Secure by Design

A Comprehensive Guide to Building Securely with Xano.

Jacques Antikadjian, Co-Founder & CSO

XANO

Letter from Xano Co-founder and CSO Jacques Antikadjian

As Co-Founder and Chief Security Officer at Xano, I understand firsthand that when building applications – from rapid prototypes to mission-critical enterprise systems - security isn't optional; it's foundational. The consequences of neglecting robust controls, data protection, and reliable backups are simply too high. Regulatory missteps can lead to significant legal and financial penalties. A single data breach can irrevocably damage customer trust built over years. Data loss due to improper configuration(s) can cripple operations and lead to substantial revenue loss. The risks are high.

Yet, clear, actionable guidance for developers and tech leaders on implementing security best practices effectively within modern development workflows is often difficult to find.

This is precisely why we've developed this security guide. At Xano, security is not merely a feature or a department; it is a core principle embedded in our architecture, our processes, and our company culture. My role is to ensure this principle translates into a platform you can trust implicitly. Whether you are just beginning your journey with Xano or scaling a complex, regulated application, this guide is designed to empower you to build with confidence and clarity.

Inside this guide, we demystify essential security concepts and demonstrate how Xano facilitates their implementation, from access controls to proactive monitoring and effective recovery strategies. We firmly believe that adopting rigorous security practices should accelerate, not hinder, your development velocity. That's why Xano is engineered to simplify and integrate security best practices, making them manageable and accessible without bogging down your production pipeline.

Safeguarding your data is Xano's paramount responsibility. This commitment is validated by our adherence to the world's most stringent industry standards and data privacy frameworks. We have invested heavily in achieving and maintaining compliance with: HIPAA, SOC 2 Type 2, ISO 27001, GDPR, and others. These certifications aren't just badges; they represent rigorous, continuous audits and third-party evaluations confirming that our platform meets and often exceeds global compliance benchmarks. This isn't just about our compliance; it's about providing you with a platform built on a verifiable foundation of trust and security.

Please remember, navigating the complexities of security is a journey, not a destination, and you are not alone. Should you encounter challenges or have specific questions as you implement these practices, our dedicated customer support and security teams stand ready to assist. Reach out to our team at support@xano.com.

For more security information, please visit https://www.xano.com/security/ or https://security.xano.com/.

Thank you for downloading this guide and for prioritizing security in your development process. Your commitment reflects the same dedication that drives us at Xano. We are confident this resource will provide invaluable knowledge and support as you build secure, scalable, and successful applications on our platform.

Sincerely,

Jacques Antikadjian

Co-Founder & Chief Security Officer at Xano

Table of contents

Introduction

This ebook covers four key components when setting up a secure application:

→ security controls

→ observation

→ recovery and backup

→ security processes

From setting up security controls at the instance and workspace levels to implementing robust data protection and API security measures, this resource covers essential aspects for safeguarding your applications. Additionally, it explores the various Xano security features that support these protocols.

Though this guide does not need to be followed in order, we recommend you read it in its entirety, as it is a comprehensive walk-through of all the steps you should take to secure your applications. And remember, you can always reach out to https://www.xano.com/contact/ to connect with our dedicated customer support and security teams.

Now let's tackle the first topic to explore when securing your application: Security controls.

Security Controls – Locking Down Your Foundation

Instance Security

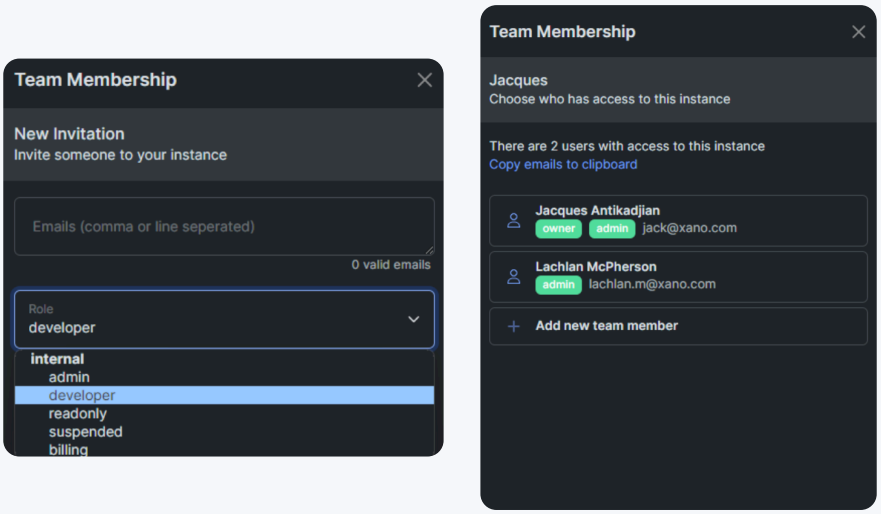
Team Management

The Team Management feature is essential for maintaining a secure, efficient, and scalable development environment. It ensures that team collaboration is seamless while protecting sensitive information and maintaining robust oversight of user activities.

It defines who can access and make changes to your Xano environment. Properly managing user roles and permissions, making sure that only authorized individuals can interact with sensitive data and critical functionalities.

Instance security →

Manage team members →



Role Type

Admin
Full access to all resources

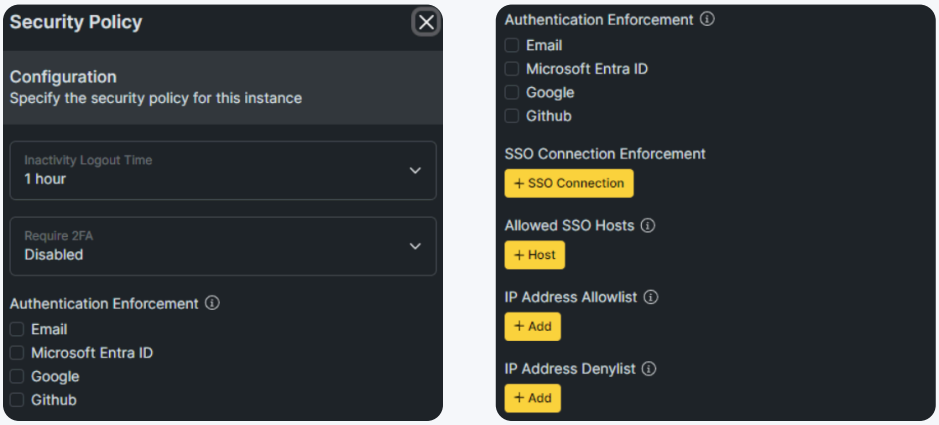
Developer
Edit APIs, view data, limited delete rights

Guest/Agency
Limited workspace access

Analyst
View-only on data and logs Guest/Agency

Security Policy

This policy serves as a guideline for maintaining security standards. By defining and enforcing a security policy, you can systematically manage risks and establish a culture of security awareness within your team.



