Neos Google Analytics Integration

Release dev-master

The Neos Team

Aug 19, 2020

Contents

This is the documentation for the Google Analytics integration into Neos, a free enterprise web content management system licensed under the GPL.

This version of the documentation covering dev-master has been rendered at: Aug 19, 2020

CHAPTER 1

Installation

The package can be installed via composer:

```
composer require neos/googleanalytics
```

CHAPTER 2

Configuration

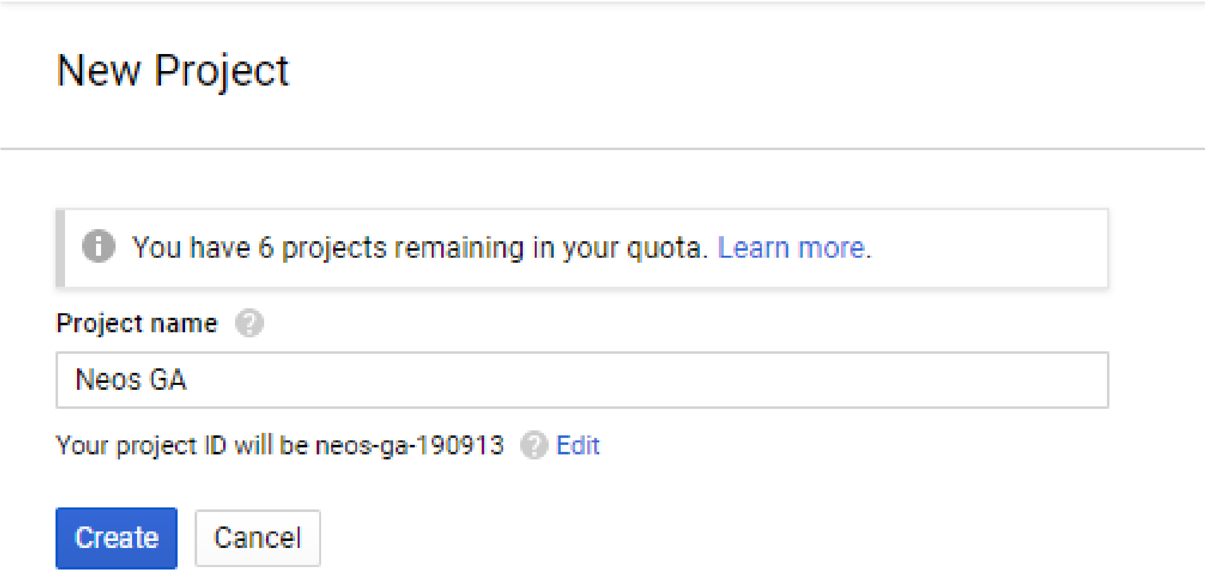
2.1 Statistics display inside Neos

Configuration requires some setup on the side of Google, before the package can be used. In a nutshell:

1. Log in to the Google Developers Console
2. Set up the Consent screen
3. Create a web application Client ID
4. Create a public access Server key
5. Enable the Analytics API
6. Configure the Google Analytics Neos integration using the created credentials
7. Connect Neos instance and Google Account

The steps in detail

Go to the Google Developers Console and create a project. This is needed to create the API credentials and keys for the Analytics integration. The project name can be chosen freely and doesn't really matter, as you will probably be the only one to ever see it.



New Project

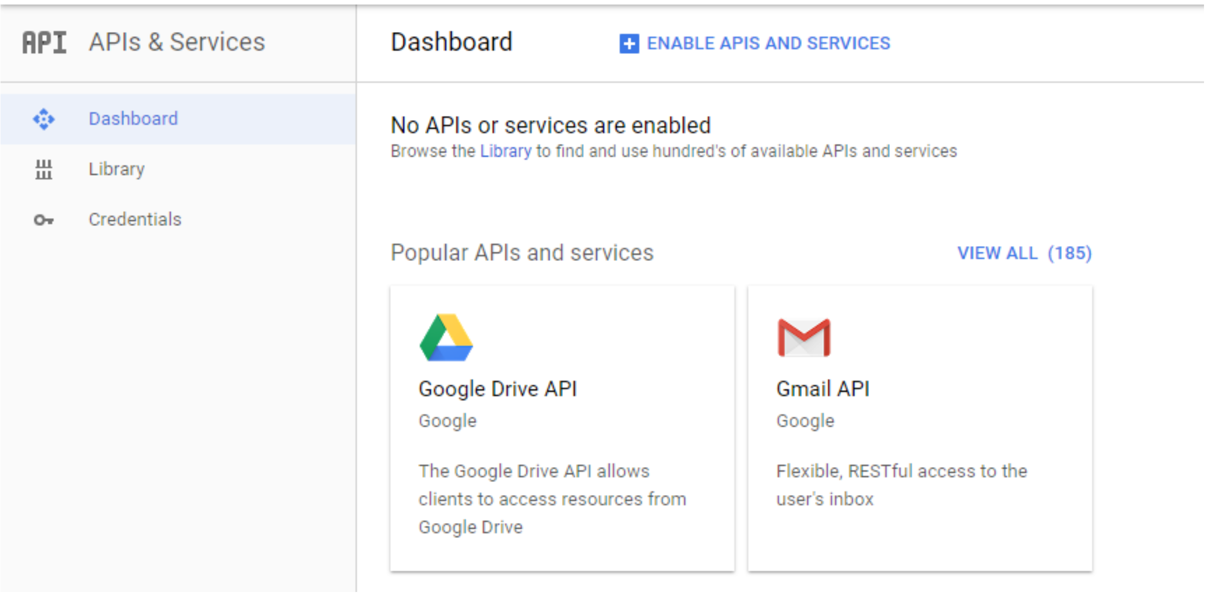
You have 6 projects remaining in your quota. Learn more.

