
Neos Google Analytics Integration

Release dev-master

The Neos Team

Apr 21, 2022

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 |
| 6.1 | Upgrade instructions (2.x -> 3.0.0) | 19 |
| 7 | enableTracking setting | 21 |
| 8 | Authentication with the Google API | 23 |

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: Apr 21, 2022

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.

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-XXXXXX-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-XXXXXX-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-XXXXXX'
```

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 an 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
```

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 parameters are added automatically as JSON to the gtag. The default setting is no parameters.

```
Neos:
  GoogleAnalytics:
    sites:
      neossitename:
        analytics:
          id: 'UA-XXXXXX-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.

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](#)

Create

Cancel


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

APIs & Services

Dashboard


Library

Credentials

Dashboard  ENABLE APIS AND SERVICES


No APIs or services are enabled
Browse the [Library](#) to find and use hundred's of available APIs and services

Popular APIs and services [VIEW ALL \(185\)](#)



Google Drive API
Google

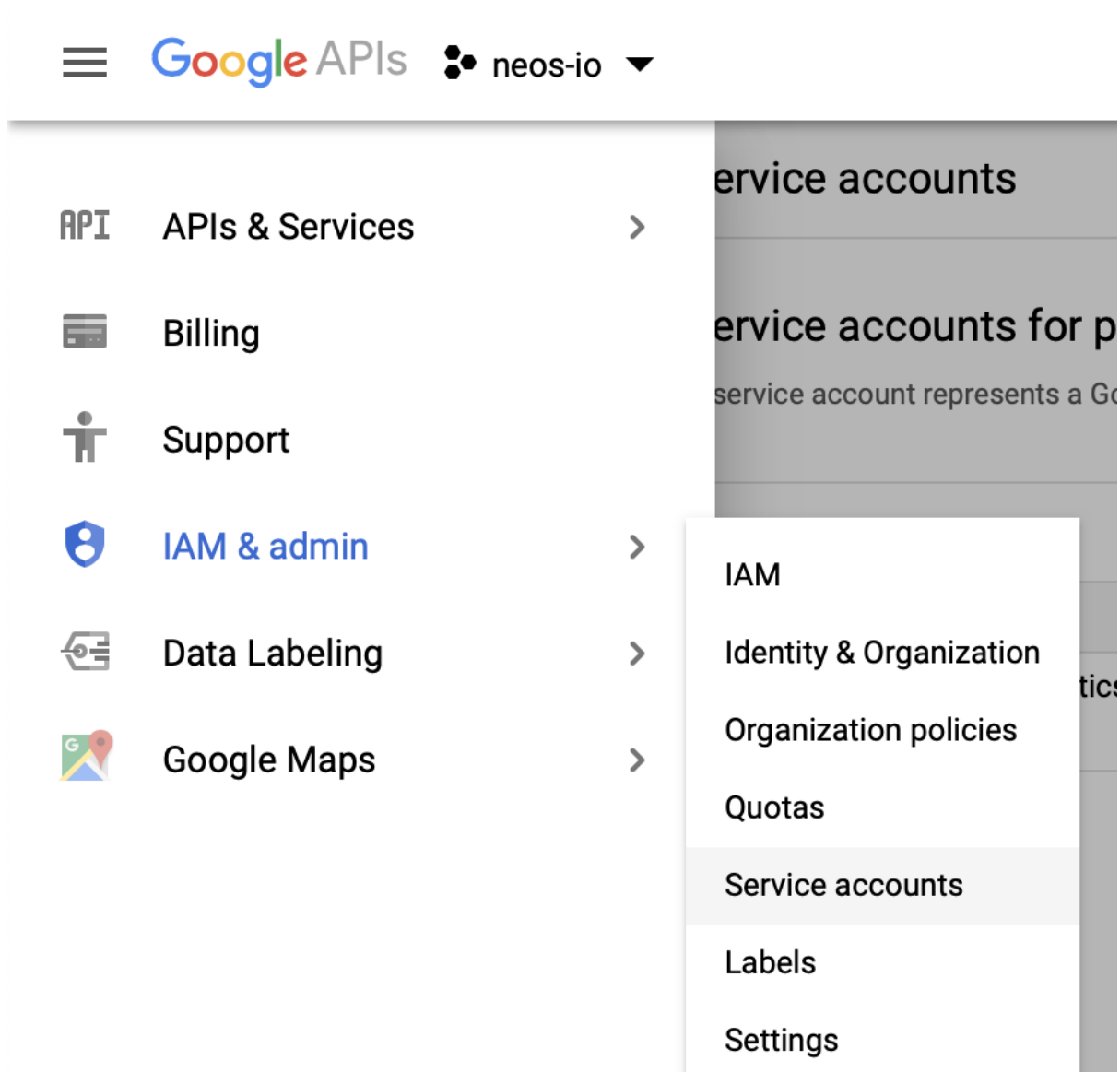
The Google Drive API allows clients to access resources from Google Drive



Gmail API
Google

Flexible, RESTful access to the user's inbox

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:

Google APIs neos-io

IAM & admin

1 Service account details — 2 Grant this service account access to project (optional)

Service account details

Service account name

Display name for this service account

Service account ID @neos-io.iam.gserviceaccount.com X ↺

Service account ID is required and must be unique.

