Neos Google Analytics Integration

Release dev-master

The Neos Team

Aug 08, 2023

Contents

1 Installation **3**

2 Configuration **5**
2.1 Configure tracking 5

3 Disable tracking **7**

4 Additional Parameters **9**
4.1 Statistics display inside Neos 9

5 Usage **17**
5.1 Customizing metrics display 18

6 Showing metrics in non-live environments **19**

7 Show only page information **21**
7.1 Upgrade instructions (2.x -> 3.0.0) 21

8 `enableTracking` setting **23**

9 Authentication with the Google API **25**

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 08, 2023

CHAPTER 1

Installation

The package can be installed via composer in your site package:

```
composer require --no-update neos/googleanalytics
```

Then run *composer update* in your projects root directory.

CHAPTER 2

Configuration

2.1 Configure tracking

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 for each site in the Analytics integration.

```
Neos:
  GoogleAnalytics:
    sites:
      ## All site specific settings are indexed by site node name
      neosSiteName:
        analytics:
          id: 'UA-XXXXX-YY'
```

The Google Analytics tracking also supports additional parameters, f.e. to anonymize ips. You can define them like this:

```
Neos:
  GoogleAnalytics:
    sites:
      neosSiteName:
        analytics:
          id: 'UA-XXXXX-YY'
          parameters:
            anonymize_ip: true
```

Instead of using the Google Analytics tracking code, you can integrate the Google Tag Manager the same way:

```
Neos:
  GoogleAnalytics:
    sites:
      neosSiteName:
        tagManager:
          id: 'GTM-XXXXX'
```

Note: If you configure both, a container and an Analytics ID, only the Tag Manager is included.

It is also possible to define default values for all sites. These will be merged with any site specific settings.

```
Neos:
  GoogleAnalytics:
    default:
      analytics:
        id: 'UA-XXXXX-YY'
```

The analytics backend module will also give you and overview of all configured sites and their ids.

CHAPTER 3

Disable tracking

You can disable tracking for a site by either setting the ‘‘id ‘‘ to ‘‘false ‘‘ (this is the default value), or leaving it blank.

```
Neos:
  GoogleAnalytics:
    sites:
      neosSiteName:
        tagManager:
          id: false
```

CHAPTER 4

Additional Parameters

If you are using the Google Analytics tracking code, you can also add additional parameters e.g. ‘‘authorize_ip ‘‘ to ‘‘true ‘‘ (IP anonymization with gtag.js). These paramters are added automatically as JSON to the gtag. The default setting is no parameters.

```
Neos:
  GoogleAnalytics:
    sites:
      neosSiteName:
        analytics:
          id: 'UA-XXXXX-YY'
          parameters:
            anonymize_ip: true
```