Inactivity logout timers
Set minimum requirements for password strength and regular updates.

2FA across accounts
Add an extra layer of security by requiring a second form of verification.

Authentication Enforcement

Requires the user to sign in with one of the following services (email, Microsoft Entra ID, Google or Github)

IP Whitelisting

Restrict access to your Xano instance from specific IP addresses or ranges.

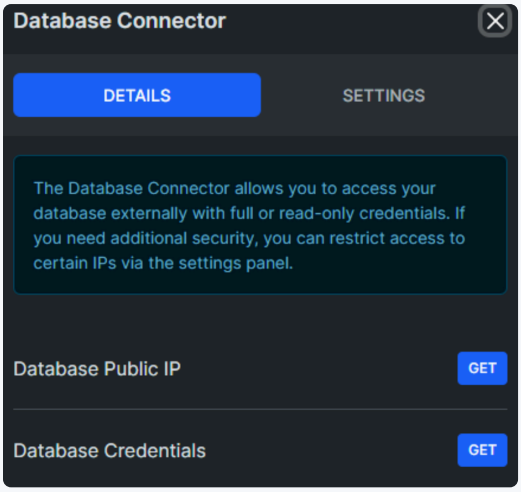
IP Denylist

Block access to your Xano instance from specific IP addresses or ranges.

Xano's 2FA →

Database Connector

The Database Connector is an important component that enables communication between your Xano database and any external tools you want to provide with access.



Making sure that this connector, when enabled, is IP restricted is important for securing database access.

IP restrictions limit access to trusted sources, reducing the risk of unauthorized connections and potential data breaches.

Watch walkthrough →

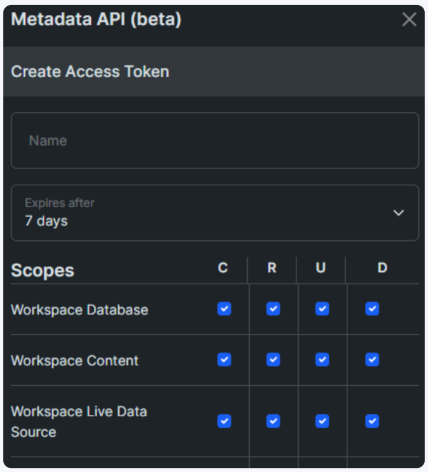
Read about the database connector →

Metadata API Keys

Metadata API keys are used to access and manage system metadata, including schema information and configurations. Keeping track of what keys have been created and making sure their secure management is important for protecting this sensitive information.

Mismanaged or exposed API keys can lead to unauthorized access and manipulation of your system's metadata, potentially compromising the integrity and security of your entire backend. Regularly reviewing and rotating these keys, along with implementing strict access controls, helps maintain the security of your Xano environment.

Read about Metadata API →

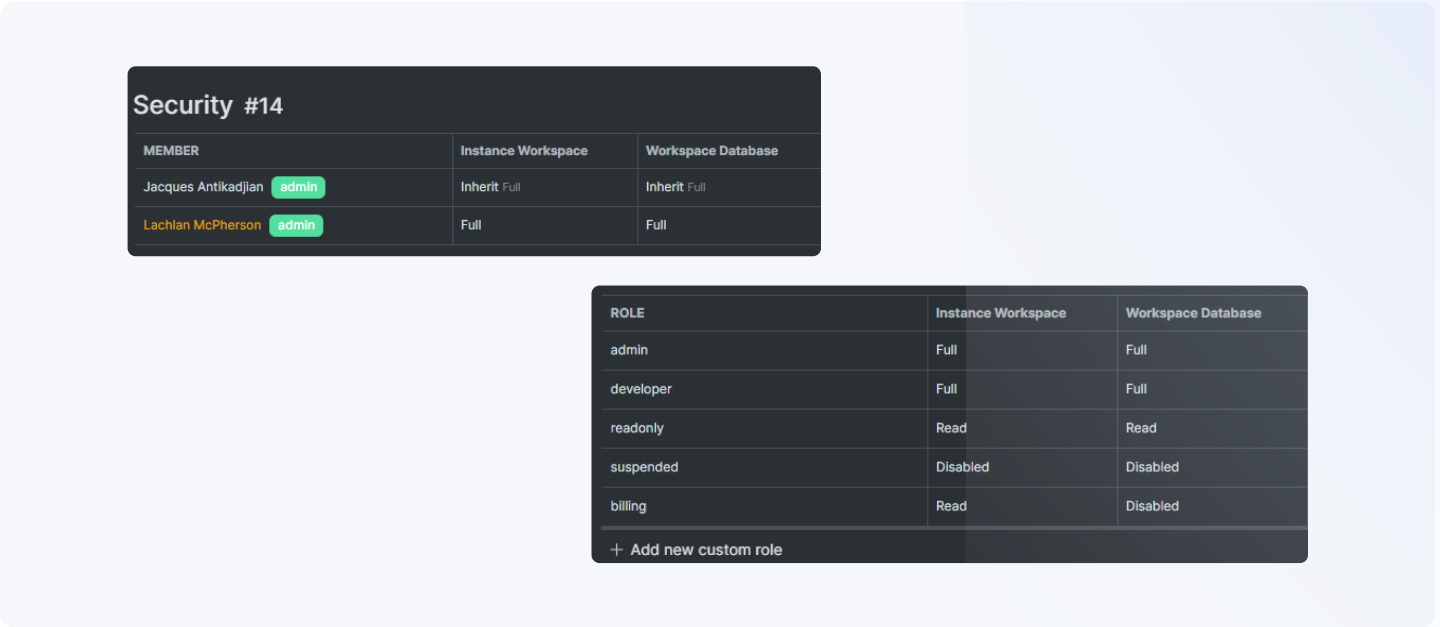


Workspace Security

Learn about editing logic →

Role Based Access Control (RBAC) Control

Apply granular permissions for each team member and workspace within an instance using role-based access control. To enhance security and compliance, make sure that access to resources within your Xano environment is appropriately restricted based on user roles.



Implementing Role-Based Access Control (RBAC) involves identifying and defining the necessary roles within your organization. Each role is then assigned specific permissions that are essential for its functions.

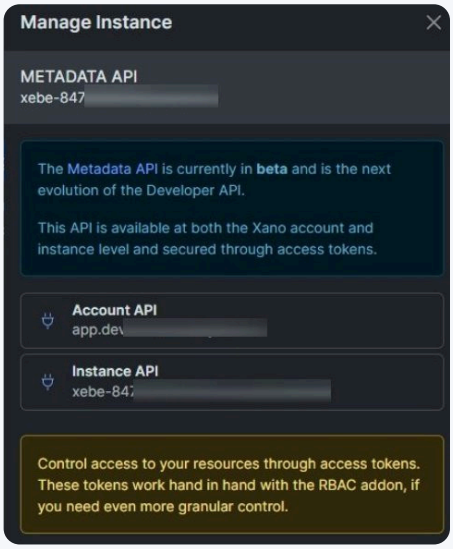
This approach, adhering to the principle of least privilege (PoLP), ensures that only authorized individuals have access to the data and operations they require. By limiting access in this way, RBAC minimizes the risk of unauthorized access to sensitive information and helps organizations maintain compliance with relevant regulations.