Project name

Neos GA

Your project ID will be neos-ga-190913 Edit

After the project has been created, you end up in the dashboard of the new project.



No APIs or services are enabled

Browse the Library to find and use hundred's of available APIs and services

Popular APIs and services

Google Drive API

Google

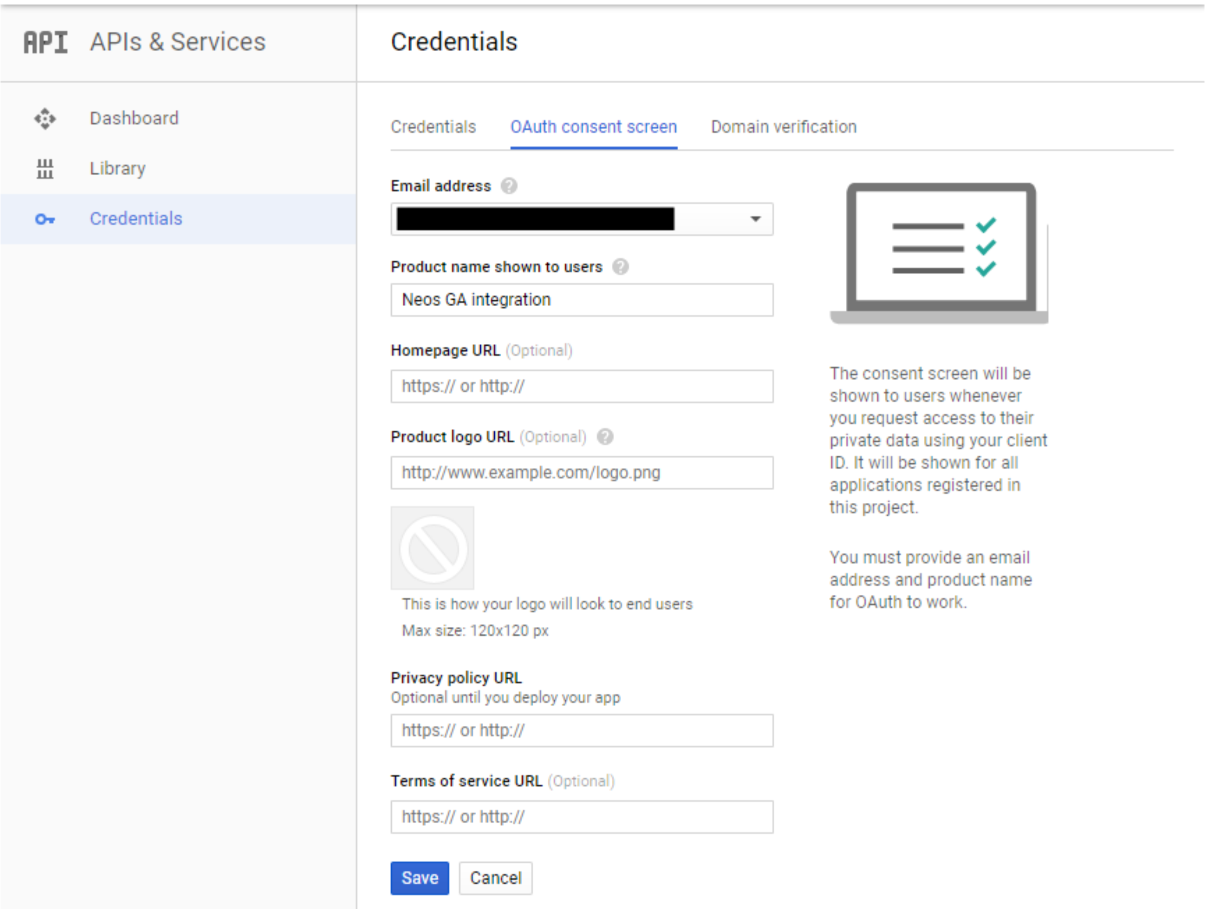
The Google Drive API allows clients to access resources from Google Drive

Gmail API

Google

Flexible, RESTful access to the user's inbox

The next step is to set up the Consent screen. Click on *APIs & Services* and then *Credentials* > *OAuth consent screen*, select an email address and fill in the product name. Again, this screen will probably only be shown to you, so don't think too hard about it. All other values in the screen are optional.



Email address

[redacted]

Product name shown to users

Neos GA integration

Homepage URL (Optional)

https:// or http://

Product logo URL (Optional)