Service account description

Describe what this service account will do

CREATE CANCEL

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”.

API APIs & Services

Credentials

Credentials OAuth consent screen Domain verification


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

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



Create service account key

Service account

Select...

Key type

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

☒ **JSON**
 Recommended

☐ **P12**
 For backward compatibility with code using the P12 format

Create

Cancel

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

Your credentials overview should now look like this:

API APIs & Services

Dashboard
Library
Credentials

Credentials

OAuth consent screen Domain verification

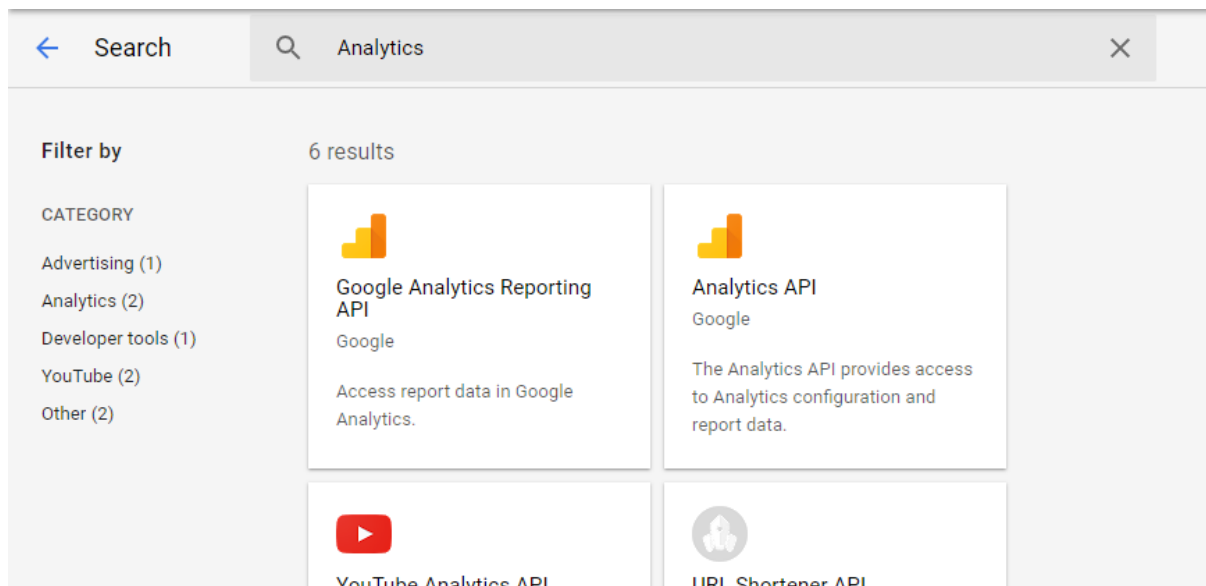
Create credentials
Delete

Create credentials to access your enabled APIs. For more information, see the [authentication documentation](#).