Learn about permissions →

Metadata API Access

Xano's Metadata API offers programmatic access to the structure and configuration of your backend. This includes details like your database schema, API endpoints, functions, and other essential elements. By utilizing this API, developers can dynamically retrieve and manage this information, enabling automated interactions with your Xano backend.

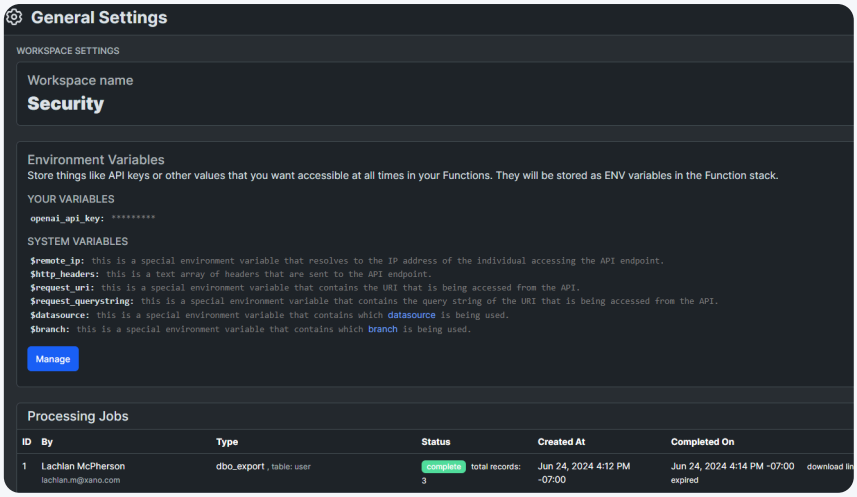
Securing the Metadata API through Role-Based Access Control (RBAC) is crucial. RBAC ensures that only authorized users with the necessary roles and permissions can access and manage metadata. This restriction safeguards sensitive system information, such as schema configurations, from unauthorized modifications and potential security breaches. By securing access to the Metadata API, you maintain the integrity and confidentiality of your system's core configurations.



Read about Metadata API →

Environment Variables

Environment variables are crucial for securely storing sensitive data like API keys, tokens, and database credentials. These variables are accessible across your Xano application, including Queries, Addons, Functions, and Tasks



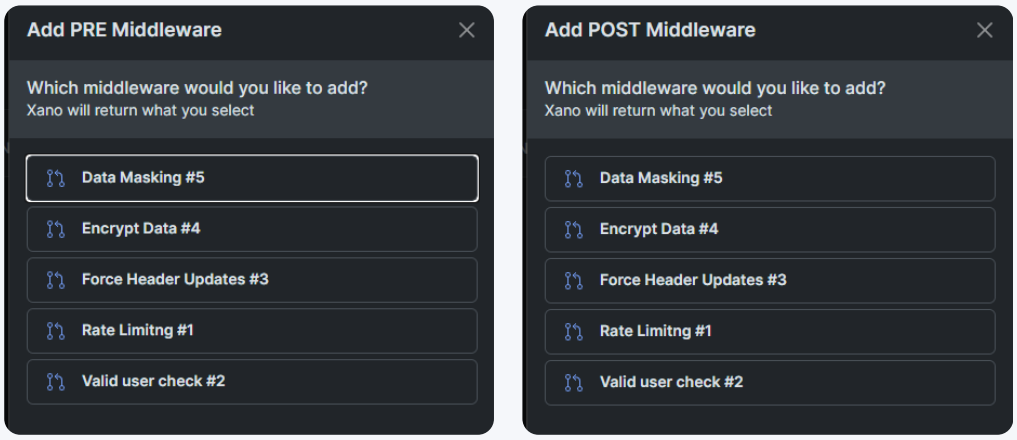
To protect these critical assets, it's essential to implement robust security measures. This includes encrypting stored values, restricting access to authorized personnel, and ensuring that only the necessary components within your application can utilize these variables.

By securely managing environment variables, you safeguard sensitive information from exposure and enhance the overall security of your Xano environment.

Read about environment variables →

Middleware

Middleware acts as an intermediary, handling requests and responses before and after they reach their intended destinations (functions and APIs). This crucial role enables the implementation of essential functionalities such as enhanced security measures, data validation, and custom business logic.



Pre-processing

Middleware can effectively handle crucial tasks such as input validation, authentication, and rate limiting. This proactive approach helps to prevent malicious inputs and unauthorized access, enhancing the security and stability of your application.

Post-processing

Middleware can be effectively utilized for tasks such as logging, response filtering, and error handling. This ensures that API responses are secure, compliant with business rules, and adhere to desired standards. You significantly enhance the security, reliability, and overall functionality of your API endpoints, leading to robust and efficient application performance.

