces (APIs) and Webhooks, analyzing their implementation for facilitating efficient, real-time data integration and interoperability within modern systems.

Importance of APIs and webhooks in modern digital systems

In our connected world, we expect technology to seamlessly work together. For the pharmaceutical industry, having different systems communicate flawlessly is not just a nice-to-have, it's essential, as the safety of the patient is top priority. Think of the systems that constantly monitor environmental conditions like temperature and humidity. They are vital for ensuring a medicine is safe, effective, and meets strict regulatory standards.

How do you get all your different systems to talk to each other?

The answer is an Application Programming Interface (API). Think of an API as a universal translator for software. It provides a shared language and a set of rules so that different programs can communicate with each other smoothly and efficiently.

In a pharmaceutical environment, this is incredibly powerful as APIs allow you to:

- **Instant monitoring and alarming:** Get live updates from all your different sensors, alarms, and monitoring tools in one central place.
- **Automated workflows:** Automatically share/integrate environmental parameters data with your Lab Reports or PowerBI, reducing manual work and errors.
- **Future-proof your setup:** This flexible Approach lets you easily scale your system. E.g.: add new technologies, connect to cloud analytics, or upgrade individual components without having to rebuild your entire system from scratch.
- **Facilitate data integration:** APIs help integrate environmental data with Laboratory Information Management Systems (LIMS), Building Management Systems (BMS), and other internal systems at work.
- **Support scalable architecture:** APIs make it easier to scale monitoring infrastructure, connect with cloud-based analytics, and implement modular upgrades without overhauling the entire system.

Webhooks: Real-Time Event-Driven Communication

While APIs enable structured data retrieval, webhooks provide a more proactive, event-driven approach. Webhooks automatically send data to a designated endpoint when predefined events occur, significantly reducing latency and improving responsiveness.

In a pharmaceutical context, webhooks can be used to:

- **Trigger immediate alerts:** Instantly notify personnel or automated systems when environmental thresholds are breached, such as a rise in temperature, humidity, or particle count in cleanrooms.
- **Ensure compliance and audit trails:** Automatically log environmental incidents and corrective actions, contributing to a robust audit trail aligned with regulatory requirements.
- **Optimize operational efficiency:** Facilitate automated responses like activating backup HVAC systems or isolating affected batches, improving response time and reducing human error.

What is an API?

Definition and key concepts

An Application Programming Interface (API) is a set of defined rules and protocols that allow one software application to interact with another. In simple terms, an API acts as a bridge that enables different systems—often built with different technologies—to exchange data and functionality securely and efficiently.

In the context of pharmaceutical environmental parameter monitoring systems, APIs are used to extract data from monitoring devices (e.g., temperature, humidity, particulate sensors) and integrate it into enterprise systems such as data lakes, compliance reporting tools, and quality management platforms.

Key concepts include:

- **Endpoints:** Specific URLs through which data can be requested or actions can be triggered.
- **Requests & responses:** APIs operate on a request-response model, where one system sends a request and receives a response in return, often in a structured format like JSON or XML.
- **Authentication:** APIs often include security measures like API keys, OAuth tokens, or mutual TLS to ensure only authorized systems can access the data.

Types of APIs

There are several types of APIs, each with its own structure and ideal use case. The most commonly used in environmental monitoring systems include:

- **REST (Representational State Transfer)**
  RESTful APIs are widely adopted due to their simplicity and scalability. They use standard HTTP methods (GET, POST, PUT, DELETE) and typically return data in JSON format. REST APIs are ideal for cloud-based platforms and IoT device integration.
- **GraphQL**
  Developed by Facebook, GraphQL allows clients to request exactly the data they need in a single query. This flexibility reduces bandwidth usage and speeds up response times. It's especially useful in systems that aggregate data from multiple sources or require custom dashboards.
- **SOAP (Simple Object Access Protocol)**
  SOAP is a protocol-based API using XML for data exchange and includes built-in error handling and security features. Although more rigid and verbose, SOAP is still used in legacy systems, especially where robust security and transaction compliance are required.
- **WebSocket APIs**
  While not as common in traditional REST-based systems, WebSocket APIs allow for two-way, real-time communication—useful for continuous environmental data streams or live dashboards in critical cleanroom environments.

By enabling dynamic and secure communication between systems, APIs are critical enablers of digital transformation in pharmaceutical manufacturing and environmental compliance.

Benefits of using APIs for integration and data exchange

In pharmaceutical environmental monitoring, APIs offer a wide range of benefits:

- **Seamless integration:** APIs enable interoperability between monitoring systems, enterprise resource planning (ERP) platforms, cloud services, and regulatory compliance tools.
- **Real-time data exchange:** With proper API configurations, data can be fetched or pushed in near real-time, enabling timely alerts and decision-making.
- **Automation and efficiency:** APIs facilitate automation and data integration with other running systems.
- **Scalability:** APIs allow systems to evolve over time. New sensors or data endpoints can be integrated without rewriting existing software architecture.
- **Security and compliance:** With secure authentication and encrypted communication, APIs support the stringent data integrity and access control standards required in the pharmaceutical industry.

