

Internet.nl Dashboard Documentation

version

internet.nl

July 08, 2024

Contents

internet.nl dashboard documentation	1
1 Installation guide	1
1.1 Overview	1
1.1.1 What do you need	1
1.2 Installation	2
1.2.1 Running the dashboard (during development of 5.0)	2
1.2.2 Running the dashboard (when 5.0 is released)	2
1.2.3 Load up default configuration	2
1.2.4 Setting up the first user	2
1.2.5 Setting up scanning	3
1.3 Performing your first scan	5
1.4 Advanced configuration	7
1.4.1 Setting up e-mail notification after scanning	7
1.5 Background information	8
1.5.1 Application component overview	8
1.5.2 Creating your own unique look and feel	8
2 Dashboard User Management	9
2.1 Overview	9
2.1.1 Abstract	9
2.1.2 In practice	9
2.2 Management	10
2.2.1 Account Management	10
2.2.2 User Management	11
3 Email Templates	13
3.1 General information	13
3.1.1 Where can i find templates	13
3.1.2 What templates are available	13
3.1.3 How do i define templates in multiple languages	13
3.1.4 What template system is used	13
3.1.5 What do i need to watch out for	13
3.1.6 What are the pros and cons of the current approach	13
3.1.7 How to set up e-mail sending	14
3.2 Scan Finished mail	14
3.2.1 When is this mail sent	14
3.2.2 How to test this mail	14
3.2.3 scan_finished tags	14
3.2.4 Example template	16
Indices and tables	19

internet.nl dashboard documentation

1 Installation guide

This is a draft installation guide that will be ready in version 5.0 of the internet.nl dashboard.

The ticket for this issue: <https://github.com/internetstandards/Internet.nl-dashboard/issues/495>

1.1 Overview

Setting up the dashboard requires an operating system with docker. This setup was developed on mac and linux, on other platforms your mileage may vary.

This tutorial will help you setting up an internet.nl dashboard instance. With this tutorial you should be up and running within a few hours, gaining familiarity and confidence with with running a dashboard installation.

All configuration setup for your operating system is out of scope. This includes configuration of firewalls, security settings, user management and so on.

For general and paid support with installations, updates and managing installations: please send a support request to vraag@internet.nl.

1.1.1 What do you need

- A computer with git and docker installed
 - 4 cores and 8 gigabyte of ram recommended
- Access to the command shell to perform installations commands
- Credential access to a running internet.nl API instance
 - This can be the official internet.nl instance
 - For information about running a batch instance,
 - See: <https://github.com/internetstandards/Internet.nl/blob/main/documentation/README.md>
- Optional: a domain name and SMTP settings

1.2 Installation

Installation is mostly configuration work inside the dashboard. Some of the below steps will be automated when 5.0 is released.

1.2.1 Running the dashboard (during development of 5.0)

In the command shell, perform the following commands.

```
git clone https://github.com/internetstandards/Internet.nl-dashboard/
cd Internet.nl-dashboard
git checkout 50
docker compose up --build
```

After a short while your dashboard instance will be ready at :8000.

You can start and restart the application by running `docker compose up --build`, use `control+c` to stop.

1.2.2 Running the dashboard (when 5.0 is released)

Download the version you want to run. These can be downloaded from the releases page, here: <https://github.com/internetstandards/Internet.nl-dashboard/releases>

Releases from version 5.0 and over support docker compose. You can also download a release from the command line, with the following command:

```
mkdir dashboard && cd dashboard
wget https://github.com/internetstandards/Internet.nl-dashboard/archive/refs/tags/v5.0.tar.gz
tar -zxvf v5.0.tar.gz
docker compose up --build
```

After a short while your dashboard instance will be ready at :8000.

You can start and restart the application by running `docker compose up --build`, use `control+c` to stop.

1.2.3 Load up default configuration

This step will be automated before 5.0 is released. For now, run these commands to make sure default configuration is loaded. For this you need to know the name of the 'backend' docker container.

You can retrieve the name of the backend docker container by running `docker ps`. Below documentation assumes the name of this container is `internetnl-dashboard-backend-1`.

Load up the config:

```
docker exec -ti internetnl-dashboard-backend-1 dashboard loaddata dashboard_production_default
docker exec -ti internetnl-dashboard-backend-1 dashboard loaddata dashboard_production_example
docker exec -ti internetnl-dashboard-backend-1 dashboard loaddata dashboard_production_periodic
docker exec -ti internetnl-dashboard-backend-1 dashboard loaddata dashboard_production_default
docker exec -ti internetnl-dashboard-backend-1 dashboard loaddata dashboard_production_default
```

1.2.4 Setting up the first user

Create a new user:

```
docker exec -ti internetnl-dashboard-backend-1 dashboard createsuperuser
```

Associate that user to the default account, assuming the `createsuperuser` made user id 1:

```
docker exec -ti internetnl-dashboard-database-1 psql --user dashboard -c "update internet_nl_dashboard_dashboarduser set account_id=1 where user_id=1;"
```