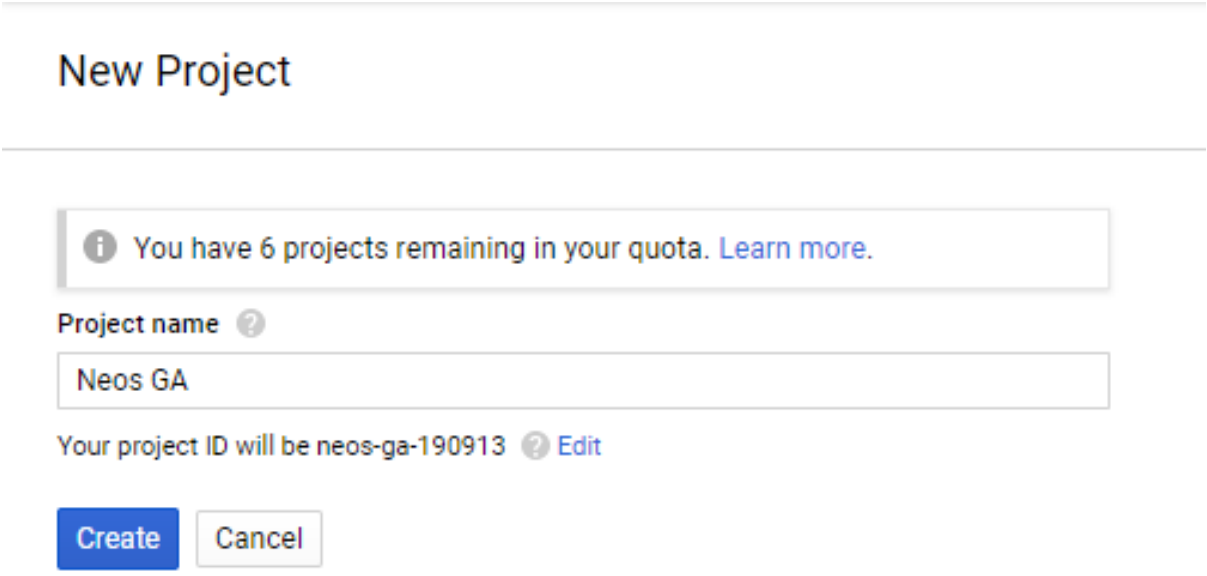
4.1 Statistics display inside Neos

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

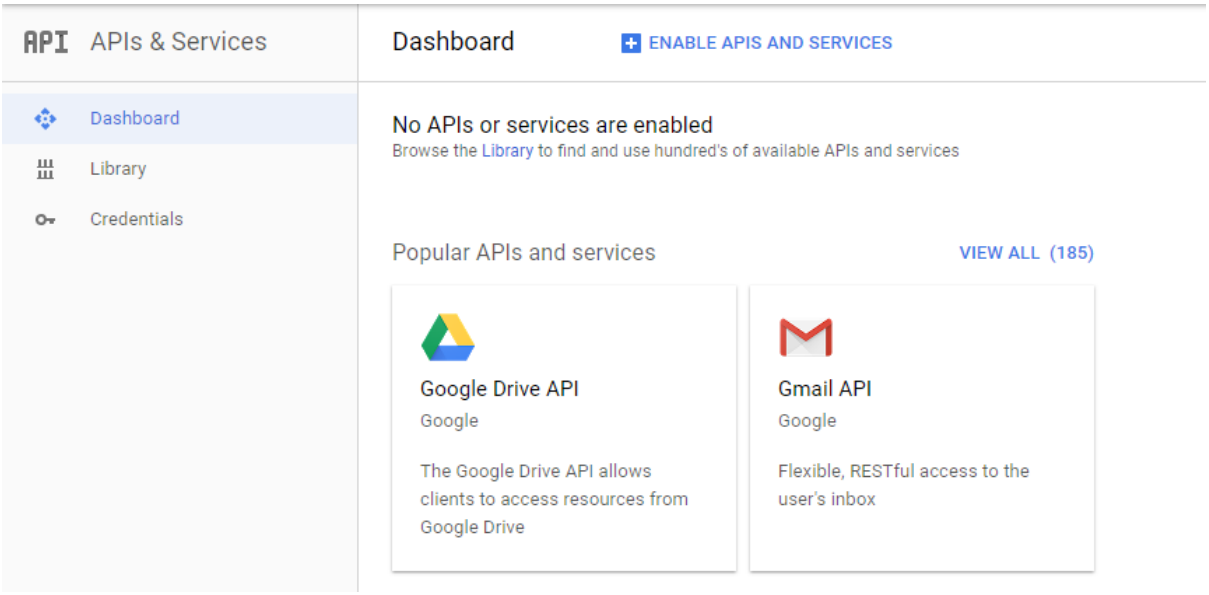
1. Log in to the Google Developers Console
2. Create a service account key
3. Enable the Analytics API
4. Give the service account permissions to your analytics account
5. Configure the Google Analytics Neos integration using the created credentials

The steps in detail

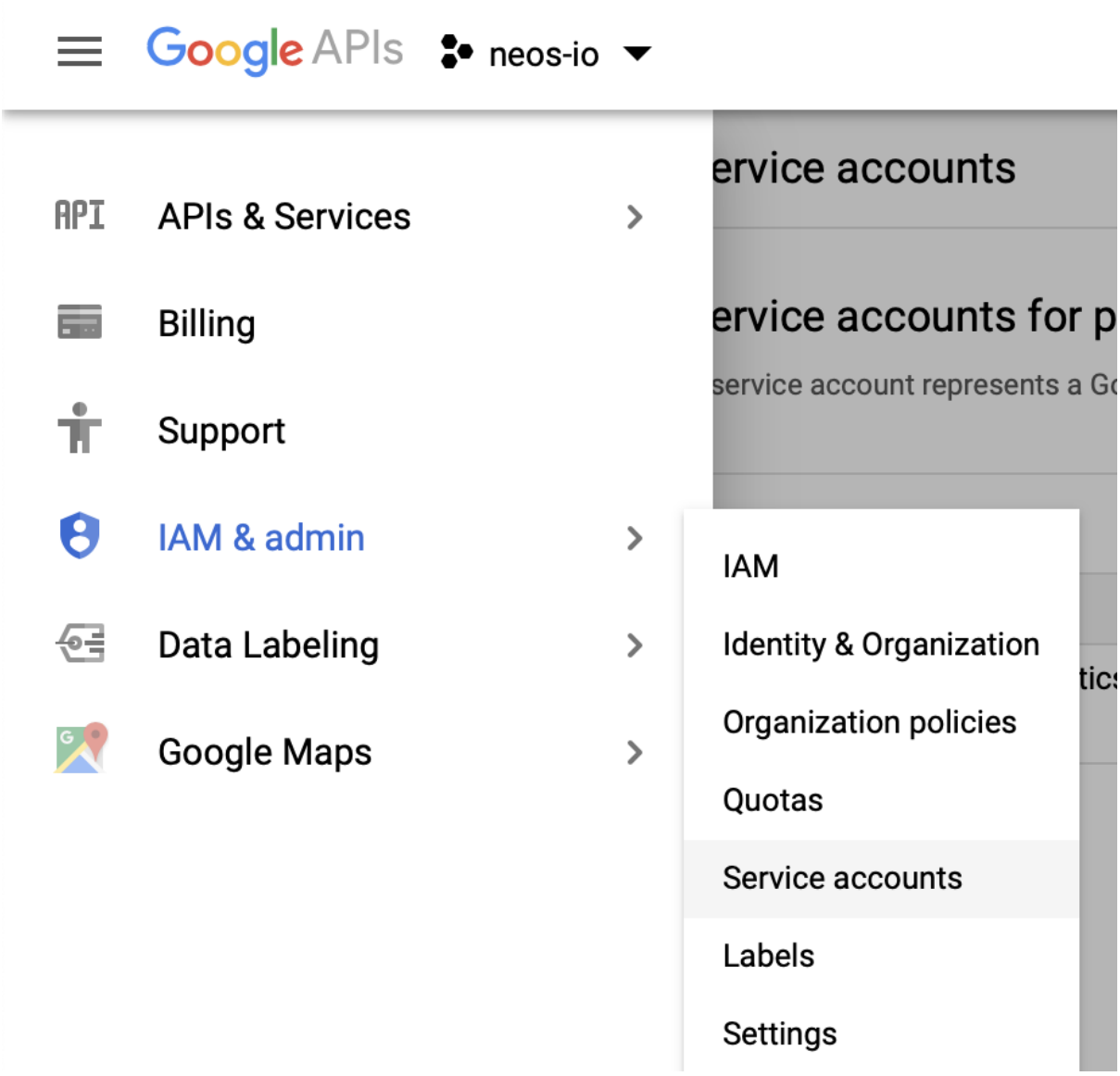
Go to the Google Developers Console and create a project. This is needed to create the API credentials and keys for the Analytics integration. The project name can be chosen freely and doesn’t really matter, as you will probably be the only one to ever see it.