By enabling dynamic and secure communication between systems, APIs are critical enablers of digital transformation in pharmaceutical manufacturing and environmental compliance.

What is a Webhook?

Definition and comparison with traditional polling

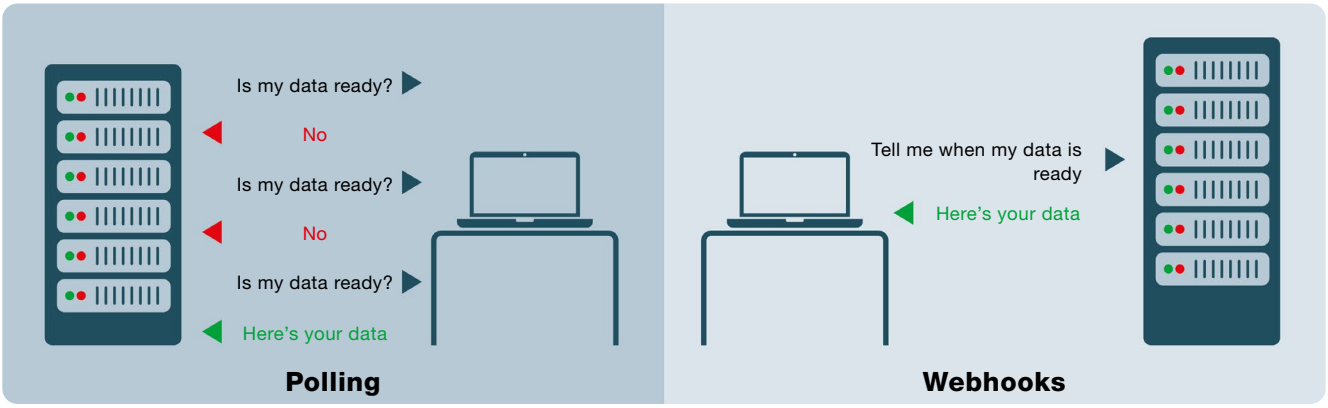
A webhook is a method of communication between systems that enables event-driven notifications. Unlike APIs, which typically require one system to request data from another at regular intervals (a method called polling), webhooks automatically send data to another system the moment a specific event occurs.

In simpler terms, a webhook is like setting up a subscription—when something important happens, you get notified immediately, without having to ask repeatedly.

Polling vs. Webhooks:

Feature	Polling	Webhooks
Mechanism	Client repeatedly asks for updates	Server pushes updates when events occur
Latency	Depends on polling frequency	Near-instantaneous
Resource usage	Higher (frequent requests)	Lower (only when needed)
Scalability	Less efficient at scale	Highly efficient

In environmental monitoring systems, traditional polling might involve querying sensor data every minute to check for threshold breaches. With webhooks, an alert is sent automatically the moment a critical value is detected—reducing delay and improving responsiveness.



Use cases for event-driven communication

Webhooks are ideally suited for real-time event notification in scenarios where time-sensitive data and immediate action are crucial. In Pharmaceutical environmental monitoring, key use cases include:

- **Threshold breaches:** Instantly notify personnel or systems when temperature, humidity, pressure, or airborne particles exceed regulatory limits in cleanrooms or storage areas.
- **System failures or sensor malfunctions:** Alert technicians immediately when a sensor stops transmitting data or behaves abnormally.
- **Workflow automation:** Integrate with maintenance or quality management systems to open a corrective action (CAPA) workflow when an out-of-specification (OOS) condition is detected.

How webhooks enhance real-time interactions

Webhooks transform passive monitoring into proactive management. In pharmaceutical environments, where product quality and patient safety depend on strict environmental control, this real-time capability is essential.
Benefits of webhooks in real-time interaction include:

- **Faster response times:** Webhooks reduce the delay between incident detection and response, minimizing product risk and potential downtime.
- **Reduced manual oversight:** By automating alerting and response mechanisms, webhooks reduce the need for continuous manual supervision of sensor data.
- **Improved compliance:** Webhooks help maintain a robust digital trail by instantly recording environmental events and actions taken, aiding in compliance with FDA, EMA, and other global regulations.
- **Operational efficiency:** Webhooks facilitate smart system orchestration—automatically adjusting HVAC settings, activating backup systems, or isolating affected zones based on incoming alerts.

In essence, while APIs offer structured access to data, webhooks deliver intelligent, event-driven responsiveness — a crucial capability for mission-critical environments in pharmaceutical operations.

Advantages in pharma

Usages of API and webhooks in pharma

In the pharmaceutical industry—where precision, compliance, and operational efficiency are non-negotiable—APIs and webhooks are transforming how systems interact, share data, and automate responses. These technologies are increasingly integral to pharmaceutical environments, especially in areas such as:

- **Environmental monitoring:** APIs pull data from sensors measuring temperature, humidity, pressure, and airborne particles into centralized platforms (e.g., LIMS, BMS, QMS).
- **Laboratory Information Management Systems (LIMS):** APIs enable seamless bi-directional communication between monitoring devices and LIMS, allowing test results to be contextualized with environmental conditions.
- **Supply chain monitoring:** Real-time webhook alerts inform teams of environmental deviations during transport or storage, reducing spoilage risk and enabling corrective actions.
- **Automated CAPA (Corrective and Preventive Actions):** Webhooks trigger workflows in quality management systems (QMS) when out-of-specification conditions are detected.
- **Third-party integrations:** APIs allow pharmaceutical systems to integrate with external analytics platforms like Power BI, notification tools like Microsoft Teams, and cloud infrastructure.