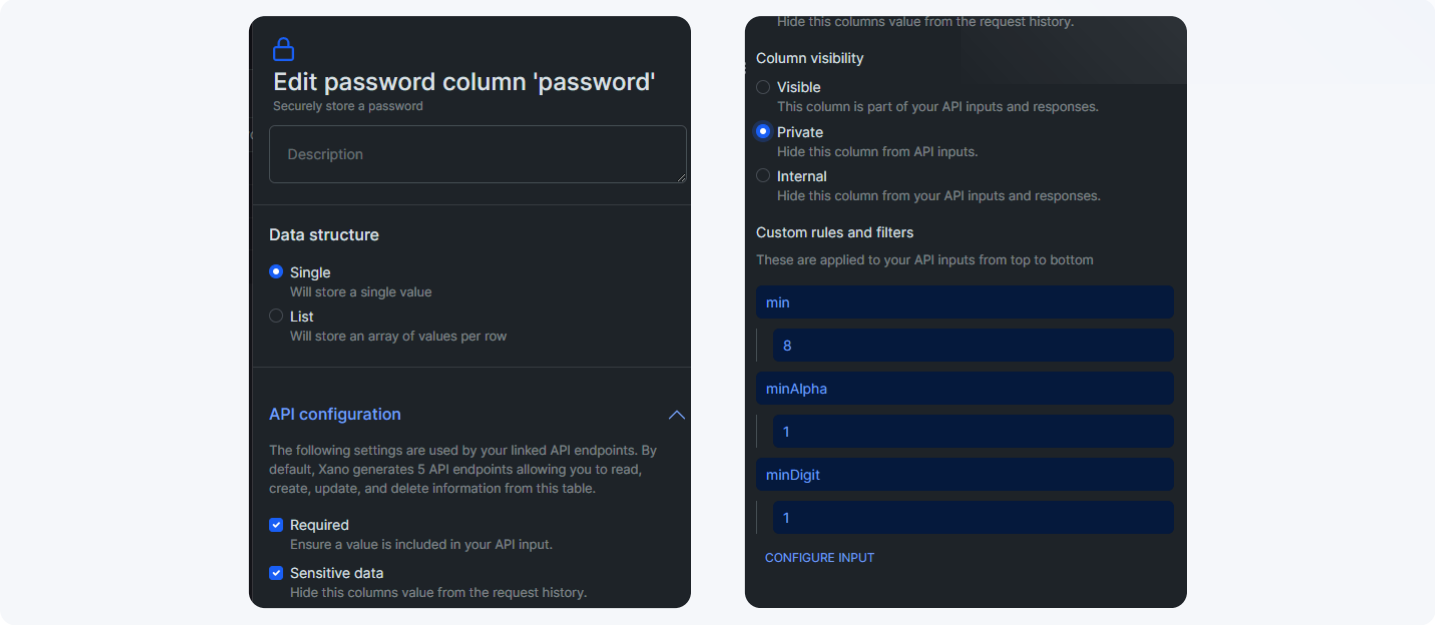
Read more middleware →

Data protection

Sensitive Field Types

Sensitive Field Types in Xano enhance data security by enabling the implementation of extra security measures for critical information like Personally Identifiable Information (PII) and financial details.

These measures may include encryption and restricted access controls.
Designating fields as sensitive is crucial for protecting user privacy, maintaining data integrity, and ensuring compliance with data protection regulations. This practice enforces stricter handling policies, minimizing the risk of unauthorized access and data breaches, ultimately strengthening the overall security of your application.

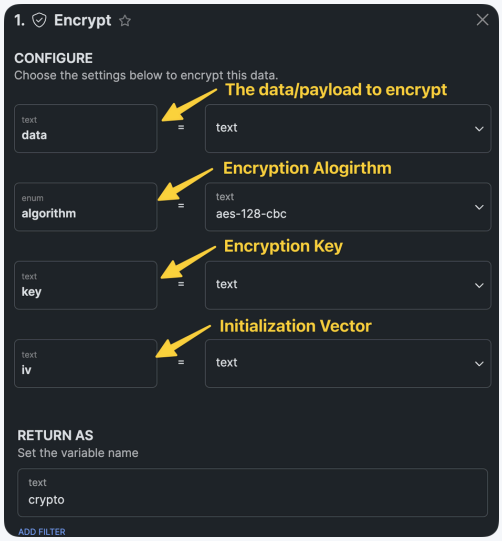


Read more about sensitive data flagging →

Encryption

Encryption is a fundamental security feature within Xano that safeguards data both while stored (at rest) and during transmission (in transit). By encrypting data, you render it unreadable to unauthorized individuals, even if it is intercepted or accessed improperly.

This crucial security layer protects sensitive information, maintains data integrity, and ensures compliance with industry standards and regulatory requirements.



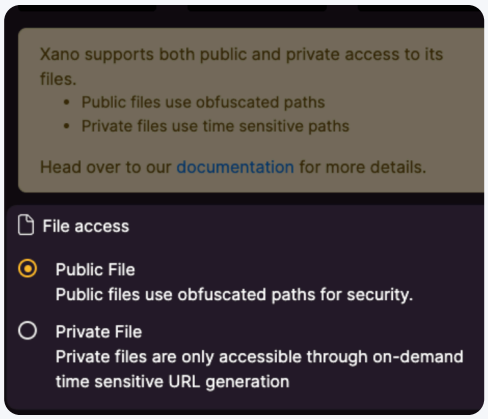
Implementing robust encryption practices within Xano provides a strong defense against potential security threats and breaches, safeguarding the confidentiality and security of user information.

Introduction to security functions and filters in Xano →

Xano's data security features, including encryption →

Private File Storage

Private File Storage in Xano enhances data security by requiring authorized, signed URLs for access. These URLs have a limited viewing period, ensuring that files are only accessible for a specific duration.



This mechanism restricts access to sensitive documents, images, and other media, particularly when dealing with confidential or proprietary information. By implementing Private File Storage, you gain tight control over access to these files, minimizing the risk of unauthorized access and misuse.

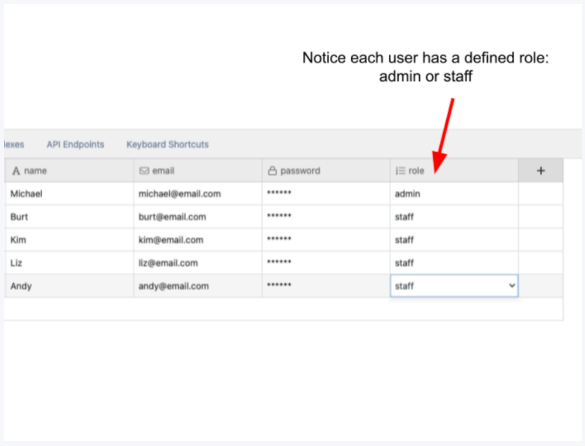
Since files in Private Storage are inaccessible until a new URL is generated within your function stack, you have greater control over who can access what and for how long.

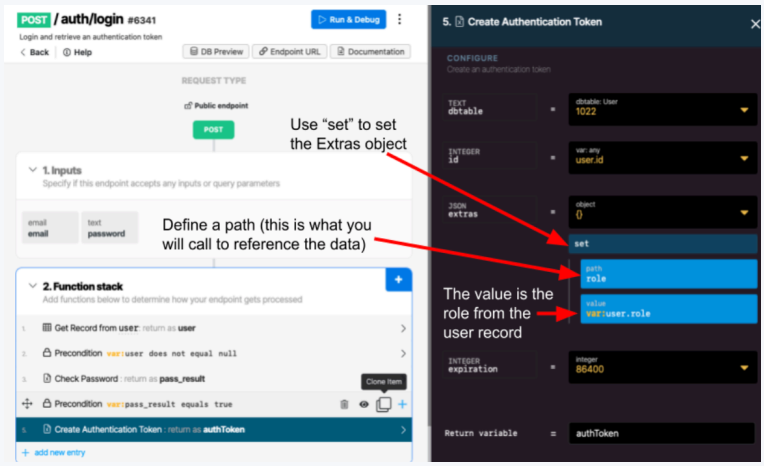
File storage and management in Xano →

User Permissions

Data security is paramount in backend development, particularly in multi-tenant systems where it's crucial to ensure that users can only access data relevant to them, even when shared databases are used.

This requires careful data segregation and access restriction. Assigning appropriate roles to users and defining relationships between them is essential to ensure that only authorized individuals can view and interact with specific data.





Role-Based Access Control (RBAC) is a key mechanism for implementing these restrictions. By defining roles and assigning users to them, we can control access to data and API endpoints based on their permissions. This guide will explore two distinct approaches to enforcing RBAC within API endpoints.