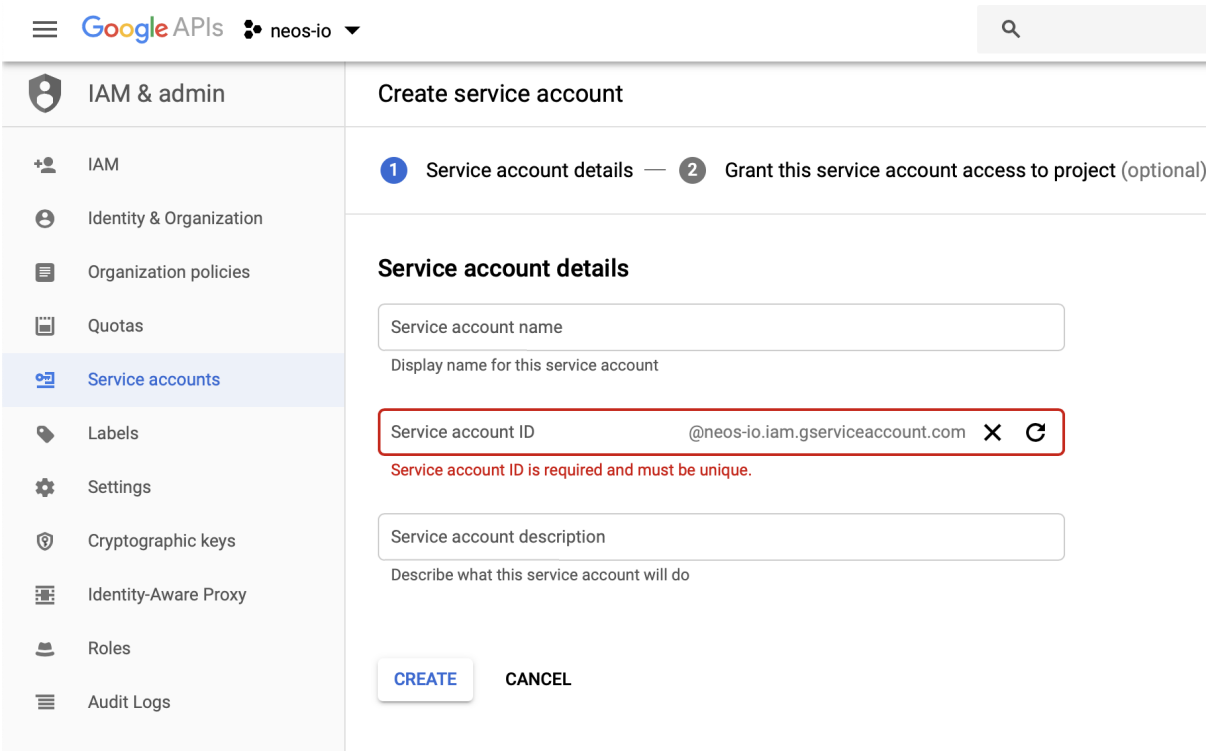
After the project has been created, you end up in the dashboard of the new project.



If you don't have one yet you have to create a new service account. For this go to "Service accounts" in the sidebar:

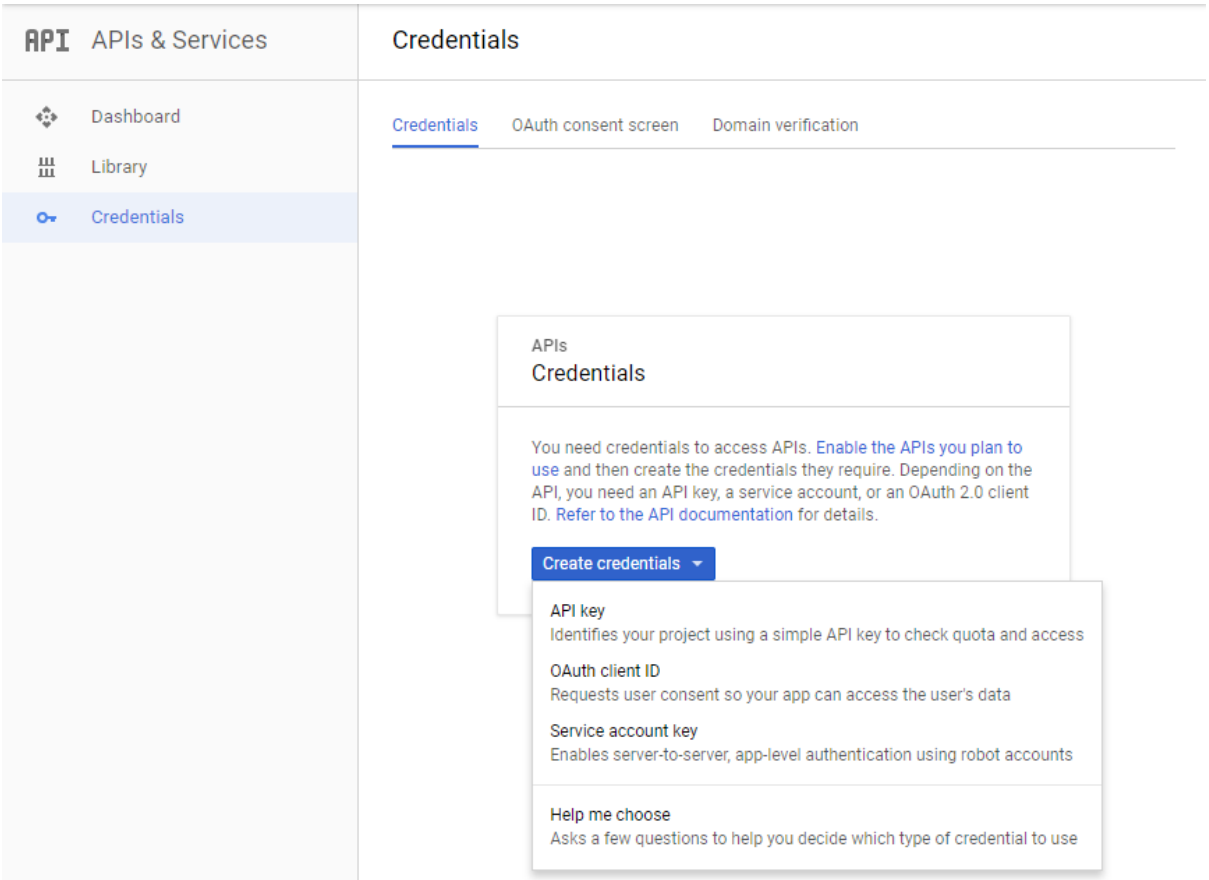


Now create a new one and fill out the required information:

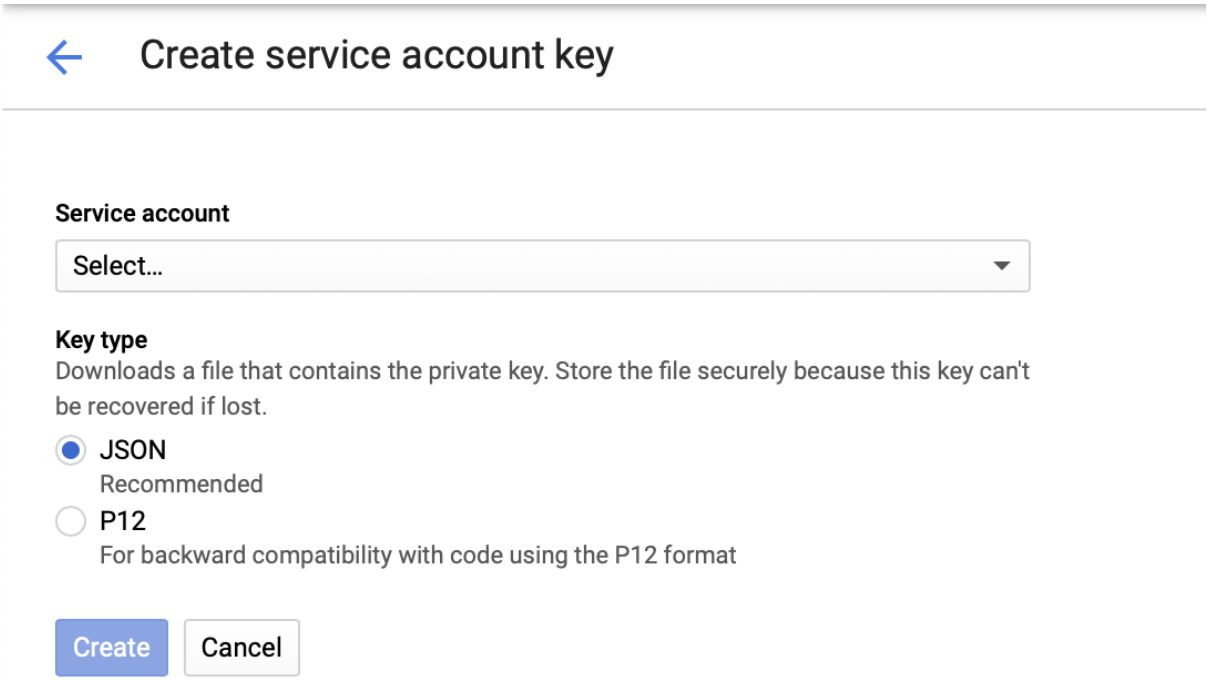


Now give the account the *Viewer* role for the project.

Now you need to create credentials for the integration. To do this, click on *Credentials* in the navigation, then *Create credentials* > "Service account key*.