By leveraging these technologies, pharmaceutical companies create ecosystems where devices, databases, and users operate in unison—eliminating data silos and driving faster, more informed decisions.

Advantages (e.g. cost savings, system efficiency)

APIs and webhooks offer a broad range of advantages for pharmaceutical manufacturers and labs, from tangible cost savings to strategic operational gains:

1. Cost savings

- **Reduced manual labor:** Automated data retrieval, alerts, and reporting drastically cut down on repetitive tasks, freeing staff for higher-value work.
- **Minimized product loss:** Real-time alerts via webhooks help catch environmental deviations early, reducing the likelihood of batch spoilage or product recalls.

2. Increased operational efficiency

- **Streamlined workflows:** APIs and webhooks enable systems to respond to events in real time—triggering alerts, logging events, and activating procedures automatically.
- **Faster decision making:** Centralized and real-time data access enables managers to act quickly and with confidence, improving both quality and throughput.
- **Reduced downtime:** Automated error reporting and real-time diagnostics allow for faster root cause analysis and recovery from system or equipment failures.

3. Improved data integrity and compliance

- **Automated, tamper-proof logging:** Events and data are logged automatically, reducing the risk of human error and ensuring alignment with global regulatory standards.
- **Secure access control:** APIs enforce robust access management policies through authentication and authorization frameworks like OAuth2 and JWT.
- **Real-time traceability:** Webhooks react to critical events (e.g., sensor failure, threshold breach) in real time by notifying the user, for example, via Microsoft Teams or even an LED light on premises.

4. Scalability and future-proofing

- **Modular integration:** APIs make it easier to add new technologies (e.g., new sensors, cloud analytics tools) without overhauling existing systems.
- **Platform agnosticism:** APIs and webhooks allow data exchange across different hardware and software vendors, preventing vendor lock-in and enabling long-term flexibility.
- **Support for cloud and Edge architectures:** These technologies support modern infrastructure approaches, enabling hybrid deployments with edge intelligence and centralized cloud analytics.

5. Enhanced communication and collaboration

- **Real-time notifications:** Webhooks push alerts directly to communication platforms like Teams, WhatsApp, or SMS, ensuring the right people are notified without delay.
- **Cross-functional insights:** APIs facilitate the sharing of environmental data across departments—from QA to IT—breaking down silos and encouraging collaboration.

In summary, APIs and webhooks are not only technical conveniences as they are also enablers of digital transformation. This help ensure product quality, operational excellence and cost efficiency.

Architecture and design

RESTful design principles

REST (Representational State Transfer) is a widely used architectural style that leverages standard HTTP protocols for communication between systems. Its lightweight, scalable, and stateless nature makes it ideal for the pharmaceutical industry, where environmental monitoring systems must integrate with enterprise platforms in a reliable, secure, and auditable manner.

Stateless communication

Each REST API request contains all the information the server needs to fulfill it, i.e. no session data is stored between requests. This statelessness improves scalability and reduces system complexity, making it easier to maintain validated environments and ensure repeatability in regulated workflows.

Resource-oriented structure

REST APIs treat data as resources, each identified by a unique URI. These URIs represent real-world objects like sensors, alarms, or reports. For example:

- /sensors/room-101/temperature
- /alerts/threshold-breach/456
- /devices/saveris-1/status

Standardized HTTP methods

REST relies on HTTP verbs to perform actions on resources:

- GET – Retrieve data (e.g., current temperature reading)
- POST – Create a new resource (e.g., an alert or audit log)
- PUT – Update existing data (e.g., sensor configuration)
- DELETE – Remove a resource (e.g., an old log entry)

The consistent method structure enhances integration with other platforms and streamlines development.
testo Saveris 1 offers REST API access via the GET method, allowing users to retrieve data and integrate it with other systems or dashboards. However, PUT, and DELETE actions are intentionally not supported, as they would conflict with 21 CFR Part 11 compliance requirements. To maintain compliance, the system must restrict access and prevent any modifications to recorded data.

Structured, lightweight data

REST APIs typically exchange data using JSON, which is compact and easy to parse. For example:

```
curl --location --request GET 'http://localhost/api/v1/bases/<Base Id>/channels/<Channel Id>' \
--header 'Accept: application/json' \
--header 'x-access-token: <API Key>'
```

This format enables easy interoperability with reporting tools, dashboards, and data lakes.

Meaningful status codes

REST APIs use standard HTTP response codes to indicate the result of an operation:

- 200 OK – Request succeeded
- 201 Created – New resource successfully created
- 400 Bad Request – Invalid input or malformed request
- 401 Unauthorized – Missing or invalid authentication
- 404 Not Found – Resource doesn't exist
- 500 Internal Server Error – Server-side failure

These codes improve transparency and error handling in integrated systems.