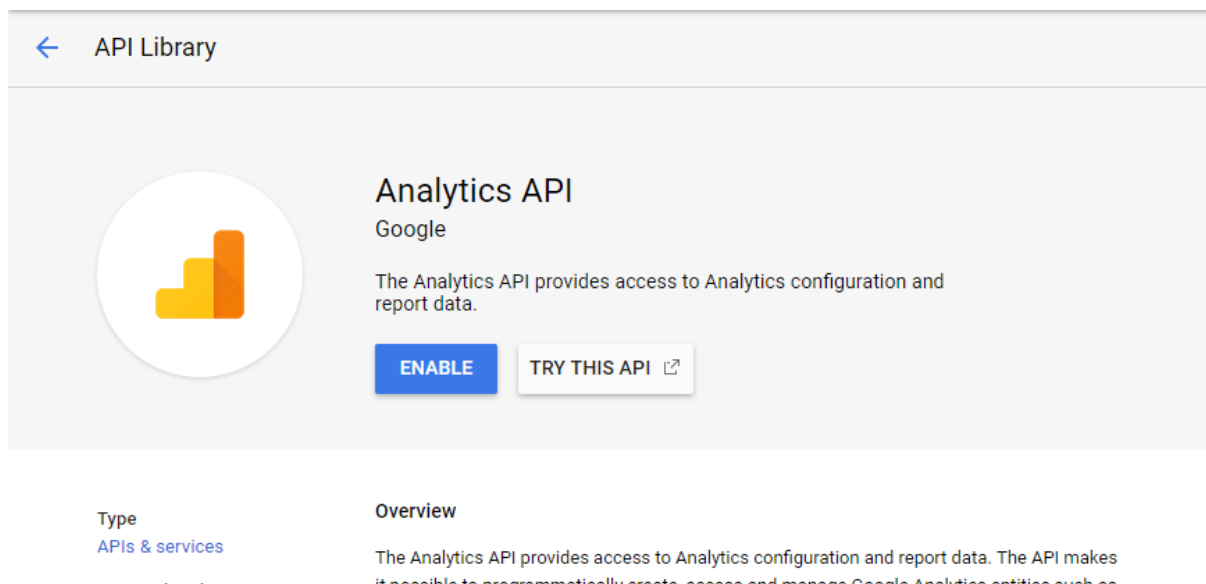
Service account keys

| ID | Creation date | Service account |
|--------------------------|---------------|-------------------------------|
| <input type="checkbox"/> | Jun 28, 2019 | Neos.io Analytics Integration |

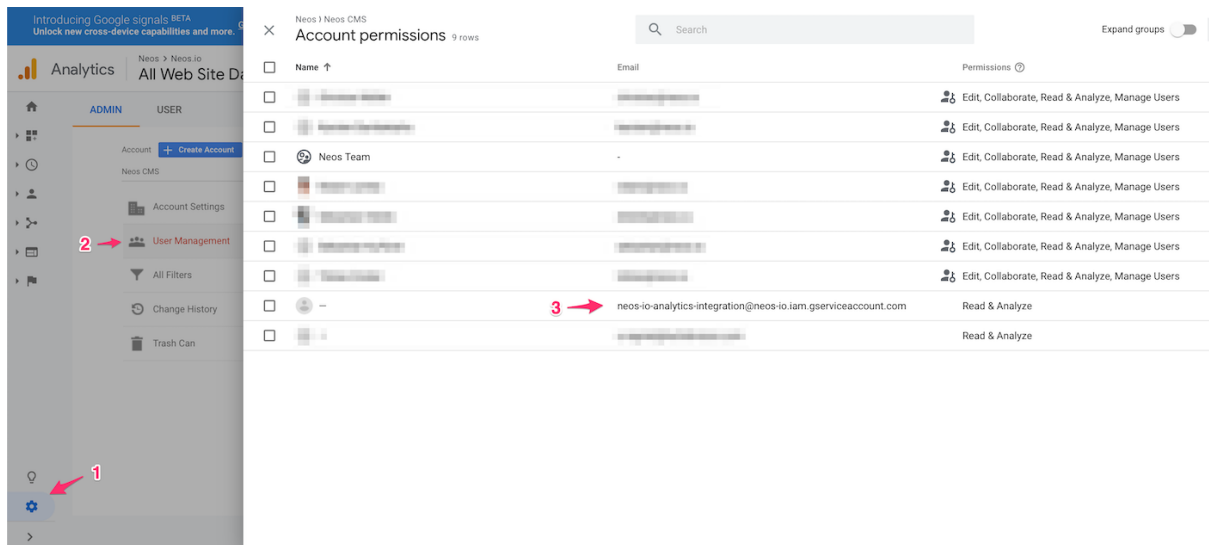
The last step in the Developers Console is to make sure the Google Analytics API is enabled. Click on *Library* in the left navigation and enter “Analytics” in the search field.