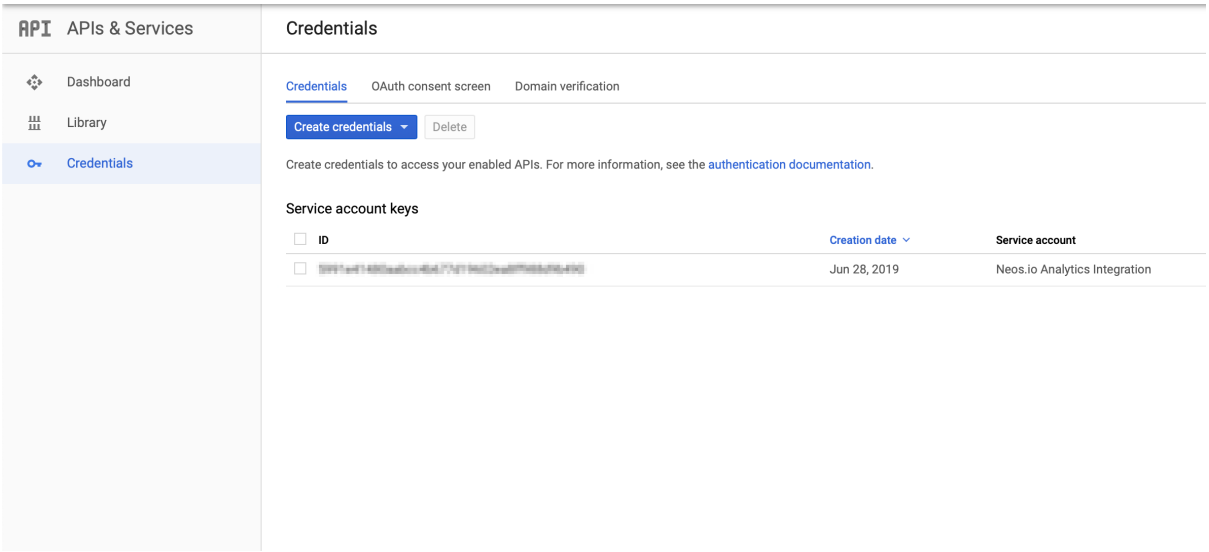


Here you need to choose your service account and select *json* as key type:

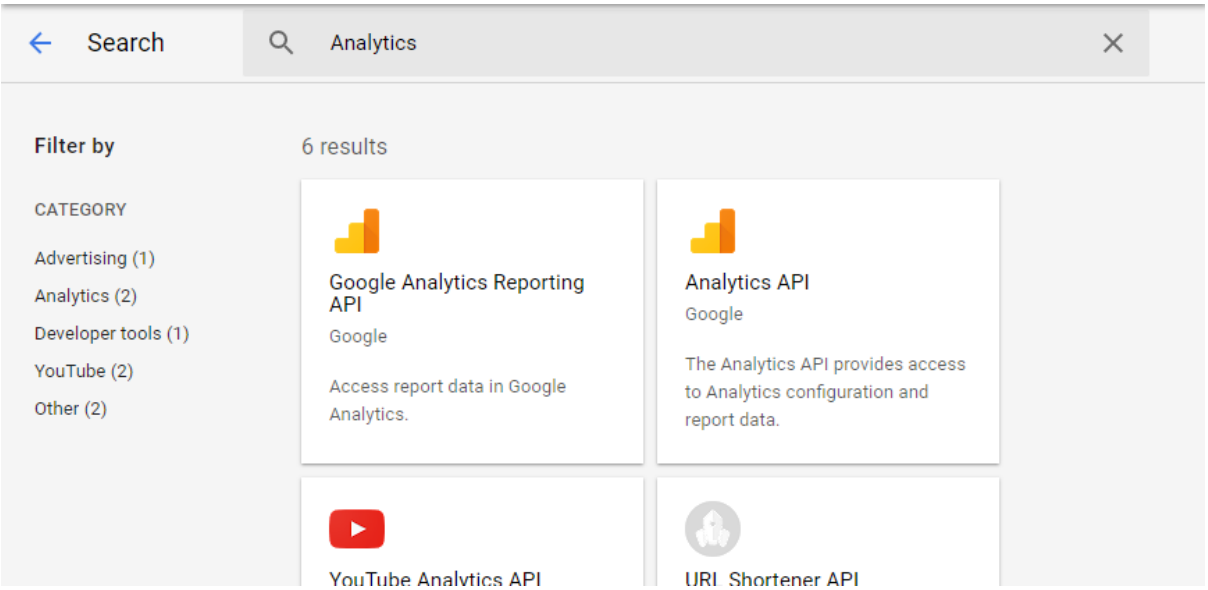


Then click *Create*. The download of your credentials file should start immediately.

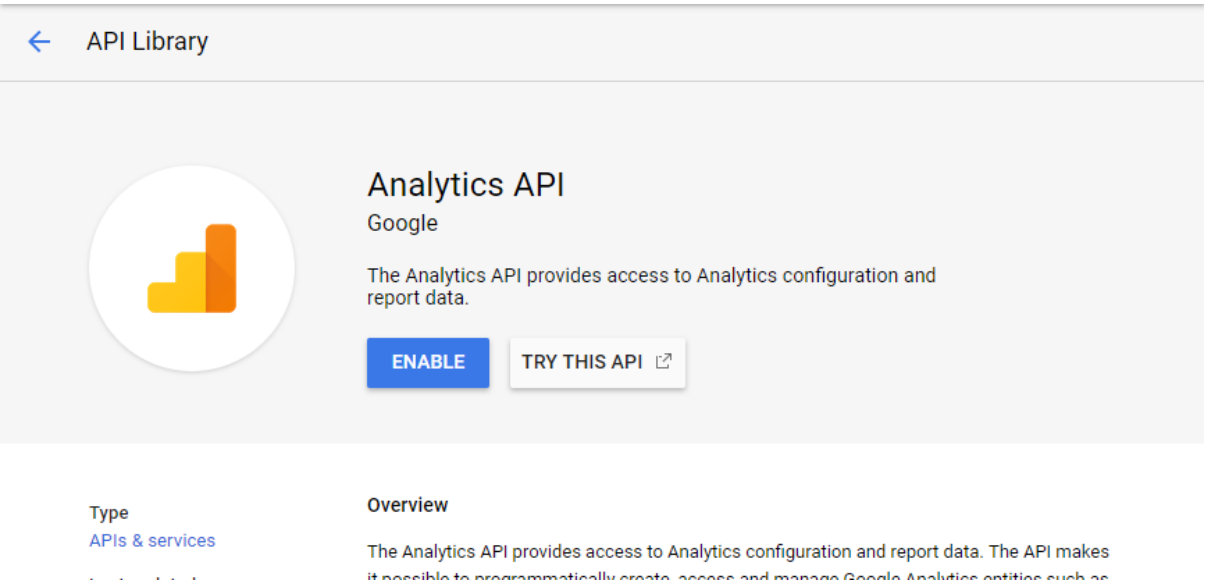
Your credentials overview should now look like this:



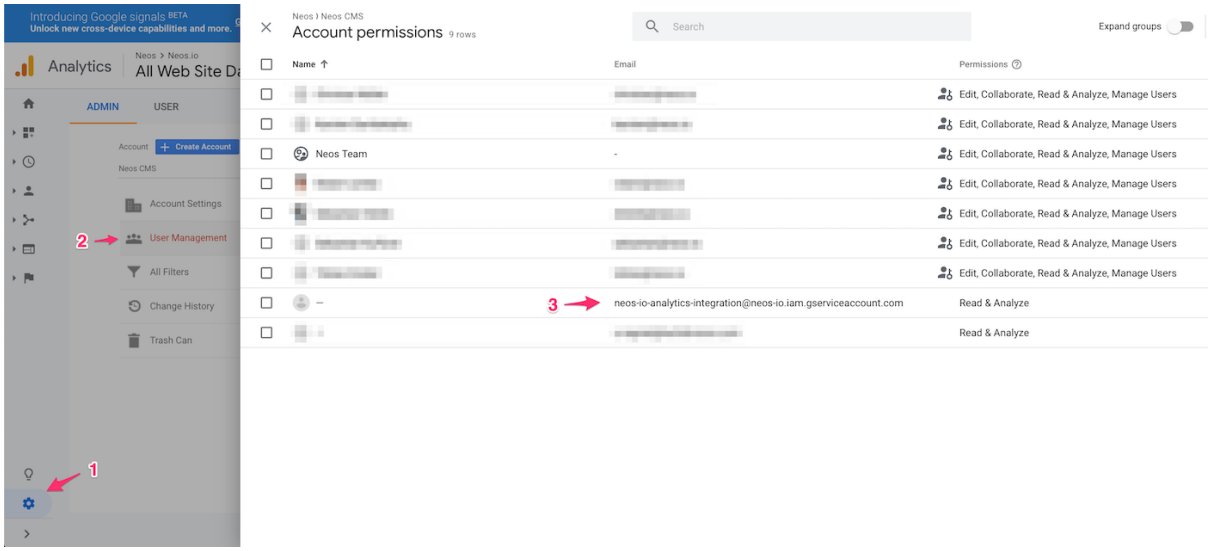
The last step in the Developers Console is to make sure the Google Analytics API is enabled. Click on *Library* in the left navigation and enter “Analytics” in the search field.



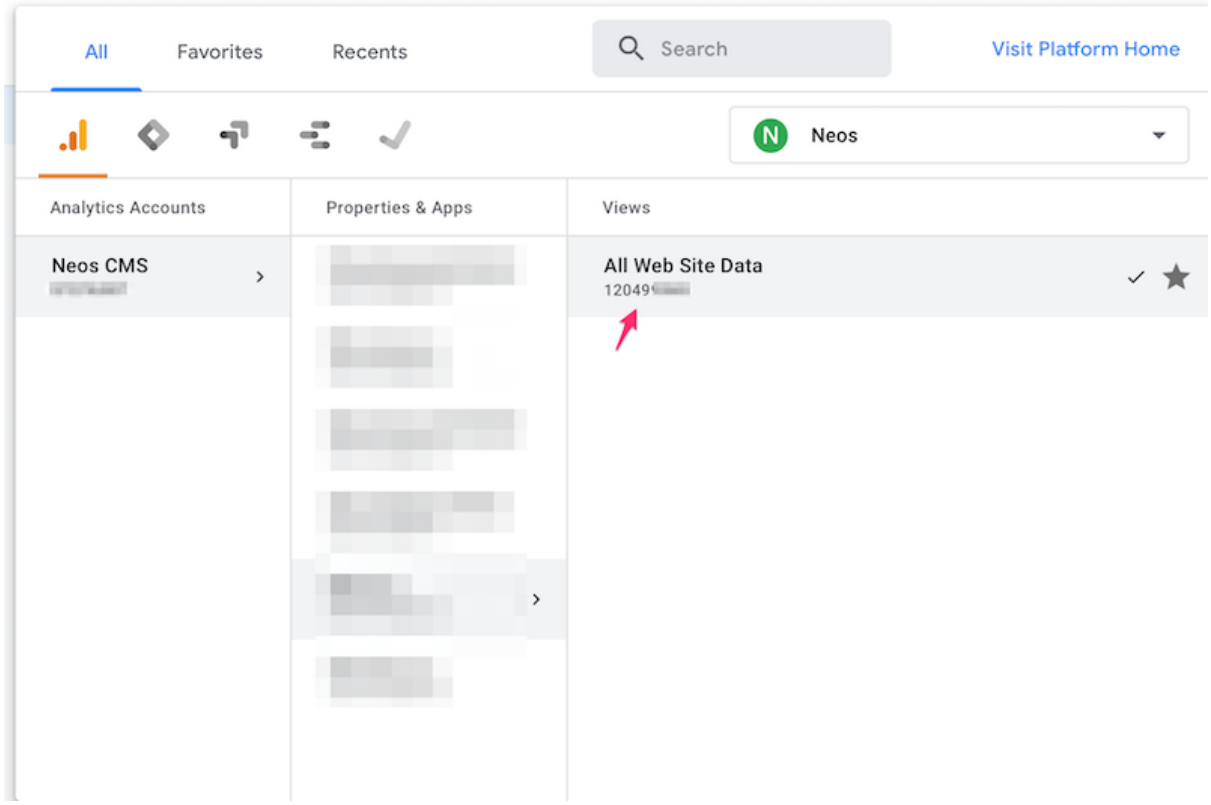
Click on the Analytics API link and on the following page on Enable.