Best practices in RESTful design

- **Versioning:** Use path-based versioning (e.g., /api/v1/) to manage updates without breaking integrations.
- **Naming conventions:** Use consistent, descriptive names (e.g. /devices, /reports).
- **Security:** Enforce HTTPS and authenticate using API keys, OAuth 2.0, or JWT tokens.
- **Rate limiting & caching:** Implement limits to prevent abuse and cache static data to reduce load.
- **Avoid verb-based URIs:** Let HTTP methods define the action (POST /alerts, not /createAlert).

Authentication and authorization

APIs in pharmaceutical environments must not only enable integration and automation—they must also protect sensitive data, restrict access based on user roles, and maintain compliance with regulatory standards such as **FDA 21 CFR Part 11**. This makes robust authentication and authorization essential components of API design.

Authentication vs. authorization

- **Authentication** verifies who is making the API request.
- **Authorization** determines what that user or system is allowed to do.

A secure API must support both.

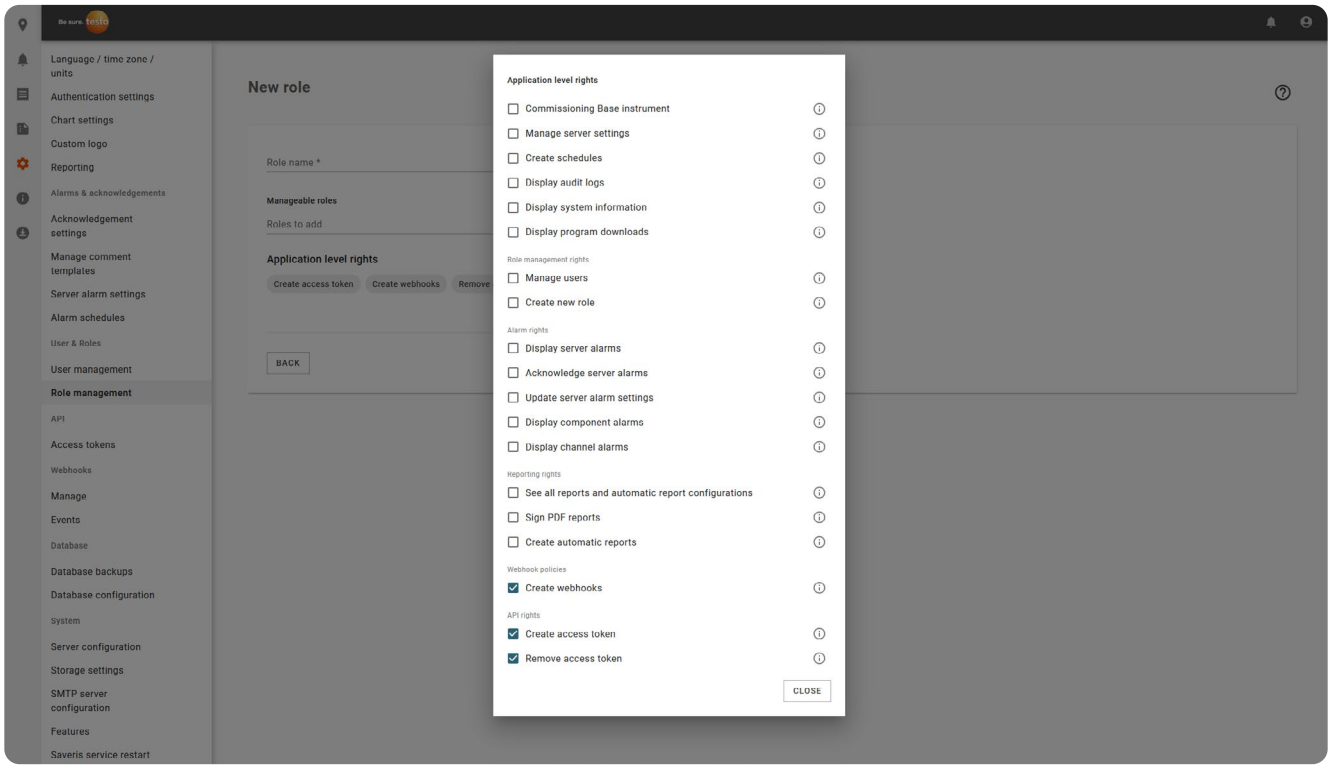


Fig. 1 testo Saveris 1 role application level rights management

Common authentication methods

API Keys

A simple and widely used method, API keys provide an access token that must accompany each request, often in a custom HTTP header. While easy to implement, API keys should always be tied to specific users or roles, and rotated periodically.

Example usage:

```
curl --location --request GET ‘http://localhost/api/v1/bases’ \
--header ‘Accept: application/json’ \
--header ‘x-access-token: <API Key>’
```

This is the approach used in testo Saveris 1, where authentication relies on API Key Auth. Each API key is:

- Bound to specific roles, users and permissions
- Sent in the x-access-token header to protect against unauthorized access

This method aligns with **CFR Part 11** requirements for traceable user access and electronic records integrity.