http://www.example.com/logo.png

This is how your logo will look to end users

Max size: 120x120 px

Privacy policy URL
Optional until you deploy your app

https:// or http://

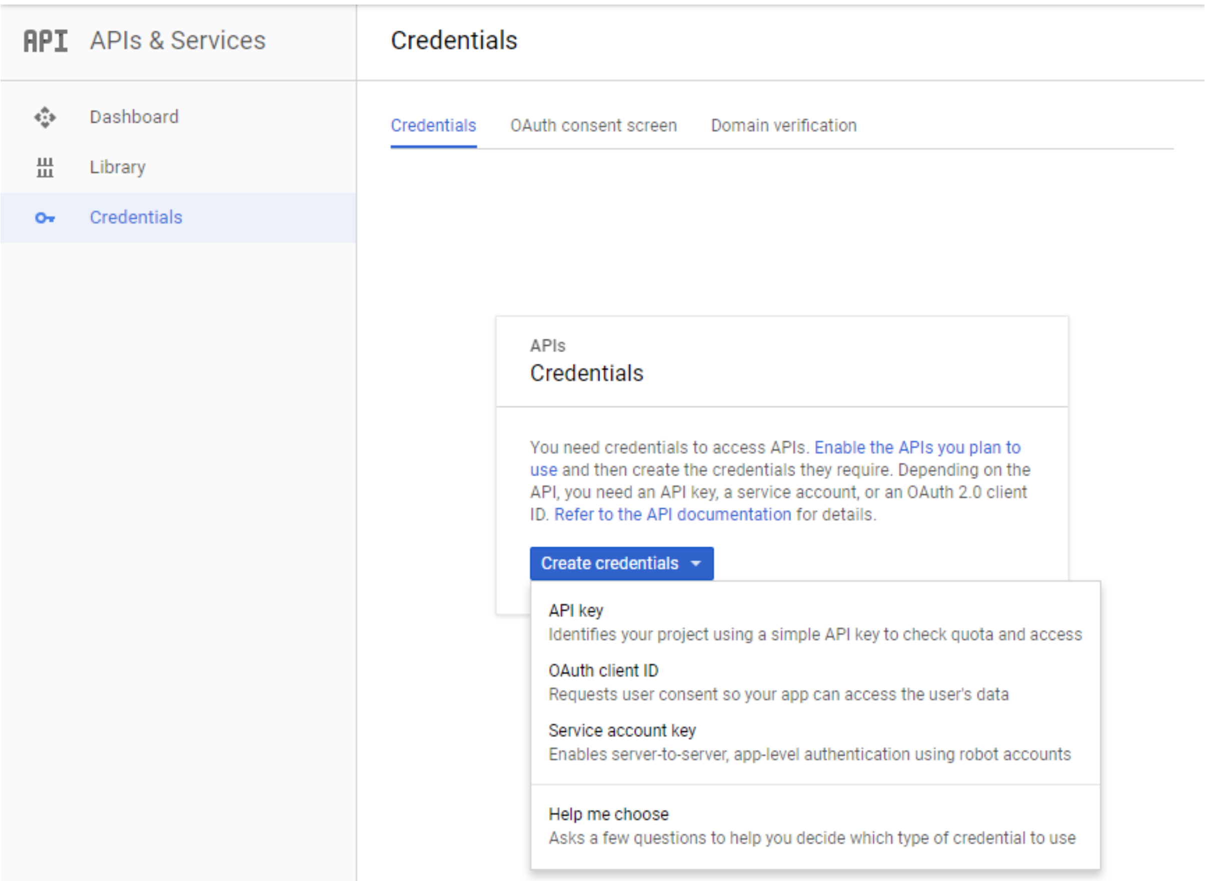
Terms of service URL (Optional)

https:// or http://

The consent screen will be shown to users whenever you request access to their private data using your client ID. It will be shown for all applications registered in this project.

You must provide an email address and product name for OAuth to work.

Now you need to create credentials for the integration. To do this, click on *Credentials* in the navigation, then *Create credentials* > “OAuth client ID*.



APIs
Credentials

You need credentials to access APIs. Enable the APIs you plan to use and then create the credentials they require. Depending on the API, you need an API key, a service account, or an OAuth 2.0 client ID. Refer to the API documentation for details.

API key
Identifies your project using a simple API key to check quota and access

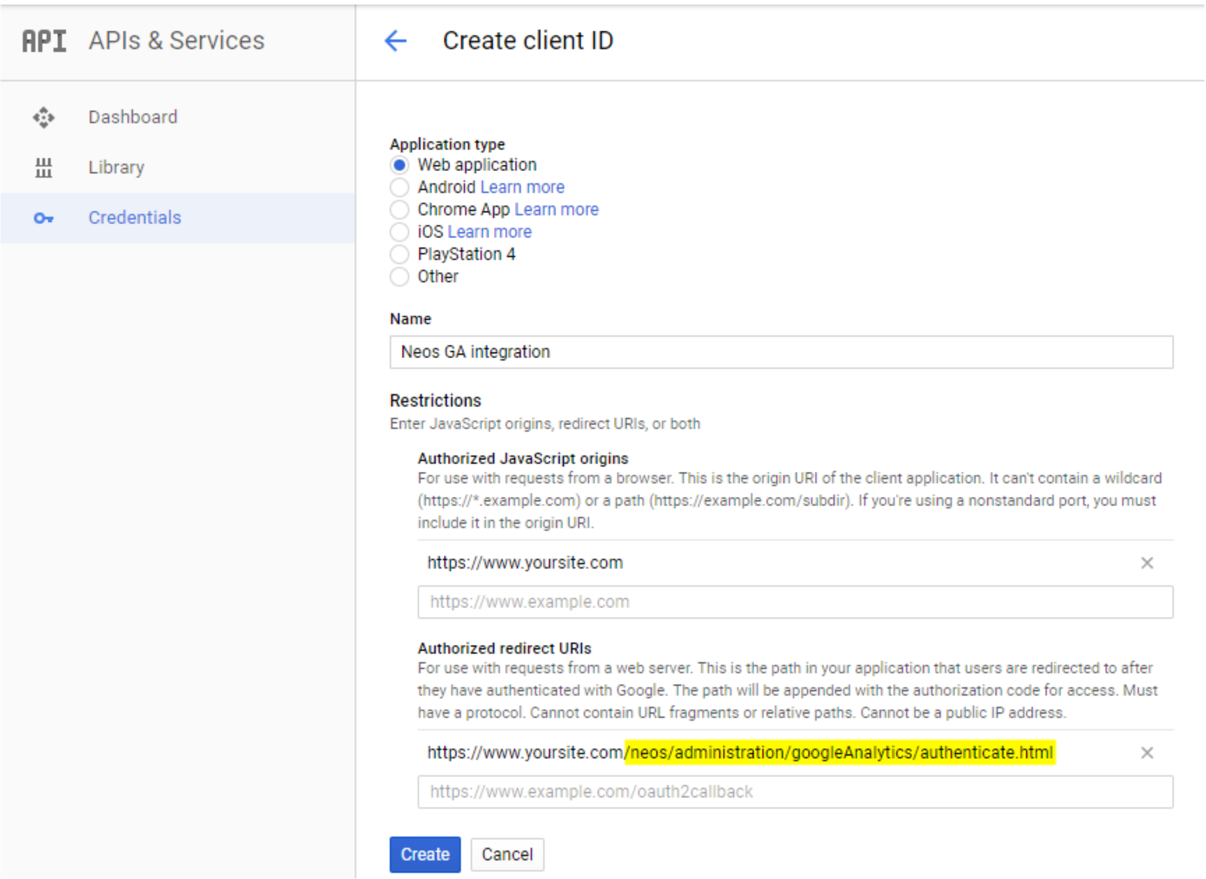
OAuth client ID
Requests user consent so your app can access the user's data