Read more about RBAC →

Restricting Access (RBAC) →

API Security

Secure your API endpoints in Xano →

Authentication on API Requests →

Authentication

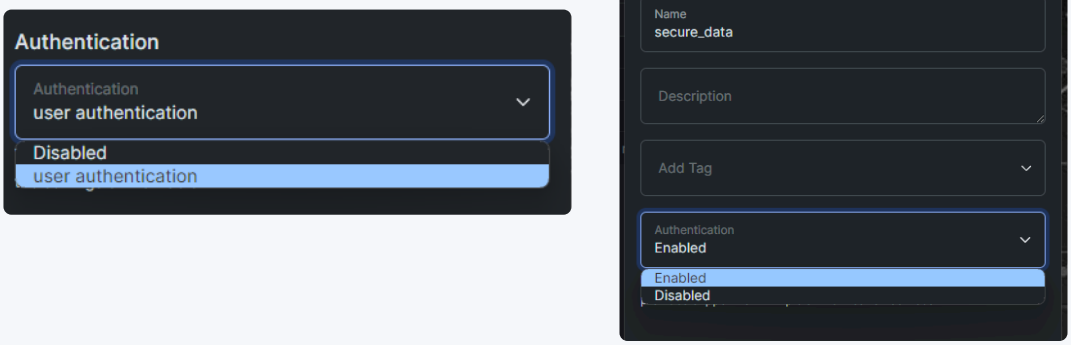
Authentication is fundamental to securing your APIs in Xano. By utilizing Xano's native authentication system with JSON Web Encryption (JWE) tokens or by developing a custom authentication solution, you can effectively restrict access to your API endpoints, ensuring that only authorized users can interact with your application.

Enable Authentication for a Table

Authentication can be applied to individual tables within your database. While typically applied to the primary user table, authentication requirements can be extended to other tables as needed. For example, you might encounter situations where user accounts are distributed across multiple tables due to legacy systems or the need for separate tables for customer and staff accounts.

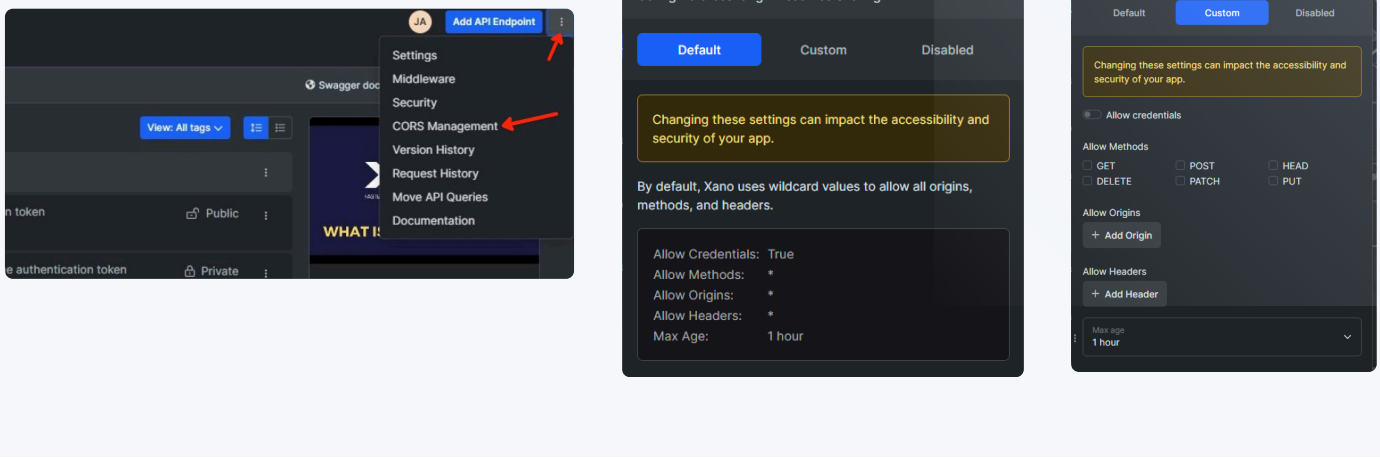
Authentication can be enforced at the individual API endpoint level

If you select a login method during the Jumpstart process, Xano will automatically create an 'Authentication' API group. This group includes pre-built endpoints for user signup and login. These endpoints facilitate user authentication within your application.



Cross-Origin Resource Sharing (CORS)

CORS is a web browser security feature that manages how web pages from one domain can request resources from another domain. It allows controlled access to resources located outside the current domain.



Browsers may send a 'preflight request' using the OPTIONS method to your API endpoints to obtain the CORS parameters. In Xano, you can define CORS options for each API group if the default settings are insufficient.

To modify CORS options, access the CORS Management panel by clicking the icon in the top-right corner of the desired API group.

Read more about CORS management for API groups →

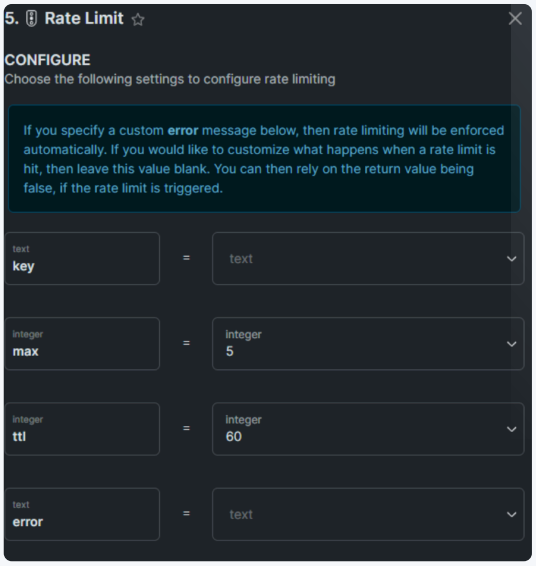
Rate Limiting

Rate limiting restricts the number of requests a user or client can make within a specified timeframe.

It is an effective strategy for preventing abuse and safeguarding your application against denial-of-service (DoS) attacks. It promotes fair API usage, reduces the risk of server overload, and improves the overall stability and performance of your application. Additionally, it aids in detecting and blocking malicious actors attempting to overwhelm your API with excessive requests.

Xano enables you to configure rate limits on your queries, allowing you to control the number of requests an API endpoint can handle within a defined time frame.