OAuth 2.0

For more complex or enterprise integrations, OAuth 2.0 provides a secure authorization framework. Instead of exposing passwords or raw tokens, OAuth issues access tokens based on scopes and roles. It's ideal when:

- APIs are accessed by multiple third parties
- Fine-grained permission control is required
- Single sign-on (SSO) is used across systems

OAuth supports token expiration, refresh, and auditing—essential features for pharmaceutical IT governance.

Role-based access in testo Saveris 1

In the testo Saveris 1 system, access to environmental data and functionality is user-based and role-restricted. Each API token is:

- Tied to a specific user within the Cockpit system
- Limited by assigned roles (e.g., Administrator, QA Reviewer, Maintenance Staff)

This granular control ensures:

- Only authorized personnel can pull data or trigger actions
- Sensitive operations

The platform's role-based access model aligns with **FDA 21 CFR Part 11**, helping ensure electronic records are attributable, secure, and audit-ready.

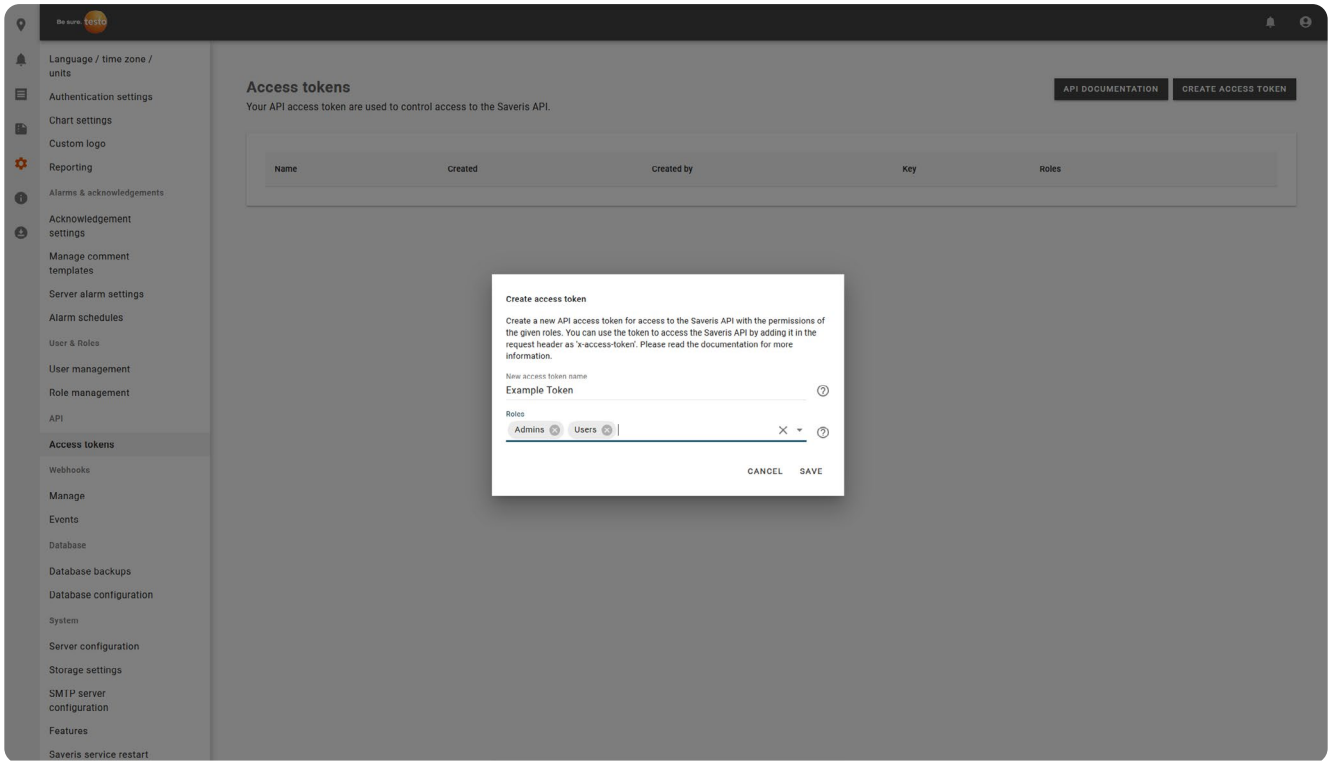


Fig. 2 testo Saveris 1 API access token creation.

Webhook architecture and design

Webhooks are a foundational building block of event-driven system architecture, allowing external systems to react to changes as they happen. Unlike traditional APIs, which require polling to check for updates, webhooks proactively notify a target endpoint the moment a predefined event occurs. This makes them ideal for time-sensitive environments such as pharmaceutical monitoring.

Webhook architecture, when designed with care, delivers the real-time responsiveness modern pharmaceutical operations require without compromising security or compliance. By focusing on structured event triggers, strong security practices, and reliable delivery mechanisms, webhook implementations like those in testo Saveris 1 help bridge the gap between environmental monitoring and immediate, auditable action. In the next sections, we will address these points.