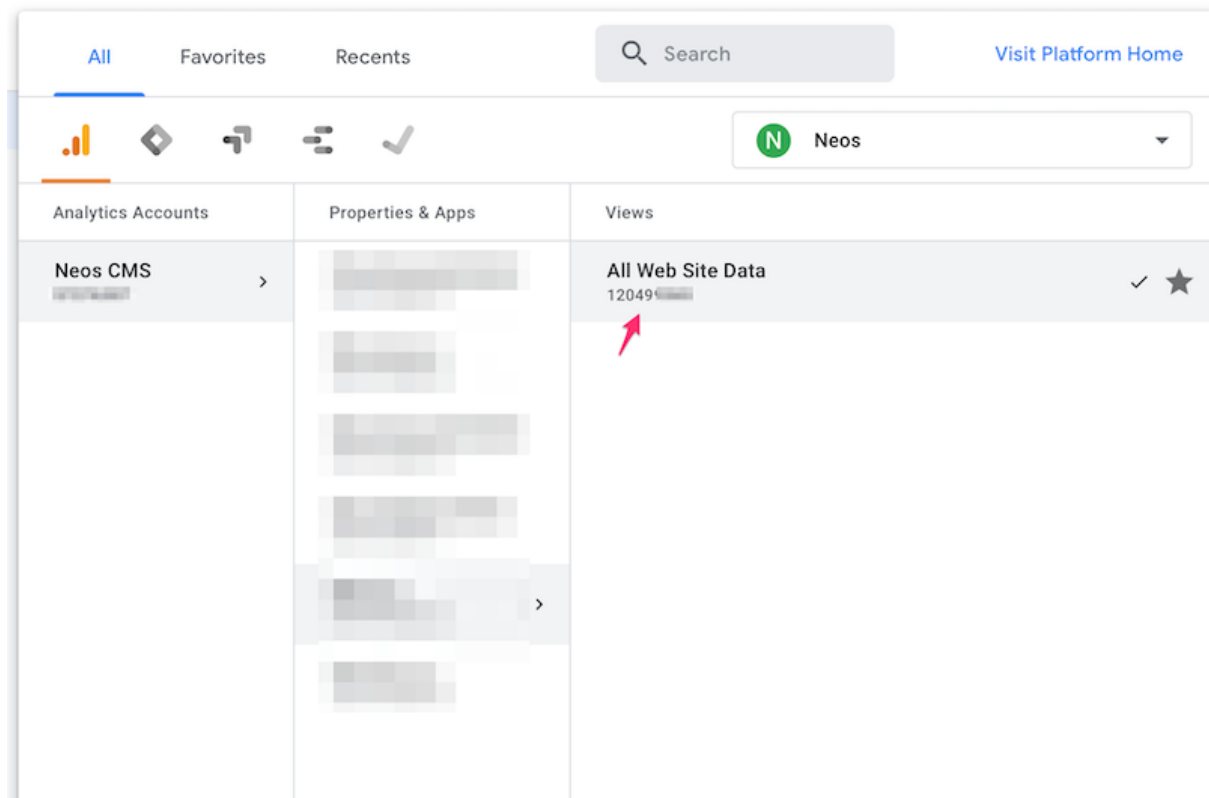
Click on the Analytics API link and on the following page on Enable.