Learn more about data caching with Xano →



The Rate Limit function includes several configurable settings:

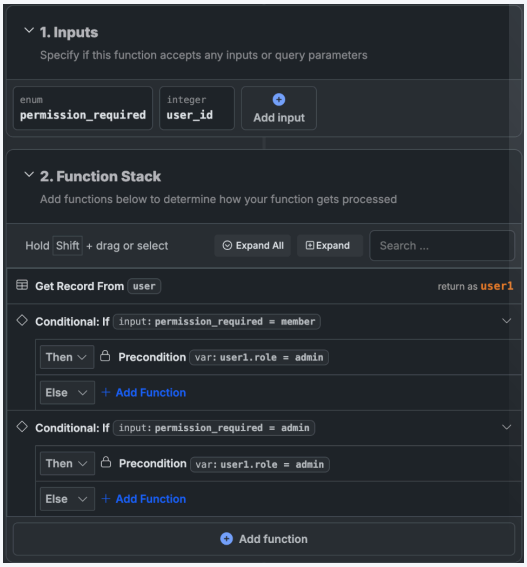
- **Key:** Define a key for the rate limit.
- **Max:** Set the maximum number of requests allowed in the specified time (TTL).
- **TTL:** Set the time to live, in seconds, for each cycle.
- **Error Message (Optional)**: If specified, this message will be displayed when the rate limit is exceeded. If no message is provided, the rate limit will return a boolean value of false, enabling the implementation of custom logic upon reaching the limit.

Read about rate limits in Xano →

User Permissions

Ensuring data security requires implementing proper user permissions and access controls. While Xano's authentication system verifies user validity, it does not inherently manage role-based access or relational permissions. .

Ensuring data security requires implementing proper user permissions and access controls. While Xano's authentication system verifies user validity, it does not inherently manage role-based access or relational permissions. .



To address this:

1. Implement a custom permission check function to validate user roles (e.g., User, Admin, Super Admin).

2. For all database queries, include conditions that reference the authenticated user's ID or related entities (e.g., company).

3. Set up appropriate data relationships to link users with the information they are authorized to access.

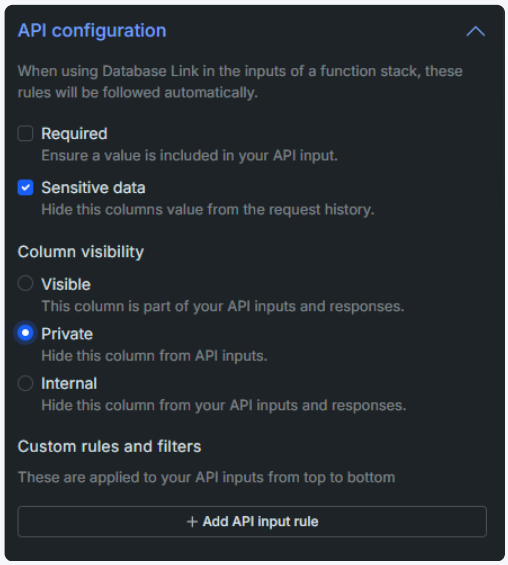
These steps ensure that users can only view and interact with data for which they have proper authorization, maintaining the security and integrity of your application.

Walkthrough calling authenticated endpoints →

Sensitive Data

This practice protects critical data such as PII, financial details, and other confidential information from being accessed by unauthorized individuals.

Implementing data masking helps in maintaining user privacy, complying with data protection regulations, and reducing the risk of data leaks. It is an essential measure for enhancing the security of data handling and ensuring that sensitive information is adequately protected. Data masking involves hiding sensitive data in both request histories and outgoing requests, making sure that sensitive information is not exposed inadvertently.



When a database field is marked as sensitive:

- The value of the field is automatically masked (e.g., replaced with *** or hidden) in the request history logs.
- This ensures that sensitive data such as personally identifiable information (PII), passwords, or credit card details do not appear in logs where they might be exposed to unauthorized viewers.

Read more about Sensitive Data flagging →

Websocket Security

Realtime can be defined as anything that enables your application to provide live updates inside your application. Behind the scenes, Realtime connections are powered by a Websocket server to maintain connections.

Read more about how Realtime works in Xano →

Review these permission examples →

Authentication

Our websocket authentication leverages the same JWE token that the API layer utilizes.

Permissions for channel access can be categorized as follows:

Anonymous Clients – can connect but cannot send messages.

Presence – allows all clients to see each other, with authenticated clients receiving additional details.

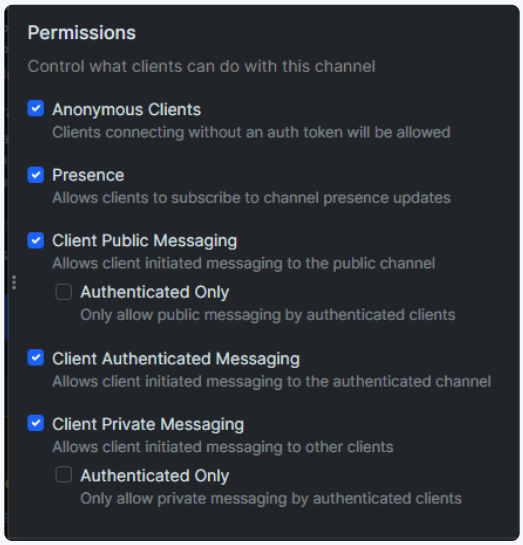
Client Private Messaging allows direct messages between any clients present in the channel.

Client Public Messaging – permits all clients to send messages to the channel, while **Authenticated Only Messaging** – restricts this to authenticated users.

Client Authenticated Messaging – makes sure only authenticated clients receive messages, although anyone can connect.

Authenticated Only Private Messaging restricts this to authenticated users.

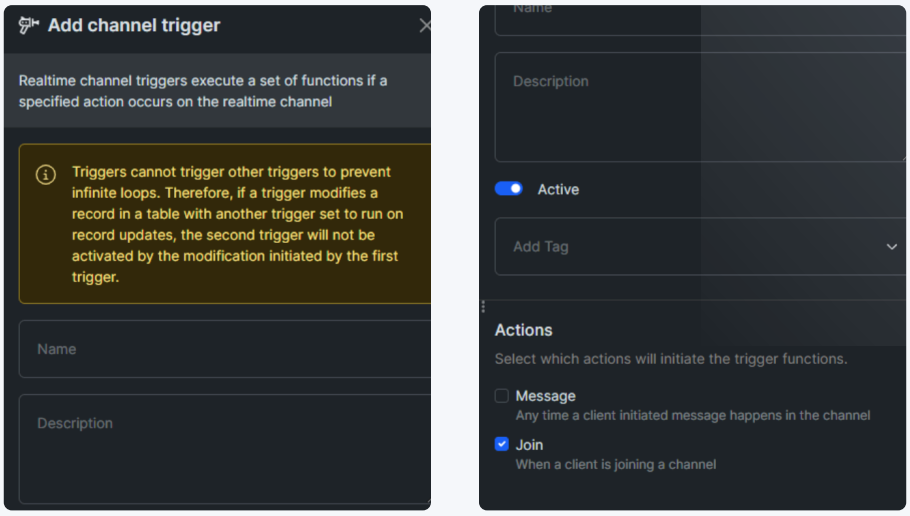
These permissions operate independently despite potential logical dependencies.