To allow the service account to access your analytics data you have to give it the necessary permissions. For that log into Google Analytics and go to the *Admin* menu on the bottom left. Then select *User Management* and add the email address of your service account as a new user with the read & analyze permission.



Copy the profile id (visible as view id) from your analytics account and add it to your sites settings in your *Settings.yaml*.



The configuration should then look like this:

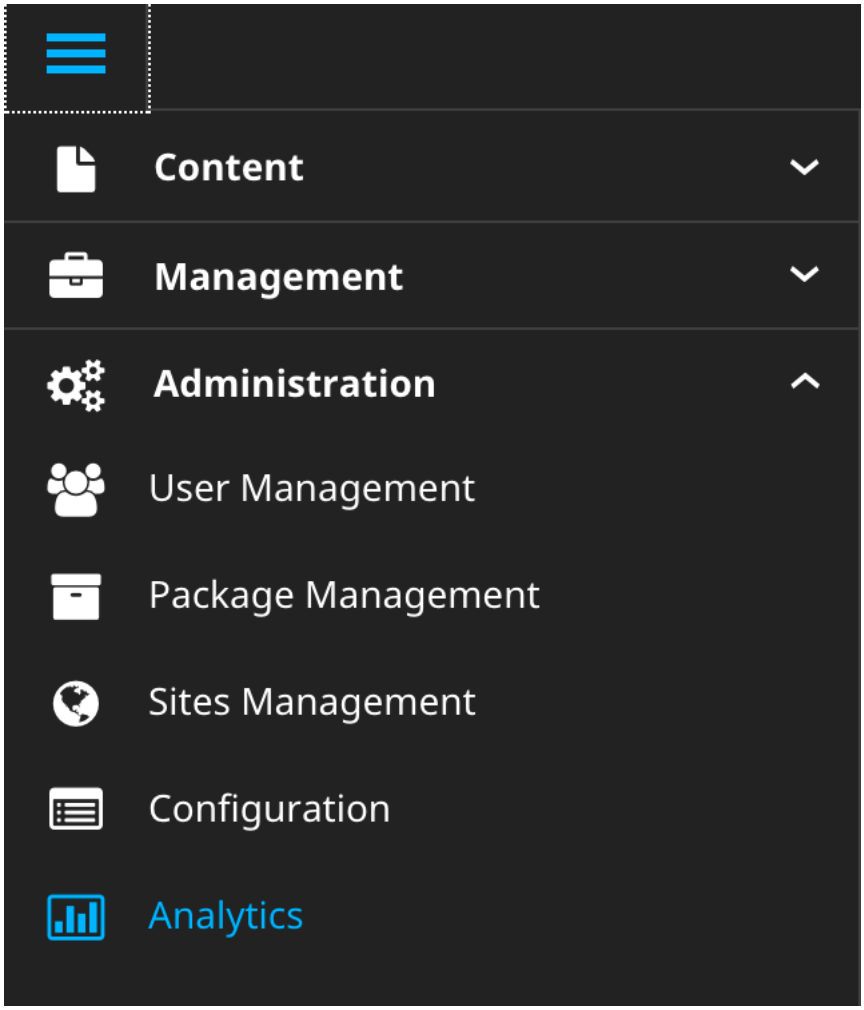
```
Neos:
  GoogleAnalytics:
    sites:
      neosSiteName:
        analytics:
          id: 'UA-XXXXX-YY'
        profileId: 123456789
```

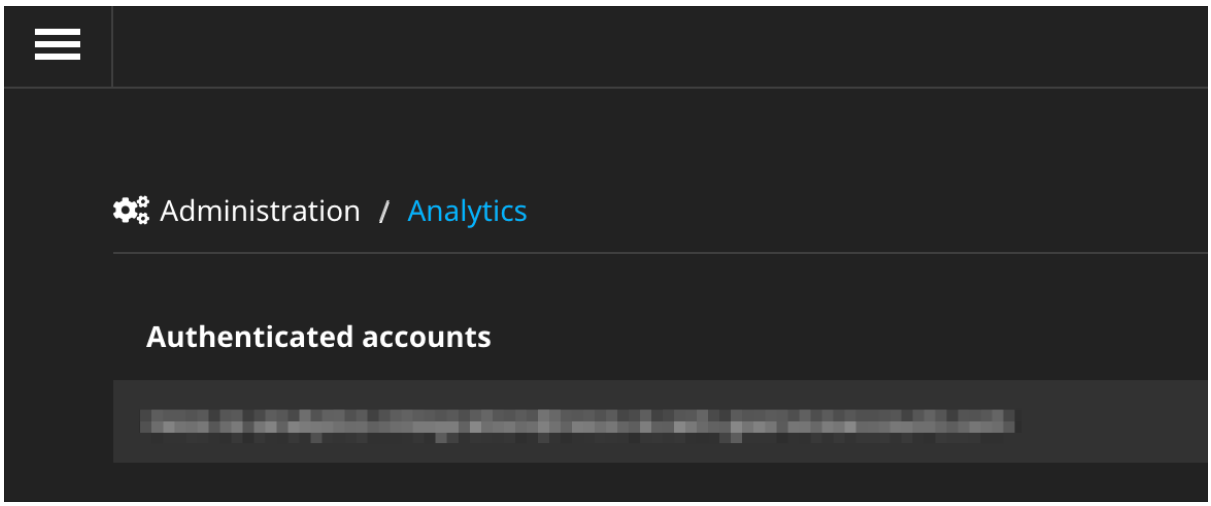
With the credentials file you downloaded before you can now configure Neos. You can either do this by storing the credentials in the persistent cache with the following command:

```
flow googleapi:storecredentials my-credentials-file.json
```

Or you can set the environment variable *GOOGLE_APPLICATION_CREDENTIALS* to the path where you store your credentials file.