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:

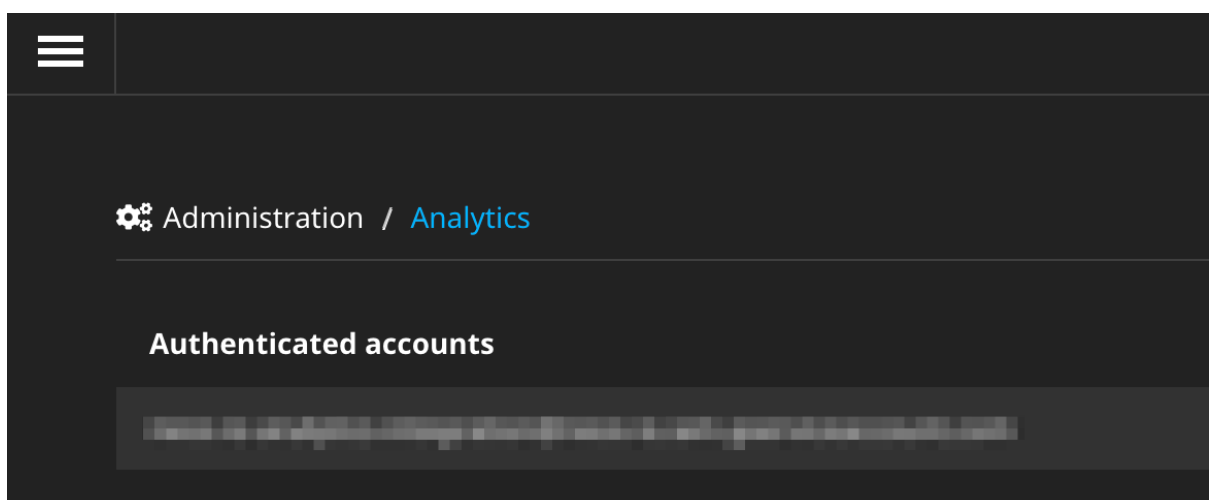
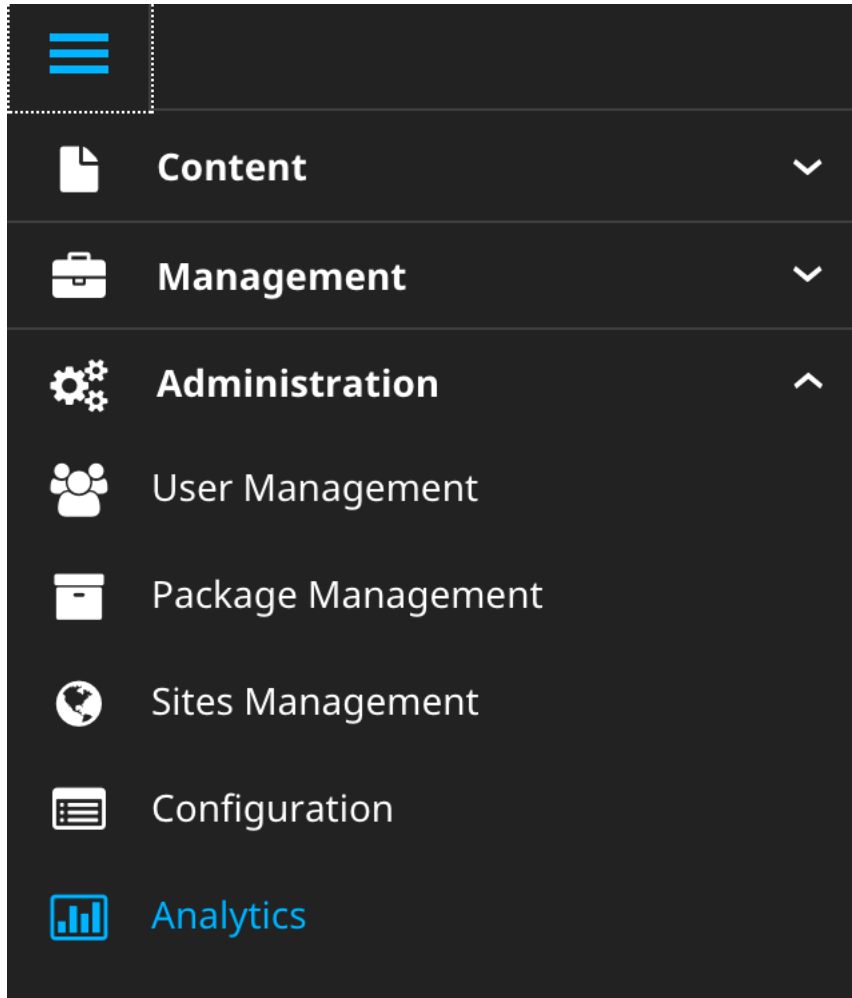
```
Neos:
  GoogleAnalytics:
    sites:
      neossitename:
        analytics:
          id: 'UA-XXXXXX-YY'
          profileId: 123456789
```

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.

