nexl

NEXL API'S

FREQUENTLY ASKED QUESTIONS

September 27, 2024

Nexl APIs - Frequently Asked Questions

What does API stand for?

Application Programming Interface

How do they work?

An application programming interface (API) is a set of routines, protocols, and tools for connecting software applications

What technology do you use for APIs?

GraphQL is a query language for APIs and a runtime for executing those queries using a type system you define for your data. Unlike the more traditional REST API, GraphQL allows clients to request the needed data, making it possible to fetch data from multiple sources with a single query. This reduces the amount of data transferred over the network and improves the efficiency of data retrieval.

What is API documentation?

API documentation is a complete, accurate technical writing giving instructions on how to effectively use and integrate with an API. It is a compact reference manual that has all the information needed to work with the API

Do you have API documentation?

Yes, you can check it out here.

Can you automate workflows using APIs?

To use Power Automate to interact with your APIs via Nexl's connector, you'll need to follow specific steps to ensure the connector is set up correctly and used effectively in your workflows. Here's how you can do it:

Step 1: Set Up Nexl Connector in Power Automate

Access Power Automate:

- Go to the Power Automate website and sign in with your Microsoft account.

Search for Nexl Connector:

- In Power Automate, navigate to the "Data" section on the left menu and select "Connectors."
- Use the search bar to find the Nexl connector. If it's available as a pre-built connector, you should be able to locate it here.

Add or Configure the Nexl Connector:

- Click on the Nexl connector and select "Add a connection."
- You might need to authenticate using API keys, OAuth tokens, or any credentials required by Nexl to access their API.
- Follow the prompts to enter the necessary authentication details.

Step 2: Create a New Flow Using the Nexl Connector

Start a New Flow:

- Click on "Create" in the left menu and choose the type of flow you want to create (e.g., Automated Cloud Flow, Instant Cloud Flow, Scheduled Cloud Flow).

Choose a Trigger:

- Select a trigger that fits your scenario. For instance, you might want the flow to start when a new record is added in another application, or when a specific event occurs.

Add Nexl Actions:

- Click "New Step" to add an action.
- Search for Nexl in the action list to find actions specific to your Nexl connector.
- Select the appropriate action you want to perform. This could be retrieving data, adding new records, updating information, etc.

Step 3: Configure the Nexl Action

Set Action Parameters:

- Enter the necessary parameters required by the Nexl API. This might include specific IDs, data fields, or query parameters relevant to your API calls.

Use Dynamic Content:

- Utilize Power Automate's dynamic content feature to insert data from previous steps or inputs into your Nexl action. This allows for dynamic and customizable API interactions.

Step 4: Implement Error Handling and Conditions

Error Handling:

- Use parallel branches and the "Configure Run After" feature to handle potential errors gracefully. You can set up alternative actions in case the Nexl API call fails.

Conditional Logic:

- Add conditions to your flow to handle different scenarios based on API responses. For example, you might want to send notifications if a certain condition is met or update a different record based on the response data.

Step 5: Test and Save Your Flow

Save the Flow:

- Click "Save" to store your flow configuration.

Test the Flow:

- Run a test by triggering the flow manually (for instant flows) or allowing it to run based on the defined trigger (for automated/scheduled flows).

Review Run History:

- Check the run history to view details of each execution, including the inputs, outputs, and any errors encountered during the API interactions.

Step 6: Monitor and Maintain the Flow

Monitor Flow Activity:

- Use the “Monitor” section in Power Automate to keep track of flow performance and success rates.

Update Connections:

- Periodically check and update any credentials or connections if your Nexl API credentials change or if the connector is updated.

Tips for Using Nexl Connector

API Documentation: Ensure you have access to Nexl’s API documentation to understand the available endpoints and required parameters.

Authentication: Make sure your authentication tokens or API keys are up to date and stored securely.

Rate Limits: Be aware of any rate limits or quotas imposed by Nexl’s API to avoid potential throttling. To ensure fair usage and maintain the performance of our APIs, we may enforce a rate limit of 100 requests per minute.

By following these steps, you can effectively use Power Automate to interact with your APIs using Nexl’s connector, enabling streamlined and automated processes within your workflows.

What are best practices for error handling:

Graceful Failures: Ensure your application can continue functioning even if an API call fails, and provide meaningful error messages to your users.