Join Triggers

Join triggers enhance the security of websocket channels by adding custom authentication and permission checks when a user attempts to join a channel.

By leveraging join triggers, you can implement complex access control logic, ensuring that only users with the appropriate permissions can join specific channels.



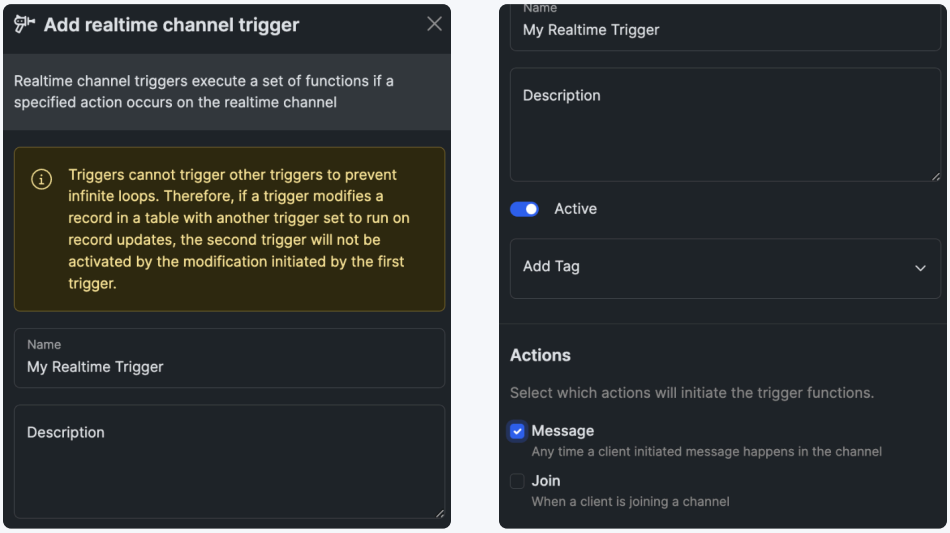
This helps in preventing unauthorized access and maintaining the integrity of the communication channels. Join triggers provide a flexible and powerful way to enforce security policies dynamically, protecting your application from unauthorized users.

Note: Realtime channel triggers execute a set of functions if a specified action occurs on the channel.

Message Triggers

Like join triggers, message triggers provide you the option to create custom logic via the function stack whenever a message is sent over a websocket channel to decide what happens when a message is sent.

Do you need to mask information? Restrict message delivery? This capability is important to enforce security and data integrity in real-time communications.



With message triggers, you can implement actions such as data masking, content filtering, and delivery restrictions based on the sender's permissions and the content of the message.

This ensures that sensitive information is not exposed and that only authorized messages are delivered to recipients. Message triggers provide a granular level of control over websocket communications, enhancing the overall security and functionality of your application.

Read more about Xano triggers →

Observation

Monitoring & Logging

Instance activity

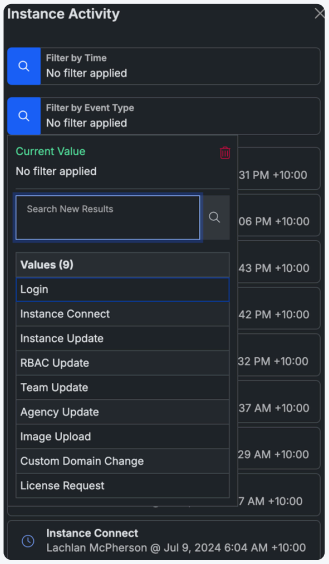
The Instance Activity log, accessible through the Instance settings panel, provides a comprehensive record of various system events and changes. This log includes:

User Actions:
- Logins
- Connections

Resource Management:
- Image uploads
- Custom domain changes

System Updates:
- Instance modifications
- Role-Based Access Control (RBAC) changes
- Team and Agency updates

Administrative Tasks:
- License requests



This log enables administrators to track and review key actions and changes across the instance.

Example of an instance activity in Xano →

Compliance Center

The Compliance Center provides essential tools for monitoring and auditing each workspace:

Middleware and Trigger Overview
- Offers a comprehensive snapshot of middleware and trigger implementations across the workspace.
- Enables you to review the current architecture and security measures in place.

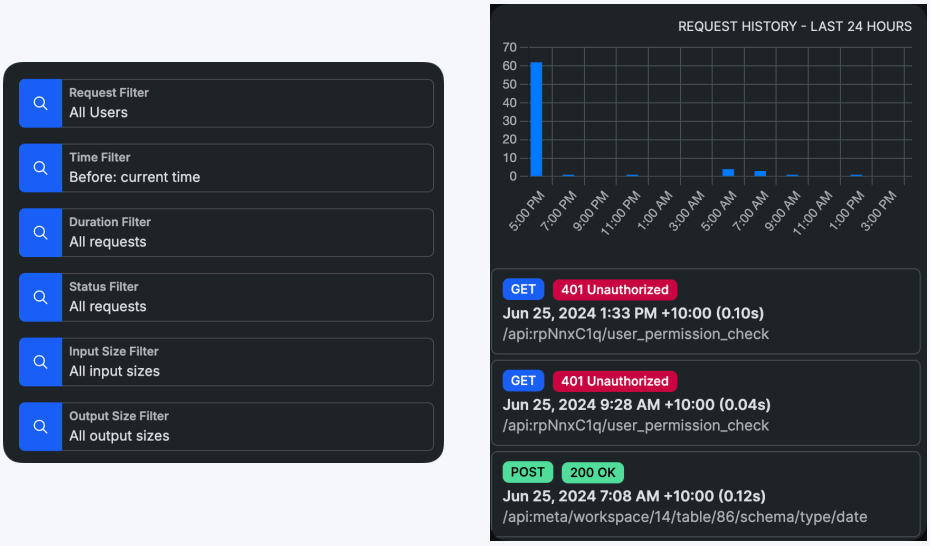
Version History Tracking:
- Records all changes made within the workspace.
- Serves as a detailed audit log for the entire workspace.

These features allow you to maintain oversight of your workspace configuration, security implementations, and change history, supporting compliance efforts and effective system management.

Read about the Compliance Center →

API Request History

API Request history is automatically tracked and stored for 24 hours; this tracks all requests from both the custom-created endpoints in Xano and requests made via the Metadata API. You are able to filter these requests by User, Duration, Time, Status, and Input & Output size.