Service account key
Enables server-to-server, app-level authentication using robot accounts

Help me choose
Asks a few questions to help you decide which type of credential to use

Here you need to choose *Web application* and fill in the domain of your domain as *Authorized JavaScript origin* and must define an *Authorized redirect URI*. It has to be your domain again, followed by:

```
/neos/administration/googleAnalytics/authenticate.html
```



Application type

● Web application
○ Android Learn more
○ Chrome App Learn more
○ iOS Learn more
○ PlayStation 4
○ Other

Name

Neos GA integration

Restrictions

Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins

For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (https://*.example.com) or a path (https://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

https://www.yoursite.com ×

https://www.example.com

Authorized redirect URIs

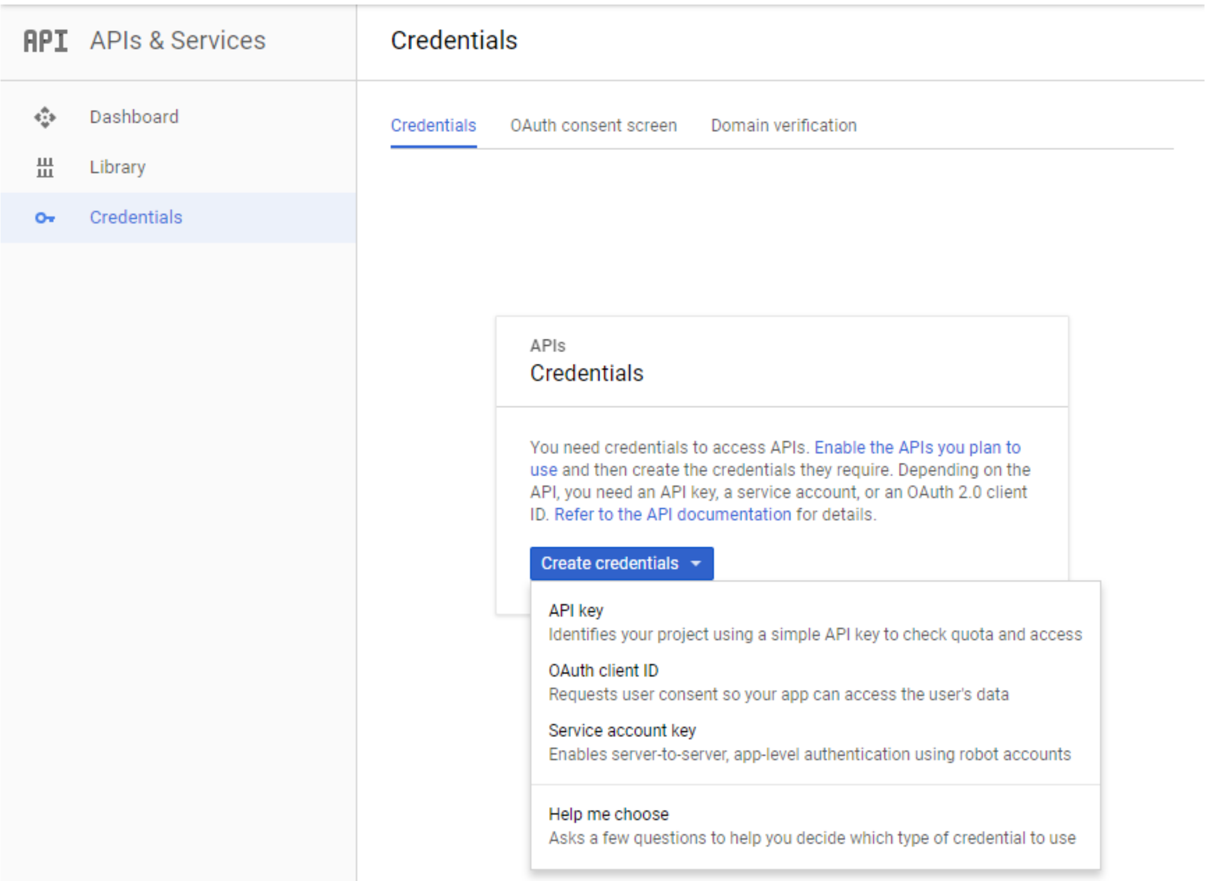
For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

https://www.yoursite.com/neos/administration/googleAnalytics/authenticate.html ×

https://www.example.com/oauth2callback

Then click *Create*.

Back on credentials page you have to create a new public API access key by clicking on *Create credentials* > *Api key*.



APIs

Credentials

You need credentials to access APIs. Enable the APIs you plan to use and then create the credentials they require. Depending on the API, you need an API key, a service account, or an OAuth 2.0 client ID. Refer to the API documentation for details.