Retries with Backoff: Implement retry logic with exponential backoff for transient errors (e.g., network issues, 500-level errors).

How can I test our custom integration before launch?

Where possible, we would recommend utilizing a Nexl Sandbox. When developing the integration on your sandbox platform, simply use the Nexl Sandbox API key.

You can also use Postman to efficiently build and test API queries, troubleshoot issues, and automate testing processes to streamline API development and integration. Postman is a powerful tool for building, testing, and debugging API queries. It provides a user-friendly interface to construct requests and view responses. Here's a step-by-step guide on how to use Postman to help build queries in APIs:

Step 1: Install Postman

- Download and Install: Go to the Postman website and download the application for your operating system. Follow the installation instructions.

Step 2: Create a New Request

- Open Postman: Launch the application.
- Create a New Request: Click on the "New" button in the top left corner and select "Request."
- Name Your Request: Give your request a meaningful name and optionally add it to a collection to organize your requests.

Step 3: Configure the Request

- Select the HTTP Method: Choose the appropriate HTTP method (GET, POST, PUT, DELETE, etc.) from the dropdown menu next to the request URL field.
- Enter the Request URL: Type the endpoint URL of the API you want to query.
- Add Headers: If your API requires headers (such as authentication tokens, content type), click on the "Headers" tab and add them.

Step 4: Add Query Parameters

- Query Parameters: If your request requires query parameters, click on the "Params" tab below the request URL field. Add each parameter name and value in the table provided.
- Body Data: For requests like POST or PUT, switch to the "Body" tab and select the appropriate data format (e.g., form-data, raw, JSON). Enter the request body data.

Step 5: Authentication

- Authentication: If your API requires authentication, click on the “Authorization” tab. Select the type of authentication from the dropdown (e.g., Bearer Token, Basic Auth) and provide the necessary credentials or tokens.

Step 6: Send the Request

- Send: Click the “Send” button to submit your request to the API.
- View the Response: The response from the API will appear in the lower section of the window. It includes the status code, response time, headers, and body.

Step 7: Analyze and Debug

- Review the Response: Check the response code to ensure your request was successful (e.g., 200 for success, 404 for not found).
- Error Messages: If there are errors, examine the response body for error messages or use the console (View > Show Postman Console) for more detailed logs.

Step 8: Save and Organize Requests

- Save Requests: Save successful requests to collections for easy access later. Click the “Save” button and add it to an existing collection or create a new one.
- Documentation: Use Postman’s ability to generate documentation for your API endpoints by describing requests and responses.

Step 9: Use Environment Variables

- Environment Variables: Create environments to store variables like base URLs, tokens, or any reusable data. This allows for quick switching between different configurations (e.g., development, testing, production).
- Set Variables: Click on the “Environment Quick Look” button (the eye icon in the top right corner) to set or create new environment variables.
- Use Variables in Requests: Reference variables in your requests using {{variableName}}.

Step 10: Automate Tests

- Write Tests: Use Postman's testing feature to write scripts that automatically check the response. Click on the "Tests" tab and use JavaScript to write test scripts.
- Example Test Script:

Additional Features

- Mock Servers: Simulate API endpoints for testing without affecting the live server.
- Monitor Collections: Set up monitors to run collections at regular intervals and check API health.

What is possible using APIs?

When working with Power BI APIs, it's important to differentiate between the two main types: the Public API accessible via API keys (mostly for embedding scenarios) and the OAuth-based API, which offers more comprehensive capabilities but requires authentication through Azure Active Directory (AAD).

1. **Public API Accessible via API Key (for Embedding)**

   The Public API is primarily used for embedding Power BI reports, dashboards, and tiles into your applications. It's more limited in scope and does not offer the full range of Power BI service capabilities.

   Capabilities:

   Embedding Reports and Dashboards:

   - You can embed Power BI reports, dashboards, and tiles into your web or desktop applications.
   - Common operations include generating an embed token, which allows users to view the embedded content.

   Access to Some Report Information:

   - Retrieve basic metadata about reports, such as report names and URLs.

   Limited Dataset Access:

   - The ability to query datasets using the REST API is typically restricted and often requires OAuth.

   Limited Admin Operations:

   - Most administrative functions, such as dataset refreshes or retrieving usage metrics, require OAuth authentication and are not accessible via API Key.