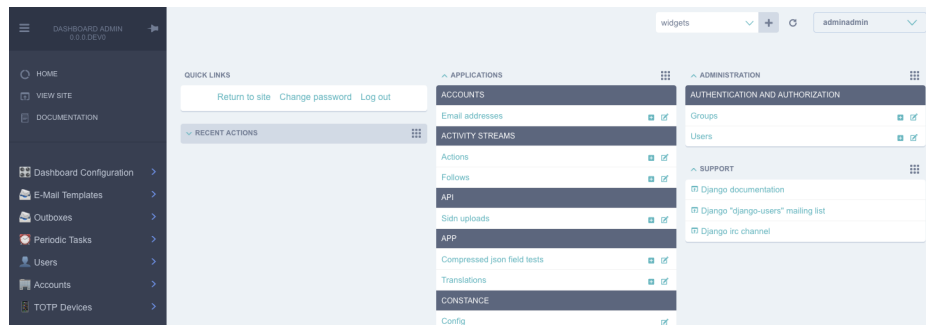
Now you can login at <http://localhost:8000/admin/>, or the same path under your server url.

If you get an error that a certain user already exists, this might not be your first attempt to install the dashboard via this method. The docker installation method shares the same database. Make sure to associate the newly created super user also is connected to an account. This can be performed with SQL and via the admin portal.

1.2.5 Setting up scanning

1. Visit the admin webpage on `/admin/` and log in. You might be redirected to the end-user website after logging in, so go to `/admin/` again.

After a successful login you will see something that looks like this:



2. Go to “**Dashboard Configuration**” in the left menu

3. Set up the “general dashboard settings”:

1. DASHBOARD_FRONTEND_URL
2. INTERNET_NL_API_URL
3. CREDENTIAL_CHECK_URL
4. INTERNET_NL_SCAN_TRACKING_NAME
5. Click the save button

NAME	VALUE	IS MODIFIED	DEFAULT
DASHBOARD_FRONTEND_URL Uri where the frontend is reachable on for end users. This uri is references in a few parts of the frontend.	<input type="text" value="https://192.168.2.42"/>	✓	https://dashboard.example.com Reset to default
INTERNET_NL_API_URL The internet address for the internet.nl API installation. This is commonly called a "batch server".	<input type="text" value="https://192.168.2.42/api/batch/v2"/>	✓	https://batch.example.com/api/batch/v2 Reset to default
CREDENTIAL_CHECK_URL The uri where internet.nl api credentials are checked. This is usually the bare INTERNET_NL_API_URL endpoint. This feature is used in the admin interface at account management. There the option 'check credentials' can be performed for each account.	<input type="text" value="https://192.168.2.42/api/"/>	✓	https://batch.example.com/api/ Reset to default
INTERNET_NL_SCAN_TRACKING_NAME This setting is used when sending API requests for tracking purposes. Setting this value make it clear who is sending API requests. A good setting contains something unique about this installation, such as an organization name. The maximum length is 40 characters.	<input type="text" value="My Organization"/>	✓	Reset to default

These settings can also be set using the command line, this saves some time clicking through the admin interface.

Use the following commands, of course with your own personal settings:

```
docker exec -ti internetnl-dashboard-backend-1 dashboard constance set DASHBOARD_FRONTEND_URL
docker exec -ti internetnl-dashboard-backend-1 dashboard constance set INTERNET_NL_API_URL h
docker exec -ti internetnl-dashboard-backend-1 dashboard constance set CREDENTIAL_CHECK_URL
docker exec -ti internetnl-dashboard-backend-1 dashboard constance set INTERNET_NL_SCAN_TRAC
```

4 . Setup the API credentials for the account.

- 1 . Go to the account management page
- 2 . http://127.0.0.1:8000/admin/internet_nl_dashboard/account/
- 3 . Click on the admin user
- 4 . Setup the “internet nl api username” and “new password” field and click save
- 5 . To test if the account was setup properly, use the ‘Check API credentials’
- 6 . If the credentials are correctly configured the check will return a checkmark symbol, otherwise a cross symbol will be visible on the row of this account.

The screenshot shows a form for managing an account. It has the following fields and controls:

- Name:** A text input field containing 'default_account'.
- Report settings:** A large text area containing 'null'. Below it, a small note says: 'This stores reporting preferences: what fields are shown in the UI and so on (if any other). This field can be edited on the report page.'
- Internet nl api username:** A text input field containing 'internetnl'. Below it, a small note says: 'Internet nl API Username'.
- New password:** A text input field. Below it, a small note says: 'Changing this value will set a new password for this account.'

At the bottom of the form are four buttons: 'PRINT', 'Save and add another', 'Save and continue editing', and 'X DELETE'.