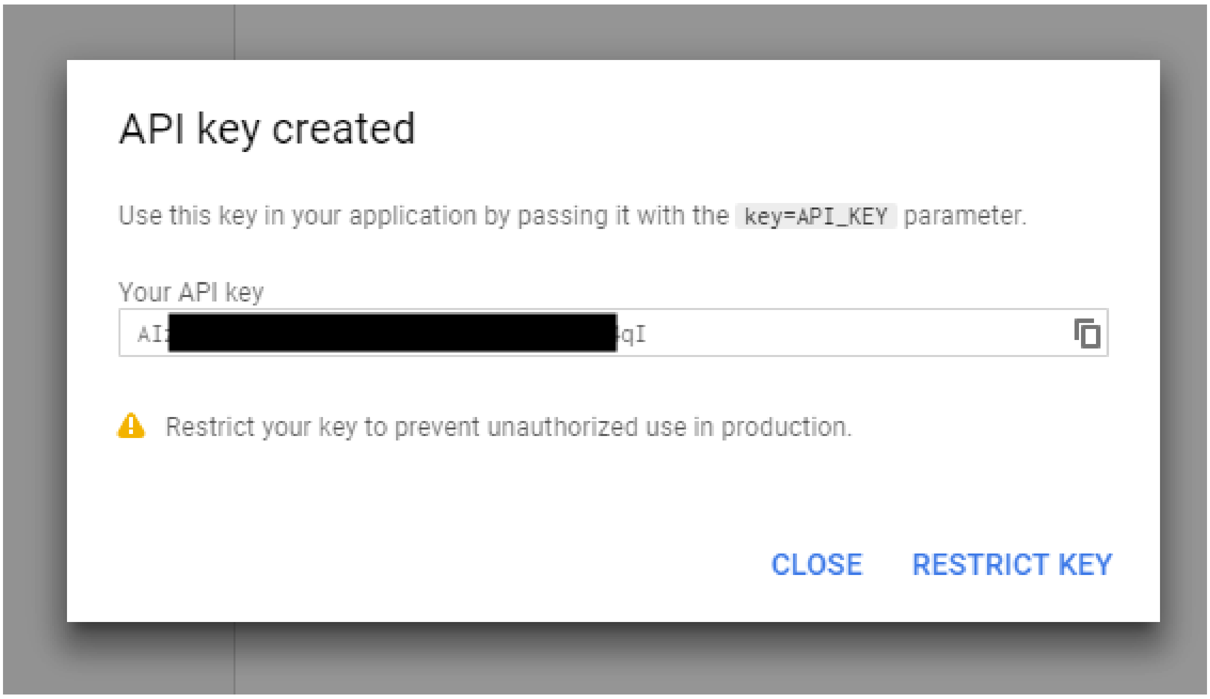
Create credentials ▾

API key
Identifies your project using a simple API key to check quota and access

OAuth client ID
Requests user consent so your app can access the user's data

Service account key
Enables server-to-server, app-level authentication using robot accounts

Help me choose
Asks a few questions to help you decide which type of credential to use



API key created

Use this key in your application by passing it with the `key=API_KEY` parameter.

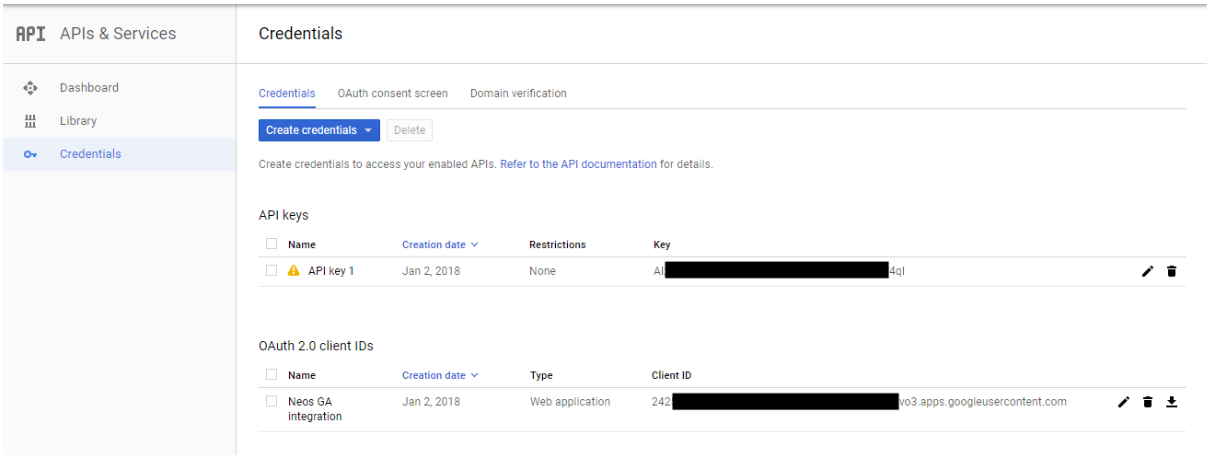
Your API key

AI[redacted]qI

⚠ Restrict your key to prevent unauthorized use in production.

In the following dialog, click *Restrict key* to define allowed IP addresses, if you like. Otherwise click “Close”.

Your credentials overview should now look like this:



Credentials

Create credentials to access your enabled APIs. Refer to the API documentation for details.