Now go to the Analytics module in the Neos user interface and verify your setup.



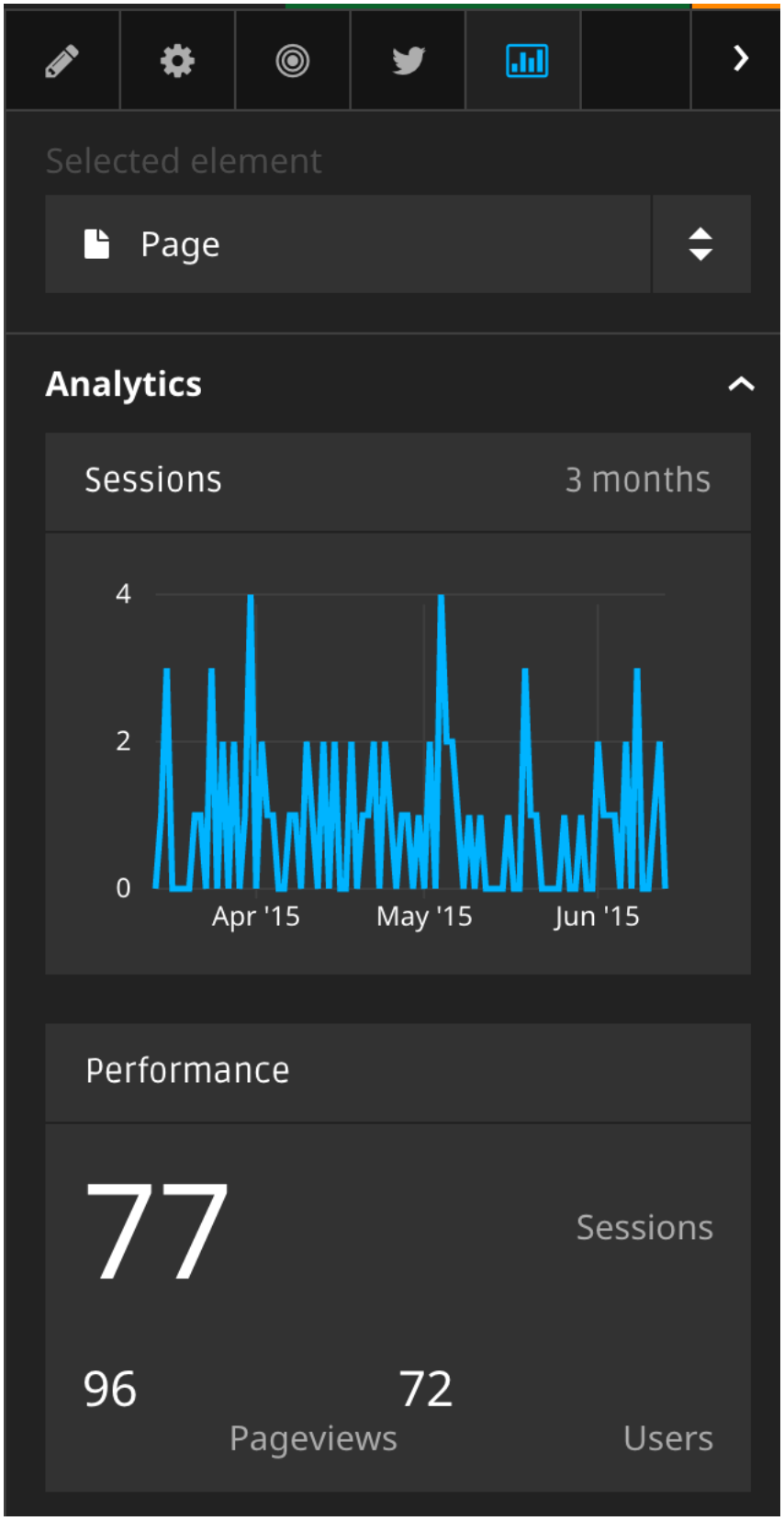


Now the integration is set up to fetch data and display the Analytics statistics in the Neos user interface.

CHAPTER 5

Usage

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



5.1 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`.

CHAPTER 6

Showing metrics in non-live environments

The configuration allows to override the hostname that is used for statistics queries. By default the hostname is used that Neos provides based on the selected pages frontend uri. This can be used to verify the functionality of the statistics module during development or in staging environments. Additionally, there is a setting for prepending a string to the frontend uri. The frontend uri is automatically created.

```
Neos:
  GoogleAnalytics:
    sites:
      neosSiteName:
        overrideHostname: 'example.org'
        pagePathPrefix: '/neosuri'
```

CHAPTER 7

Show only page information

To show the information for the whole site and not only the current page, the setting 'includePagePath' can be set. This removes the filter for the frontend uri from the API request.

```
Neos:
  GoogleAnalytics:
    sites:
      neosSiteName:
        includePagePath: false
```

7.1 Upgrade instructions (2.x -> 3.0.0)

Configuration for the tracking code has been changed:

```
Neos:
  GoogleAnalytics:
    sites:
      neosSiteName:
        analytics:
          id: 'UA-XXXXX-YY'
```

CHAPTER 8

`enableTracking` setting

Tracking code is now only included if you provide either a container or an Analytics ID. The `enableTracking` setting has therefore been removed.

CHAPTER 9

Authentication with the Google API

The OAuth authentication in the older version of this package was often unreliable and caused many issues.

Therefore you now need to create a service account and matching credentials. See the *Configuration* chapter above.