Event-driven systems

In a webhook-enabled system, events serve as triggers. These could be system state changes (e.g., “temperature exceeds threshold”) or operational actions (e.g., “door opened in cleanroom”). When such an event occurs, the system automatically generates a payload—typically in JSON format—and sends it to a registered webhook endpoint.

This model:

- Reduces latency between event occurrence and system response
- Minimizes unnecessary API calls and resource usage
- Supports modularity and decoupling between systems

For example, in the testo Saveris 1 system, webhook events could be generated when:

- A sensor reports a temperature or humidity deviation
- A sensor reports a temperature or humidity changing at a certain speed
- A smart relay reports an action (e.g. door left open)

By adopting this event-driven architecture, pharmaceutical operations can ensure real-time responsiveness without relying on inefficient polling.

Security best practices

Because webhooks push data externally, security is critical. A poorly secured webhook could expose sensitive environmental data or be used to spoof alerts. To mitigate risks, several best practices are recommended:

- **Use HTTPS:** All webhook payloads should be transmitted over secure HTTPS connections to prevent interception.
- **Secret tokens or HMAC signatures:** Each webhook payload should include a signature (e.g., HMAC) generated using a shared secret. The receiving server can then verify that the payload is authentic and untampered.
- **IP whitelisting:** Limit webhook delivery to known, trusted IP addresses to reduce exposure to unauthorized sources.
- **Authentication headers:** Include a unique token or API key in the header to authenticate the source of the webhook.
- **Replay protection:** Include timestamps and unique IDs in each event to prevent replay attacks or accidental reprocessing.

In pharmaceutical environments, these measures support compliance with data integrity and traceability requirements under regulations such as **FDA 21 CFR Part 11**.

Retry mechanisms and idempotency

Webhooks are only effective if the receiving system actually gets the message. But networks can be unreliable, and endpoints may be temporarily down. That's why robust retry logic is essential.

- **Retries on failure:** If a webhook fails (e.g., times out or returns a 500 error), the system should retry using exponential backoff and a maximum retry limit.
- **Delivery confirmation:** HTTP status codes are used to confirm delivery. A 2xx response indicates success; anything else may trigger a retry.
- **Dead letter queues (optional):** For long-term reliability, failed events can be logged for later review or manual reprocessing.

To prevent duplication, webhook handlers must be idempotent. This means:

- Processing the same event multiple times will produce the same result
- Events are tagged with a unique identifier so they can be safely ignored if already processed

This is especially important in regulated environments where redundant logging or actions could confuse audit trails or trigger unintended system behavior.