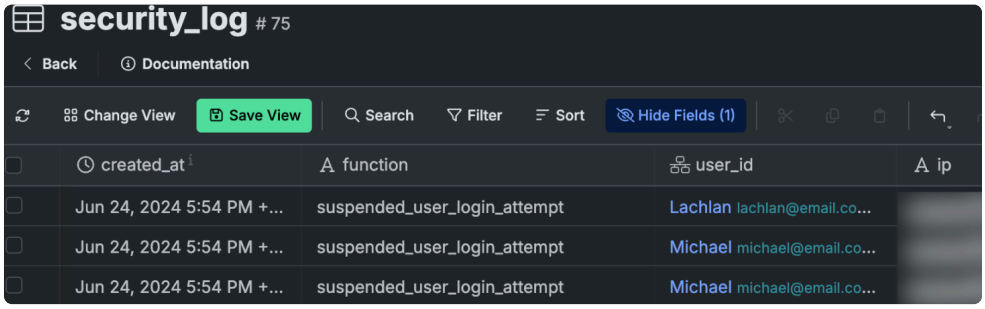


Request history is also available via a Metadata API endpoint, allowing for the automation of storing request history for extended periods beyond 24 hours if required. This requires you to set up a custom workflow.

Learn about API request history →

Custom Logging

Xano allows you to create your own custom security logs in the database. For example, you might have a suspended user validation check implemented via middleware. Each time a suspended user attempts to access the system you can create a record in your database.



Watch how to setup custom logging →

External Monitoring

External monitoring tools provide real-time insights into errors, performance bottlenecks, and security threats, helping you proactively address issues.

Enhance security and performance by integrating these popular monitoring tools with Xano:

PostHog SENTRY mixpanel DATADOG

PostHog
Analyze user behavior and detect suspicious activity

Sentry
Track and fix errors with detailed insights

New Relic
Monitor app performance and detect anomalies

Datadog
Correlate logs, metrics, and security signals

Connecting these tools takes some configuration, but we're actively working on native integrations to simplify the setup process.

Walkthrough on how to set up external testing →

Recovery and Backup

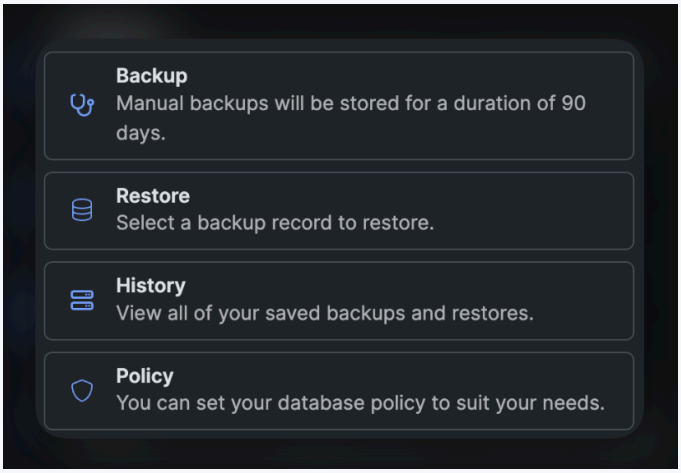
Backups

Instance activity

Automatic Daily Backups:
- Frequency: Every 24 hours
- Retention: 3 days by default
- Configurable backup time via policy settings

Manual Backups:
- User-initiated
- Extended retention: 90 days

Enhanced Enterprise Options:
- Improved backups up to point-in-time restore available
- Supports stricter compliance and disaster recovery requirements



Workspace

Xano provides two methods for workspace data management:

UI-Based Export:
- Exports configuration, database content (JSON), and optional media files

Metadata API Functions:
- Export schema (tables and branches)
- Export entire workspace with content
- Import schema into a new branch
- (optional: set live)
- Import and replace entire workspace

Metadata API Functions:

- Export schema (tables and branches)
- Export entire workspace with content
- Import schema into a new branch
- (optional: set live)
- Import and replace entire workspace

Watch a walkthrough of Xano's Metadata API →

Database

Xano provides two methods for managing individual table data:

In-App UI:

- Export table data as CSV files
- Import CSV files into tables
- Ideal for quick, manual data operations

Metadata API:

- Export / Import table data in JSON format
- Enables automated backup processes
- Suitable for regular backups and integrations

These options offer flexibility for various data management needs, from simple data transfers to automated backup routines.

Function Stack

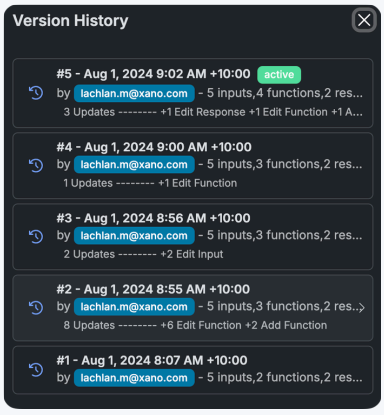
Function stacks, used across APIs, Functions, Tasks, Triggers, Add-ons, and Middleware, feature a robust two-tier versioning system:

Change Tracking:

- Every individual change is logged
- Users can revert to any previous state

Version History:

- Users can publish the function stack, creating a new version
- Published versions serve as major checkpoints
- Any published version can be restored



This system provides granular control over changes and enables easy rollbacks, supporting both quick fixes and significant revisions across Xano's various components.

Watch the Function Stack walkthrough →

Read more about backups →

Security Processes

Security Processes to Perform

Read more about access controls →

Review and Update Access Controls

→ Use Role-Based Access Control (RBAC) to assign the least privilege necessary for each user.

→ Immediately revoke access for users who no longer require it, such as departing team members

→ Periodically audit user roles and permissions in Xano.

Use Secure Authentication Practices

→ Enable two-factor authentication (2FA) for all users accessing the Xano environment.

→ Rotate passwords and API keys regularly and avoid sharing them.

Implement Data Backup and Recovery Plans

→ Schedule automated backups of your data and workflows.

→ Periodically test data restoration to verify that backups are functional and up-to-date.

Secure File Storage

→ Use Xano's private file storage option for sensitive or critical files.

→ Regularly audit stored files to remove unnecessary or outdated items.

Educate and Train Team Members

→ Train your team on secure practices, including recognizing phishing attempts and avoiding unsafe connections.

→ Share guidelines on securely managing sensitive data within Xano.

Use a Secure Network

→ Access the Xano environment through secure networks, such as a VPN.

→ Avoid accessing Xano from unsecured public Wi-Fi networks..

Xano’s Commitment to Security

Security Processes to Perform

Xano holds a comprehensive set of global certifications, attestations, regulations, and recognized frameworks confirming our unwavering dedication to data security, privacy, a nd compliance.

Read more in our Security Center →

Security Standards & Regulations

Xano is a robust backend platform that streamlines the creation, management, and protection of data-driven applications. It involves multiple layers of protection, starting with rigorous physical security measures at data centers and extending to encryption of data both in transit and at rest. Xano follows strict access controls, conducts regular vulnerability assessments, and applies continuous monitoring to protect against emerging cyber threats.

Additionally, Xano’s internal policies, employee training programs, and security audits align with international standards and regulatory requirements. Third-party validation through independent audits and regular reviews helps validate operational effectiveness and data protection capabilities. This disciplined approach demonstrates our commitment to maintaining the confidentiality, integrity, and availability of client data.

Our dedicated customer support and security teams stand ready to assist. Reach out to our team at https://www.xano.com/security/.