LexisNexis® RISK SOLUTIONS

TECHNICAL REFERENCE GUIDE

OAuth 2.0 Authentication

For LexisNexis® RiskNarrative®

BIS645-EN_US — 9 September 2026

The recipient of this material (hereinafter "the Material") acknowledges that it contains confidential and proprietary data the disclosure to, or use of which by, third parties will be damaging to LexisNexis Risk Solutions and its affiliated companies (hereinafter "LexisNexis Risk Solutions"). Therefore, recipient agrees to hold the Material in strictest confidence, not to make use of it other than for the purpose for which it is being provided, to release it only to employees requiring such information, and not to release or disclose it to any other party. Upon request, recipient will return the Material together with all copies and modifications, if any, to LexisNexis Risk Solutions.

LexisNexis Risk Solutions shall not be liable for technical or editorial errors or omissions contained herein. The information in this publication is subject to change without notice and is provided "as-is" without a warranty of any kind. Nothing herein should be construed as constituting a warranty, as any applicable warranty is exclusively contained in your signed agreement with LexisNexis Risk Solutions.

All names, addresses, birth dates or other identifying information in the text, or on the sample reports and screens shown in this document, are of fictitious persons and entities and have been created for illustrative purposes only. Any similarity to the name of any real person, school, business, other entity, address, date of birth, or other identifying information is purely coincidental.

Contents

OAuth 2.0 Authentication

OAuth 2.0 authentication enhances the security of your API (application programming interface) requests to LexisNexis® Risk Solutions web services.

Integration in RiskNarrative

In the LexisNexis® RiskNarrative® platform, you can use OAuth 2.0 to securely submit API requests. Optionally, you can disable basic authentication for API interactions.

The OAuth 2.0 integration into RiskNarrative involves specific configuration steps:

- Initial Setup:
  - Set OAuth Permissions on page 4
  - Generate Credentials on page 5
- Using OAuth 2.0 Credentials:
  - Request an Access Token on page 8
  - Submit API Request Using Token on page 10

Set OAuth Permissions

To access OAuth 2.0 options, you configure settings on the RiskNarrative portal to include OAuth permissions for the specific users.

Before you begin this task:
You must be an admin user.

Procedure:

1. Update a role to include OAuth permissions:
   a. Navigate to **Admin** > **Roles**.
   b. For the role that you want to edit, click **Actions**, and select **Edit**.
   c. In the Permissions pane, select **Full** level of access for the Security module.

   d. Click **Save Changes**.
2. Assign the role to the user:
   a. Navigate to **Admin** > **Users**.
   b. For the user that you want to edit, click **Actions**, and select **Edit**.
   c. In the Roles pane, select the role that you updated, and click **Add**.
   d. Click **Save Changes**.

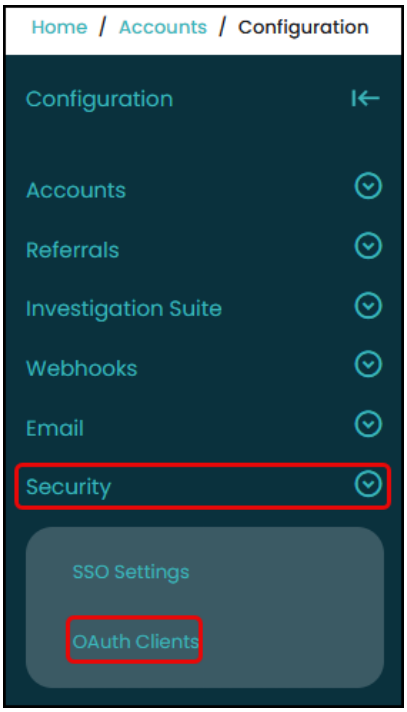
Generate Credentials

You create OAuth credentials that you can use for authentication.

Before you begin this task:
Sign in to the portal as a user with the appropriate permission level. See Set OAuth Permissions on page 4.

Procedure:

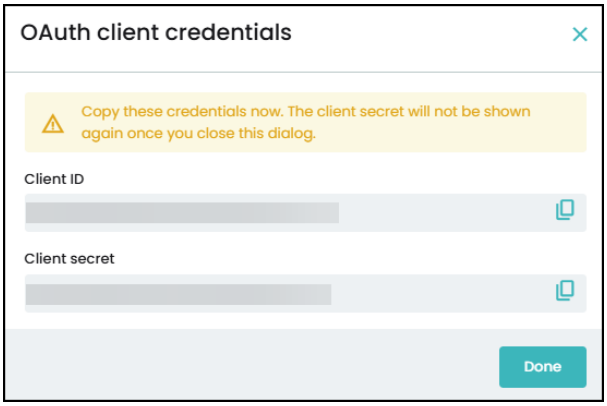
1. Navigate to **Strategy** > **Configuration**.
2. In the Configuration pane on the left, click **Security**, and select **OAuth Clients**.



3. Click **Create an OAuth Client** on the top right.
4. Enter a client name.

   This is a required free-text field to identify and trace the credentials. Maximum length is 200 characters.
5. Click **Next**.

   A pop-window with client credentials is displayed.



6. Copy the client ID and client secret by clicking  (Copy), and securely save the information.

   The client secret is displayed only once. Ensure that you securely store the client secret.

7. Click **Done**.
   The Edit OAuth Client page is displayed.
8. In the Roles pane, select the appropriate role with OAuth permissions, and click **Add**.
9. Click **Save Changes**.
   The client name and client ID are now accessible from the OAuth Clients page.