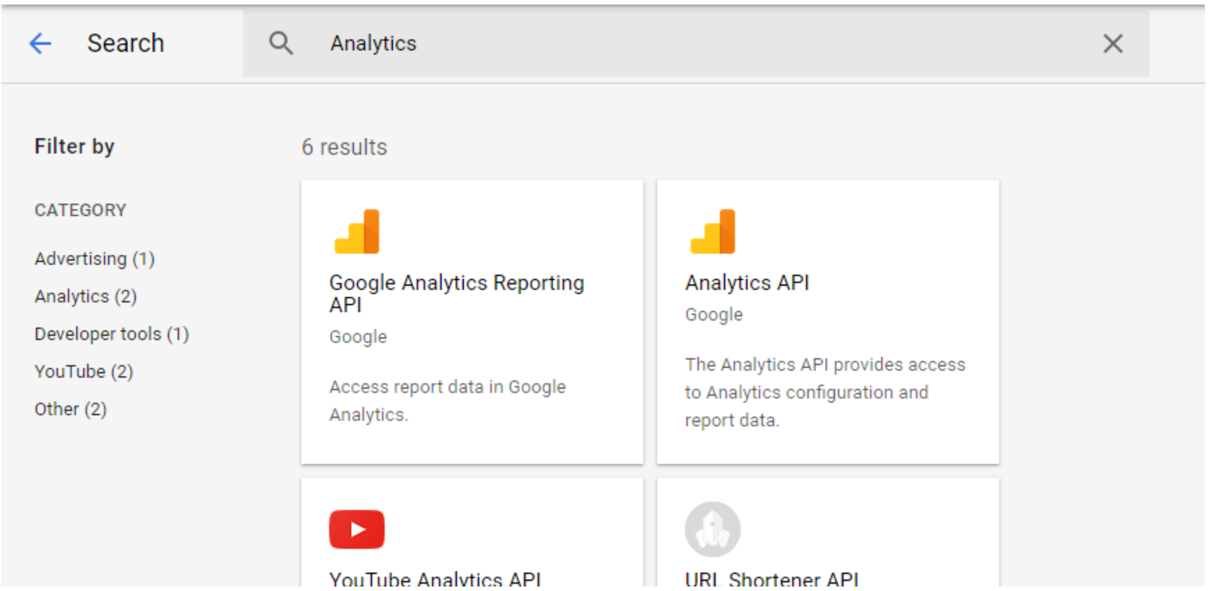
API keys

Name	Creation date	Restrictions	Key
API key 1	Jan 2, 2018	None	AI[redacted]4ql

OAuth 2.0 client IDs

Name	Creation date	Type	Client ID
Neos GA integration	Jan 2, 2018	Web application	242[redacted]vo3.apps.googleusercontent.com

The last step in the Developers Console is to make sure the Google Analytics API is enabled. Click on *Library* in the left navigation and enter “Analytics” in the search field.



Filter by

CATEGORY

Advertising (1)
Analytics (2)
Developer tools (1)
YouTube (2)
Other (2)

6 results

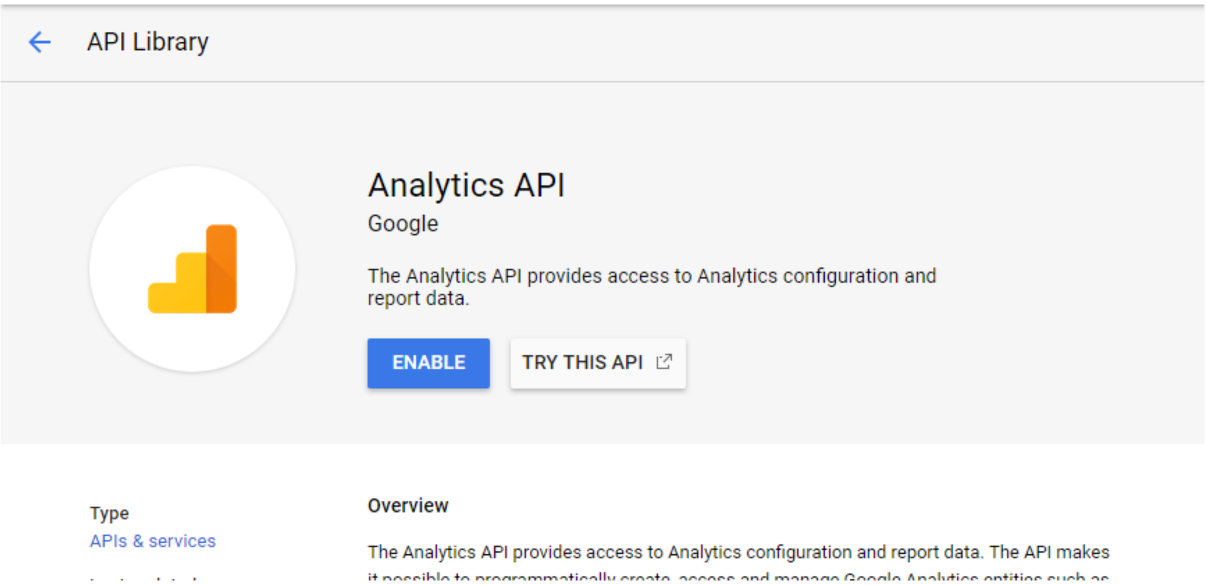
Google Analytics Reporting API
Google
Access report data in Google Analytics.

Analytics API
Google
The Analytics API provides access to Analytics configuration and report data.

YouTube Analytics API

URL Shortener API

Click on the Analytics API link and on the following page on Enable.



Analytics API
Google
The Analytics API provides access to Analytics configuration and report data.