2. OAuth-based API (More Comprehensive)

The OAuth-based API offers full access to the Power BI service, allowing you to manage and interact with virtually every aspect of your Power BI environment. This requires authentication via Azure Active Directory (AAD) and is intended for more advanced integration and management tasks.

Capabilities:

Dataset Management:

- Refresh Operations: Trigger refreshes of datasets.
- Refresh History: Retrieve the refresh history, including success rates, timestamps, and error messages.
- Data Source Management: Update and manage the data sources for your datasets.

Report and Dashboard Management:

- Report Management: Create, update, delete, and list reports in your workspace.
- Export and Clone Reports: Export a report's content or clone an existing report.
- Dashboard Management: List, create, or delete dashboards.

User and Group Management:

- Manage User Permissions: Add or remove users from workspaces or assign roles.
- Workspace Management: Create, update, or delete workspaces.
- Audit Logs: Retrieve audit logs to track user activities within the Power BI service.

Admin Operations:

- Capacity Management: Monitor and manage Power BI capacities.
- Service Principal Integration: Use service principals for app-only authentication scenarios, particularly useful in automation scripts and enterprise applications.
- Usage Metrics: Retrieve detailed usage metrics for reports, datasets, and dashboards.

Embedding Operations (More Advanced):

- Similar to the Public API but with additional options for controlling how content is embedded, including more granular control over permissions and user experiences.

Automation and Monitoring:

- Automated Deployment: Use the API to automate the deployment of reports and dashboards across different environments (e.g., Dev, Test, Prod).
- Monitoring: Create automated scripts that monitor the health and performance of datasets, including regular checks on refresh success rates.

Comparison Summary:

Public API (API Key):

- Included in Nexl API Documentation
- Mainly for embedding scenarios.
- Limited dataset and admin access.
- Easier to set up but less powerful.

OAuth-based API:

- Excluded from Nexl API Documentation
- Full-featured API with extensive capabilities for managing all aspects of Power BI.
- Requires AAD authentication, which adds complexity but provides greater control.
- Suitable for enterprise-level management, automation, and monitoring.

Use Cases:

Public API (API Key):

- Use this for embedding reports in external applications where you want users to view Power BI content without requiring AAD authentication.

OAuth-based API:

- Ideal for internal tools, enterprise applications, or scripts that need to perform a wide range of Power BI operations, including automation, monitoring, and detailed management of resources.

By choosing the appropriate API based on your needs, you can optimize how you integrate and manage Power BI within your applications or organizational workflows.

Can I use the same API key for multiple integrations?

Yes. You can use the same API Key as many times as desired!

How secure is Nexl's APIs?

Nexl's APIs are designed with robust security measures to protect your data and ensure reliable access. We implement industry-standard security protocols, including:

- Encryption: All data transmitted through our APIs is encrypted using TLS (Transport Layer Security) to protect it from interception and tampering.
- Rate Limiting: To prevent abuse and ensure service availability, we employ rate limiting and throttling strategies.
- Monitoring and Logging: Continuous monitoring and logging help us detect and respond to suspicious activities in real time.
- Regular Security Audits: We perform regular security assessments and audits to identify and mitigate potential vulnerabilities.

These measures help ensure that Nexl's APIs maintain the highest standards of security, providing peace of mind for our users.

How do I get an API key?

You can request an API key directly from your Nexl Client Service Manager

How can I get API support?

You can request API support directly from your Nexl Client Service Manager

1 Sussex Street
Sydney, NSW 2000

nexl.cloud

Nexl is a leading provider of innovative legal technology solutions aimed at creating efficiencies, accelerating growth and providing data-driven insights that drive smart decision making. We offer scalable solutions for law firms of all shapes and sizes Founded by legal professionals with vast experience working within legal firms, Nexl grew out of a desire to fundamentally change the way that legal firms grow and engage with internal and external stakeholders. The existing way of managing these relationships doesn't work. Today, we're driving the delivery of smart, innovative and technology-driven services to clients worldwide. Smart, insight-driven solutions underpin Nexl's purpose. Drawing from our own practical experience working within the professional services industry, our purpose is to develop and deliver technology solutions that will drive positive change in how law firms manage their client relationships and grow their practice.