Edit Credentials

You can edit OAuth credentials.

Procedure:

1. Navigate to **Strategy** > **Configuration** > **Security** > **OAuth Clients**.
2. Click the name of the client that you want to edit.
3. Update the client name and the role, and click **Save Changes**.
   The client credentials are updated.

Delete Credentials

You can delete OAuth credentials.

Procedure:

1. Navigate to **Strategy** > **Configuration** > **Security** > **OAuth Clients**.
2. For the client whose credentials you want to delete, click **Actions**, and select **Delete**.
3. Provide confirmation by clicking **Delete**.
   The credentials are invalidated.

Request an Access Token

You request an access token using your client credentials.

You enter the request in the following format.

```
POST <hostname>/auth/oauth2/token
Authorization: Basic <base64(<Client_ID>:<Client_Secret>)>
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
```

The request consists of the following information.

Get Token Request

Parameter	Description
Method	The method is `POST`.

Host	`<hostname>` is the base URL for your environment or region.
	The following URL values are available:
	• Pre-Production: https://pp.trunarrative.cloud/api
	• Production:
	  • US: https://us.trunarrative.cloud/api
	  • UK: https://uk.trunarrative.cloud/api
	  • EU: https://eu.trunarrative.cloud/api
	  • APAC: https://apac.trunarrative.cloud/api

Endpoint	Use the following endpoint to request a token: `<hostname>/auth/oauth2/token`.
Authorization	Use `Basic` as the authorization type, and the client ID and client secret that you generated on the portal. See Generate Credentials on page 5.
Content-Type	The content type must always be `application/x-www-form-urlencoded`.
Body	The body consists of grant type, which is always `client_credentials`.

Response

If your token request is successful and your credentials are validated, then an access token is returned in the response.

The following example shows a successful response.

```
{
    "access_token": "zdfjhdLL12EKJgjf657jgjhHH76JraWrWssgdw9348dn....",
    "token_type": "Bearer",
    "expires_in": 299
}
```

The response consists of the following information.

Get Token Response

Parameter	Description
Access Token	This is the token that you use when you make API calls to the web service.
Token Type	The type of token is authorization bearer token. The value is `Bearer`.
Expires In	This is the time in seconds that a token is valid from the time that it is issued. After the token expires, the token is considered invalid.

Errors

If the token request was not successful, then an error response is returned.

The following table contains the error codes, the error descriptions, and remediation actions.

Get Token Errors

Code	Description	Comments
`400`	`Invalid grant/request`	The grant type or any other input value is incorrect. Check the input values and submit the request again.
`401`	`Invalid client credentials`	The client ID or client secret is invalid. Regenerate the credentials and submit the request again.
`500`	`Internal Server Error`	Submit the request after some time.

Submit API Request Using Token

You can add OAuth 2.0 authentication to the API requests using your access token.

Use the following endpoint: `<hostname>/public/v1/<API_call_name>`, where `<hostname>` is the base URL for your environment or region (see Request an Access Token on page 8) and `<API_call_name>` is the specific API call (for example, `journey/run-journey` to run a journey, or `monitor/start` to start monitoring.)

To authenticate the API request with OAuth 2.0, add your access token to the HTTP header in the following format, replacing `<token>` with the specific token value.

```
Authorization: Bearer <token>
```

Ensure that the token has not expired. If it has expired, then request the token again (see Request an Access Token on page 8).

API Response

The API response indicates whether OAuth 2.0 authentication was successful.

The following table describes the codes that could be returned as related to authentication success or failure.

API Response

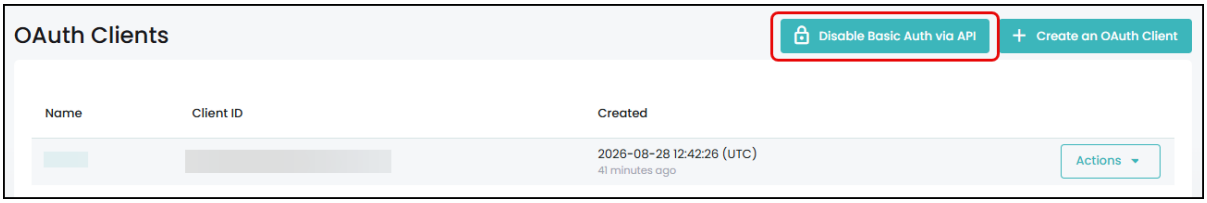
Code	Description	Comments
200	Success	Authentication is successful.
401	Unauthorized	The token is missing, invalid, or expired. Regenerate the token and submit the request again.
403	Forbidden	The token is valid, but the client does not have the access that this endpoint requires. A required role has not been assigned. Assign the appropriate role with OAuth permissions (see Set OAuth Permissions on page 4), and submit the request again.

Disable Basic Authentication

You have the option to disable basic authentication to make sure that API requests are authenticated only through OAuth credentials. When basic authentication is disabled, all API requests that use basic authentication are rejected.

Procedure:

1. On the portal, navigate to **Strategy** > **Configuration**.
2. In the Configuration pane on the left, click **Security**, and select **OAuth Clients**.
3. Click **Disable Basic Auth via API**.



4. Click **Disable Basic Auth via API** to confirm.
   Basic authentication is disabled for the OAuth clients.