Type
APIs & services

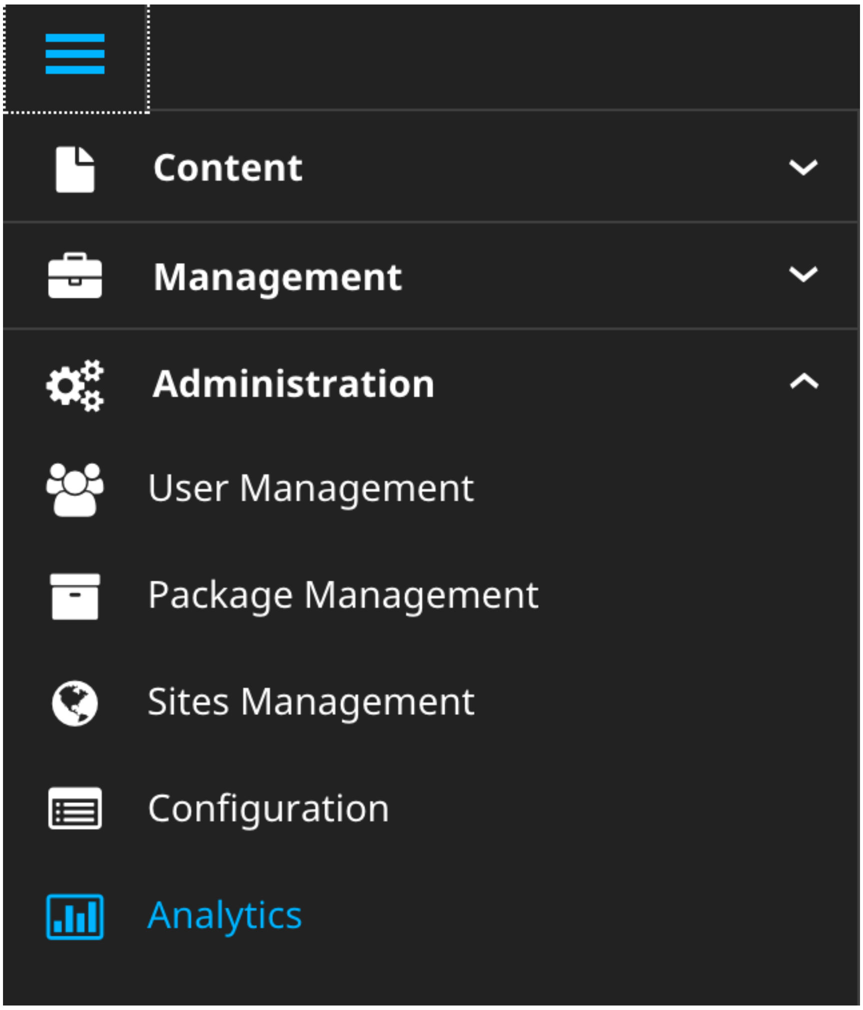
Overview

The Analytics API provides access to Analytics configuration and report data. The API makes it possible to programmatically create, access and manage Google Analytics entities such as

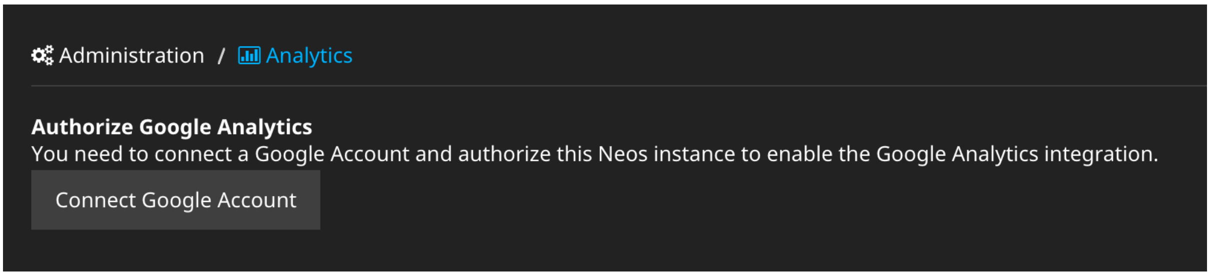
With the data from the credentials overview screen you can now configure the Neos integration, just fill in the blanks:

```
Neos:
  GoogleAnalytics:
    authentication:
      # Application name for the consent screen
      applicationName: 'Neos GA'
      # "Client ID" of the OAuth 2.0 "Client ID for web application"
      clientId: ~
      # "Client Secret" of the OAuth 2.0 "Client ID for web application"
      clientSecret: ~
      # "API Key" of the Public API access credential "Key for server applications"
      developerKey: ~
```

Now go to the Analytics module in the Neos user interface.



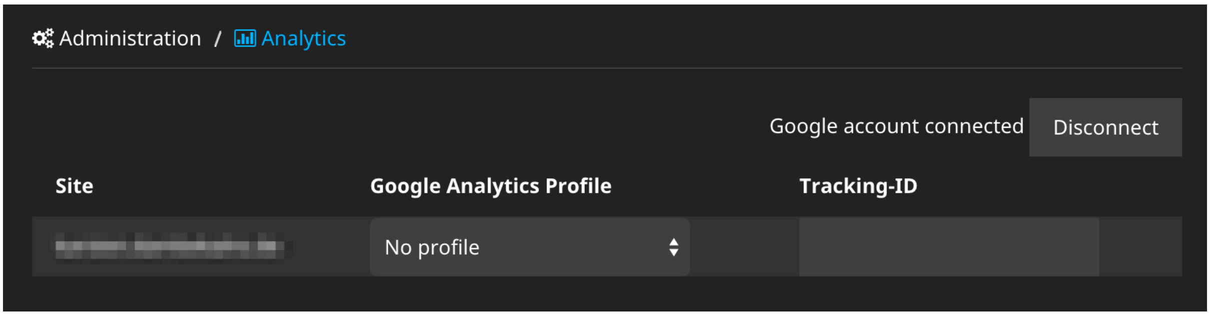
You will see a button to connect your Neos instance to a Google account.



Authorize Google Analytics

You need to connect a Google Account and authorize this Neos instance to enable the Google Analytics integration.

Click it and you will be redirected to Google, showing you your consent screen and asking for offline read access to Google Analytics data. Agree and you will be redirected back to Neos:



Site	Google Analytics Profile	Tracking-ID
[redacted]	No profile	

Now the integration is set up to fetch data and display the Analytics statistics in the Neos user interface.