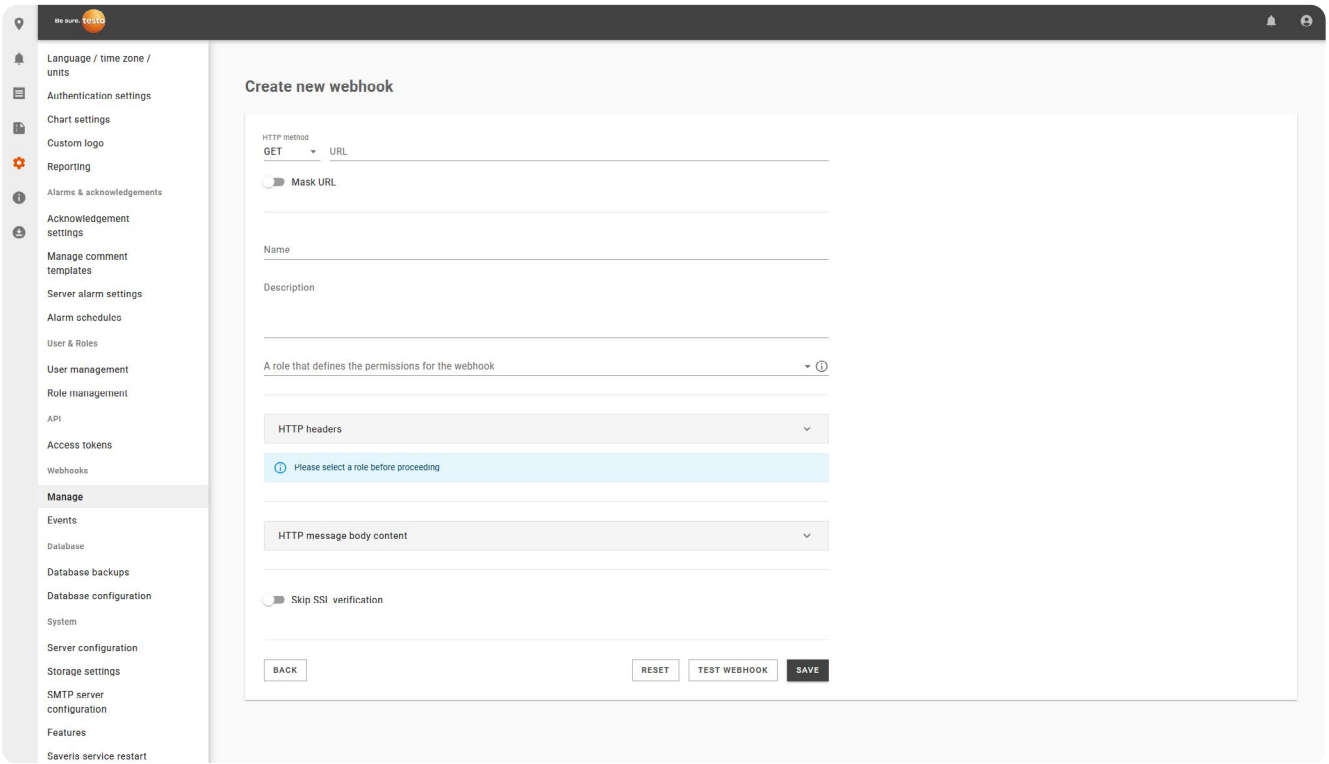


Fig. 3 testo Saveris 1 webhook event creation window.

Use cases

In complex industrial and laboratory environments, data is often distributed across multiple specialized systems, such as:

- **Environmental monitoring systems** (e.g. temperature, humidity, pressure)
- **Inventory management systems** (e.g. product storage conditions, expiry dates)
- **Energy monitoring systems** (e.g. HVAC performance, power usage)
- **Production systems** (e.g. batch data, process parameters)

testo Saveris 1 is designed as a robust environmental monitoring platform that ensures secure and compliant data logging. Through its REST API, it enables authorized users and applications to extract structured measurement data, including time-stamped environmental readings, alarm events, and system metadata. This data can then be integrated into external platforms and tools, enabling higher-level analysis, cross-system correlation, and visualization.
Representative integration use cases include:

- **Climate control optimization:** Temperature and humidity trends exported from testo Saveris 1 can be analyzed alongside HVAC control logic to support demand-based ventilation and improve environmental stability.
- **Energy efficiency tracking:** When combined with production batch data, energy usage metrics (retrieved from energy monitoring systems) can be correlated with testo Saveris 1 conditions to monitor consumption per unit produced.
- **Product quality risk analysis:** Temperature deviations or excursions logged in testo Saveris 1 can be programmatically linked with inventory records to estimate exposure-related product risks, supporting QA decision-making.
- **Process stability assessment:** Historical environmental data from testo Saveris 1 can be merged with process quality parameters to identify environmental influences on defect rates or production variation.
- **Predictive maintenance:** Abnormal environmental patterns (e.g. sudden changes in pressure or temperature) can signal equipment stress or failure, allowing integration with maintenance platforms for early intervention.
- **Data aggregation and reporting:** Exported data can be visualized in platforms such as Power BI or integrated into LIMS systems to support compliance audits, batch traceability, and continuous improvement initiatives.

By offering open access to its data via a secure and documented REST API, testo Saveris 1 empowers organizations to build integrated digital workflows without compromising system integrity or compliance. This supports data transparency, drives efficiency improvements, and enables clients to align environmental monitoring with broader operational intelligence systems.

Best Practices and CFR compliance

Security

Robust security is fundamental when implementing APIs and webhooks in pharmaceutical environments. This includes enforcing encrypted communication via HTTPS, employing strong authentication methods such as API keys or OAuth 2.0, and implementing role-based access control to restrict data access. Additionally, role and user verification ensure that only trusted systems can exchange information, protecting sensitive environmental data from unauthorized access and tampering.

Safety and CFR Compliance

Safety and regulatory compliance under FDA 21 CFR Part 11 are maintained by ensuring all API and webhook interactions are fully auditable, traceable, and attributed to specific users. Electronic records generated through these interfaces are secured against unauthorized changes, and every data transaction is logged with timestamps and unique identifiers. This ensures integrity, accountability, and compliance with regulatory mandates for electronic records and electronic signatures.

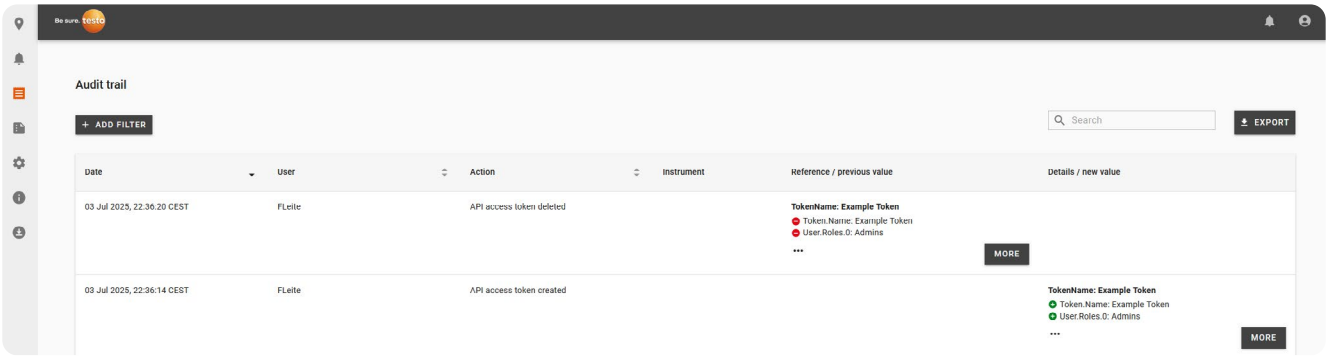


Fig. 4 Saveris 1 Webhook event creation window.

