Neos RedirectHandler Adapter

Release 4.3.x-dev

Neos Team and Contributors

Dec 01, 2023

CONTENTS

Whenever you change the *URL path segment* or move a document node, a redirect will automatically be generated as soon as it is published into the live workspace.

Note: To get an overview over all currently active redirects you can always run `./flow redirect:list`. For further details check the *Neos Command Reference*.

ONE

POSSIBLE CONFIGURATION FOR REDIRECTS

You can configure the default behaviour for automatically generated redirects within `Settings.yaml`.

```yaml
Neos:
 RedirectHandler:
  features:
    hitCounter: true
  statusCode:
    'redirect': 307
    'gone': 410
```

1.1 Options

hitCounter
    turn on/off the hit counter for redirects.

statusCode
    define the default status code for redirect or gone status (node deleted).

It is also possible to add, change or remove redirects within the CLI. The available CLI commands for custom redirect management can be found in the *Neos Command Reference*.

RESTRICT GENERATION

It is possible to restrict the generation of redirects to a certain node path or node type. For instance, you can use it in a multi site environment or avoid massive redirect generation if you don't need it.

The generation of redirects can be disabled by using one of the following three methods.

Note: Adding rules here with a value of *true* (e.g. *Neos.Neos:Document: true*) will add them to a blacklist for redirect generation. This might be confusing, because you could expect that you need to set a rule to *false*, which is not the case.

2.1 restrictByNodeType

Restrict redirect generation by node type.

```yaml
Neos:
 RedirectHandler:
   NeosAdapter:
    restrictByNodeType:
      Neos.Neos:Document: true
```

2.2 restrictByPathPrefix

Restrict redirect generation by node path prefix.

Note: No redirect will be created if you move a node within the restricted path or if you move it away from the restricted path. But if you move a node into the restricted path the restriction rule will not apply, because the restriction is based on the source node path.

```yaml
Neos:
 RedirectHandler:
   NeosAdapter:
     restrictByPathPrefix:
        - '/sites/neosdemo': true
```

2.3 restrictByOldUriPrefix

Restrict redirect generation by old URI prefix.

```
Neos:
  RedirectHandler:
    NeosAdapter:
      restrictByOldUriPrefix:
        '/some/uri/path': true
```

2.4 enableAutomaticRedirects

Allows completely disabling redirect generation.

Note: There might be edge cases where you need to disable redirect generation completely (comes in handy when using a dedicated subcontext). Redirect generation can slow down large node operations, imports etc.

```
enableAutomaticRedirects: false
```

EXPORTING & IMPORTING REDIRECTS

You can export a CSV file with redirects from the commandline with this command:

` flow redirect:export --filename my-export.csv `

You can also add the optional arguments *host* to only export redirects for one site and *includeHeader* to also render a row with column headers (enabled by default).

If you have the *Neos.RedirectHandler.Ui* [package](https://github.com/neos/redirecthandler-ui) installed you can also export the redirects via the UI in the redirect handler backend module.

3.1 Importing redirects

3.1.1 Via the CLI

You can import redirects from a CSV file in the commandline with this command:

` flow redirect:import --filename my-export.csv `

You can also add the optional argument *delimiter* to set a custom delimiter used in your CSV file.

After the import is done you will receive a protocol which will list all changes and errors, if any occurred.

The expected format is this:

` "Source Uri","Target Uri","Status Code",Host,"Start DateTime","End DateTime",Comment, Creator,Type my-source-uri,my-target-uri,307,,2019-06-28-13-11-00,2019-06-30-13-11-00, "New product",myName,manual `

It is recommended to first export your redirects and work on the resulting file, so you already have this structure. However, you can also create a CSV file yourself. The first row with headings is optional as are all columns except the first three *Source Uri*, *Target Uri* and *Status Code*. That means you can create a CSV file with 3 columns and without a header row and the import should work fine. If you use the other fields to, you must have them in the correct order or errors might occur.

No deletions will be done, when the imported CSV file doesn't contain existing redirects.

Deletions might occur when redirect chains are resolved. So a redirect from *A -> B* and a redirect from *B -> C* will be combined into one redirect *A -> C*. If you still need the redirect from *B -> C*, you might need to add that one afterwards again.

3.1.2 Via the UI

If you have the *Neos.RedirectHandler.Ui* [package](https://github.com/neos/redirecthandler-ui) installed you can also export the redirects via the UI in the redirect handler backend module.

The same restrictions and requirements apply as for the import via the CLI.

FOUR

REDIRECT GENERATION WHEN PUBLISHING VIA CLI

When publishing nodes from CLI f.e. via *flow workspace:publish* you might need additional configuration to make sure that redirects are properly generated. As commands don't have a http request, a fake http request has to be generated to make the url generation work. In this case the setting *Neos.Flow.http.baseUri* will be used. If this is not set *http://localhost* is used.

This should not be a problem in most cases as without an active domain for a site only relative redirects are generated. If a site has an active domain this on will be used to set the *Origin domain* for a new redirect.