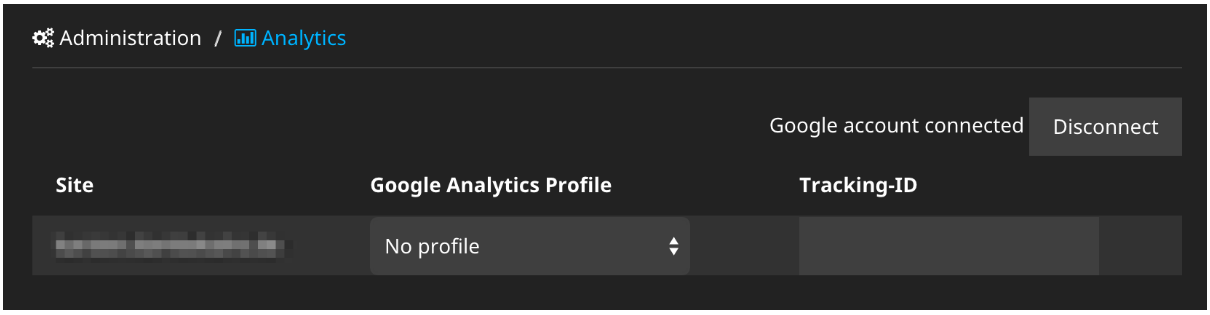
2.2 Assigning tracking IDs

To actually track visits with Google Analytics, some JavaScript with the tracking ID has to be included in the markup. You can do this manually in your template, but the easier way is to set up tracking IDs for each site in the Analytics integration.

This can be done in two ways. The first is via settings and allows for versioning the settings and easy deployments:

```
Neos:
  GoogleAnalytics:
    sites:
      ## All site specific settings are indexed by site node name
      neossitename:
        trackingId: 'UA-XXXXX-YY'
        profileId: '12345678'
```

The second way is to use the Analytics module in the Neos user interface to select an Analytics property for each site (the `profileId` from the settings above) and configure a tracking ID.

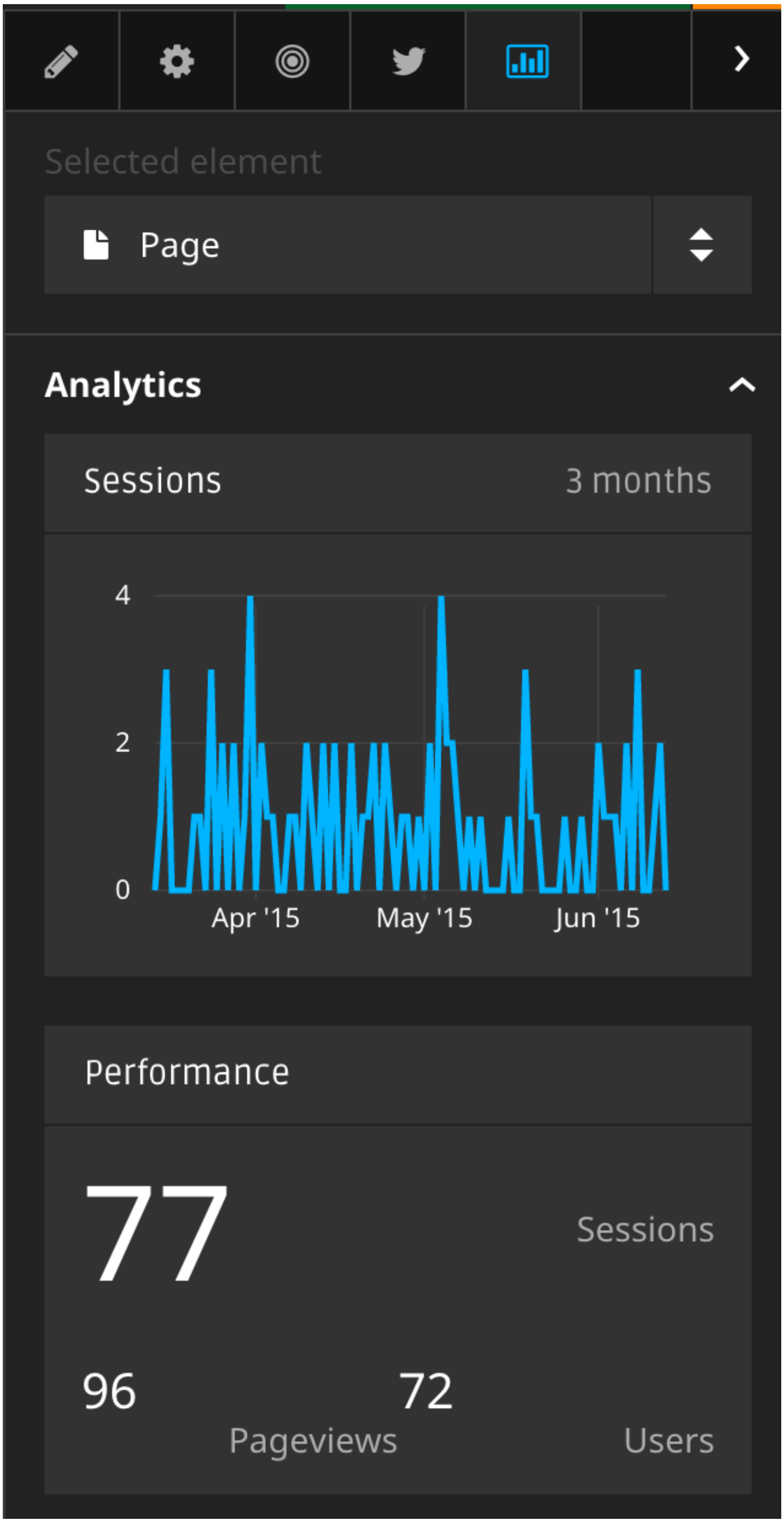


Site	Google Analytics Profile	Tracking-ID
[redacted]	No profile	

CHAPTER 3

Usage

If Google has collected statistical data for your site, you will get the most important data shown in an inspector tab for documents:



Selected element

Page

Analytics

Sessions 3 months

Performance

77 Sessions

96 Pageviews

72 Users

3.1 Customizing metrics display

Its is possible to adjust the displayed metrics through configuration. The package comes with preconfigured data, examine `NodeTypes.yaml` and `Settings.yaml`.