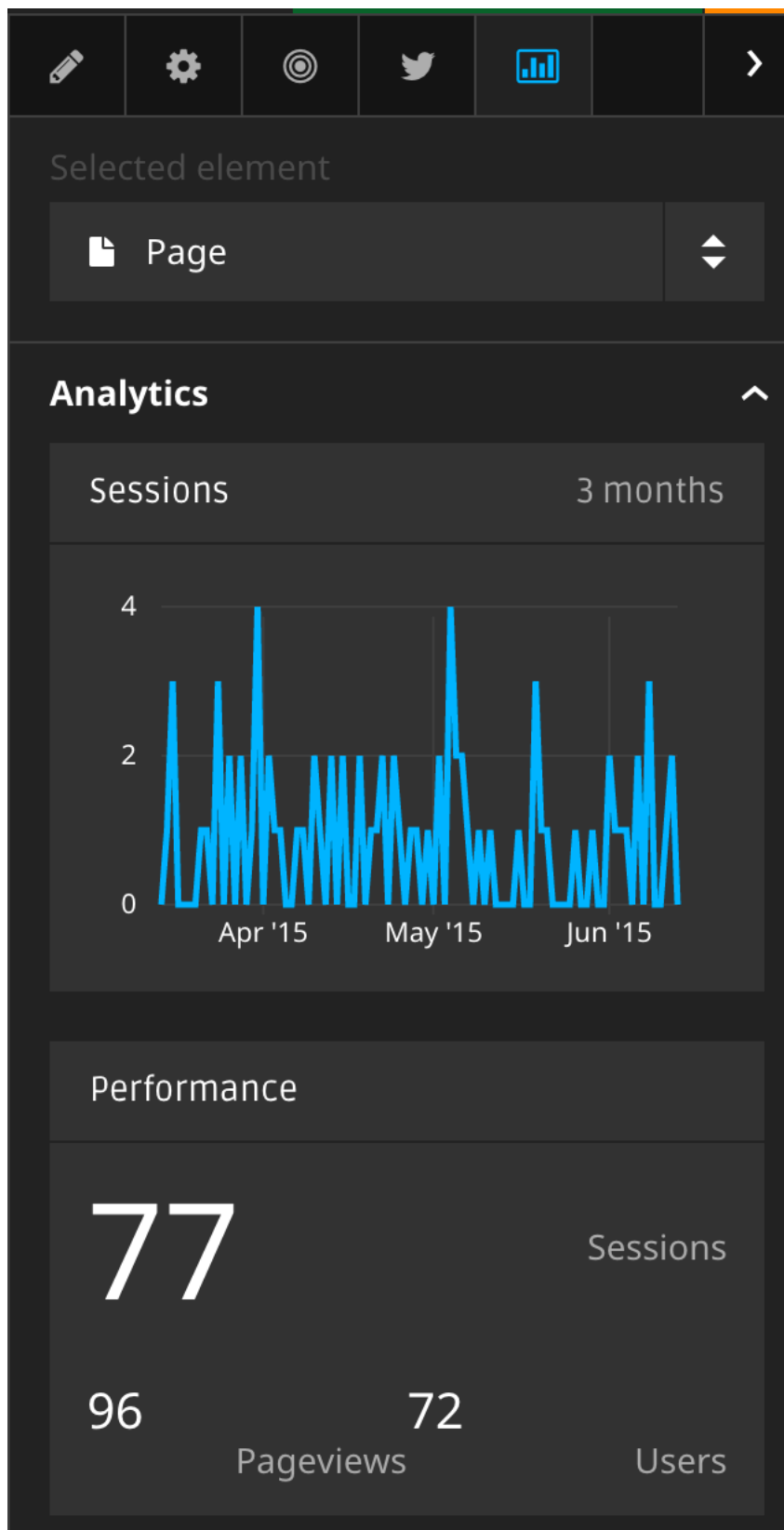


Now the integration is set up to fetch data and display the Analytics statistics in the Neos user interface.

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

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

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.

```
Neos:
  GoogleAnalytics:
    sites:
      neossitename:
        overrideHostname: 'example.org'
```

6.1 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 7

`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.

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.