Compliance in Data Transmission

Data transmission compliance requires that all environmental monitoring data exchanged via APIs and webhooks be encrypted in transit and securely logged. Time synchronization between systems is critical for accurate timestamping and retry mechanisms with idempotency prevent data loss or duplication. Together, these measures guarantee that transmitted data remains intact, reliable, and verifiable throughout its lifecycle, meeting stringent regulatory requirements for data integrity and secure communication.

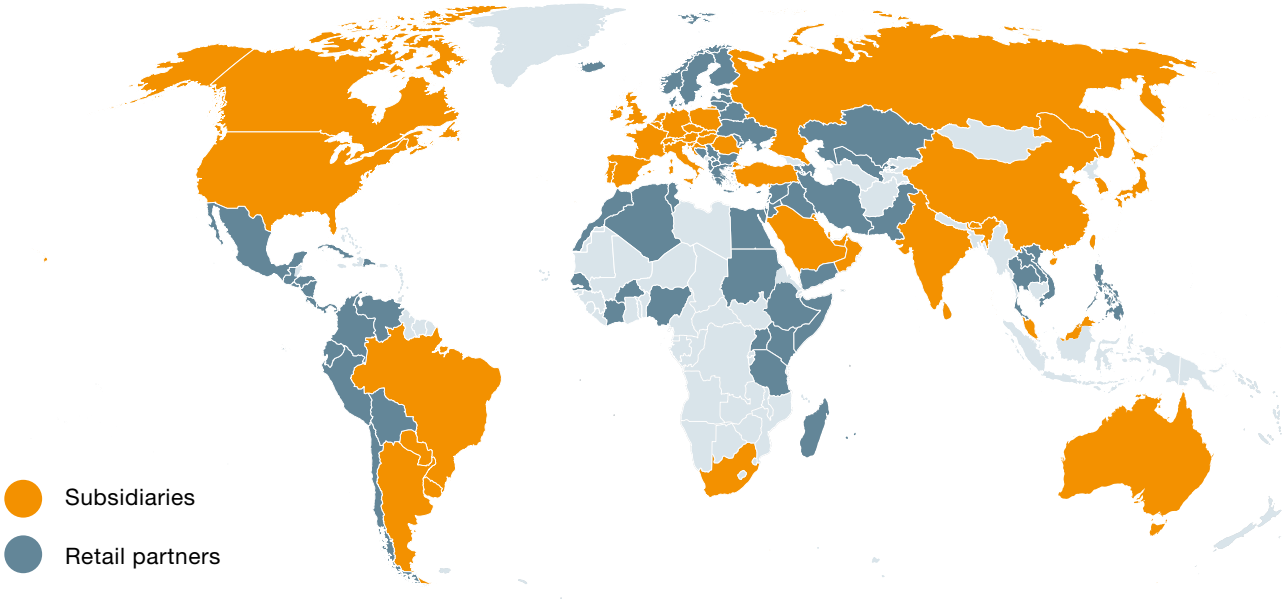
Conclusion

APIs and webhooks are essential tools for enabling efficient, real-time integration within pharmaceutical environmental monitoring systems. testo Saveris 1's API and webhook functionalities provide a practical framework that supports secure, compliant data integration into other systems and event-driven responsiveness, aligning with regulatory requirements such as **FDA 21 CFR Part 11**.

The architecture facilitates scalable integration with existing systems like LIMS, BMS and QMS, helping organizations reduce manual effort, improve data accuracy, and respond faster to critical events. Implementing testo Saveris 1's capabilities offer a solid foundation for modern, flexible system design that supports operational efficiency and regulatory compliance.
Organizations looking to enhance their environmental monitoring, and digital workflows should consider leveraging these features to build resilient, interoperable solutions that meet current and future demands.

Be sure. testo

High-tech solutions from southern Germany.



For over 60 years, Testo has been known for creating innovative measuring solutions made in Germany. As a world market leader in portable and stationary measuring technology, we help our customers to save time and resources, protect the environment and people's health and improve the quality of goods and services.
More than 3700 employees work in research, development, production and marketing for the high-tech company in 37 subsidiaries all over the world. Testo provides more than 1 million customers around the globe with high-precision measuring instruments and innovative solutions for the measurement data management systems of the future.
An average annual growth rate of over 10% since the company's foundation in 1957 and a current turnover of just short of 400 million Euros impressively demonstrate that southern Germany and high-tech systems go hand in hand.
The above-average investments in the future of the company are also a part of Testo's recipe for success.
Testo invests about a tenth of annual turnover in research and development.

Do you have any questions?

Hassellunden 11A, 2765 Smørum
Tel. 45 95 04 10
info@buhl-bonsoe.dk
www.buhl-bonsoe.dk

BUHL BØNSØE
MÅLBAR VIDEN