The screenshot shows a table with the following columns: NAME, INTERNET NL API USERNAME, and CAN CONNECT TO INTERNET NL API. There is one row with the following data:

NAME	INTERNET NL API USERNAME	CAN CONNECT TO INTERNET NL API
default_account	internetnl	✓

Below the table, there is a search bar with a 'GO' button. Below the search bar, there are two buttons: 'Delete selected accounts' and 'Check API credentials'.

Note that the password in step 4.3 is being saved inside the database as an encrypted value. The key to that encryption was auto-generated using this setup. This key is stored inside a file, if you change it the currently stored passwords cannot be used anymore.

You are now set to perform your first scan.

1.3 Performing your first scan

1. Visit the dashboard frontend and log in.

2. Go to 'domains'
3. Create a new scan:
 1. Click on 'create new list'
 2. Enter the name for the list
 3. Set the list to scan both web and mail
 4. Save the list

4. Add domains to the list
 1. Click 'add domains'
 2. Add the domains in the text field
 3. Click 'add the above domains to the list'
 4. After adding successfully, click 'close'

Add domains

Add domains in the text field below:
internet.nl_dashboard.internet.nl

Close

Add the above domains to the list

Add domains

✓ Success!

New domains have processed, see the status report for details.
At Monday, 8 July 2024 (less than a minute ago).

2 domains added to this list. Take care: if the new domains are not shown in the list right away, wait a short while and reload the list.

My First List

Scan now

Configure

Delete

Last scan started at: Not scanned before.

Type of scan performed: all

Scan frequency: disabled

Not automatically shared

Add domains

Upload domains

Download domains

-- type to filter

Clear

Domain

Labels

internet.nl

dashboard.internet.nl

Page 1/1 of 2 domains.

These domains will be included in the scan. Eligibility for scanning is checked just before requesting the scan. The icons before each domain is based on archived information on the most recent reachability of services. This is checked and updated prior to every scan.

5 . start a scan by clicking 'scan now' and confirming the scan

Confirm to scan

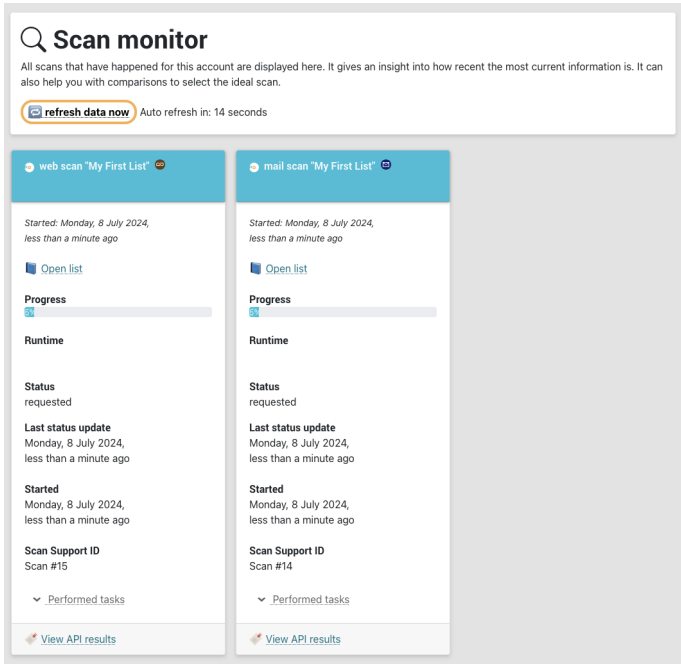
Click 'Scan now' to start scanning.

The scan is visible at the scan monitor. You can cancel a scan at any time in the scan monitor.

Cancel

Scan now


6 . On the *scan monitor* page the list of currently running scans can be seen.
After a scan has finished a report will be ready.




1.4 Advanced configuration

1.4.1 Setting up e-mail notification after scanning

After a scan completes it's possible to receive an e-mail. An SMTP server has to be configured.

1. Visit the admin interface on `/admin/` and log in.
2. In the sidebar click “ Outboxes”
3. Fill in the form with all SMTP details and click save. Only one outbox is needed. .. image:: installation/email_add_outbox.png

The image shows a form for configuring an email provider. It includes fields for 'Name' (pre-filled with 'My Email Provider'), 'EMAIL_USE_TLS' (checked), 'EMAIL_USE_SSL' (unchecked), 'EMAIL_SSL_KEYFILE', 'EMAIL_SSL_CERTFILE', 'EMAIL_HOST' (pre-filled with 'smtp.myexampleprovider.com'), 'EMAIL_HOST_USER' (pre-filled with 'my_username'), 'EMAIL_HOST_PASSWORD' (pre-filled with 'my_password'), 'EMAIL_PORT' (pre-filled with '587'), 'EMAIL_TIMEOUT' (pre-filled with '10'), and an 'Active' checkbox (checked).

The e-mails that are being sent are stored as templates in the “ E-Mail Templates” section. The default language for templates is English and several templates are pre-installed to be customized. For more information about these templates check the **email templates** chapter.

1.5 Background information

1.5.1 Application component overview

The dashboard contains of three application components: dockerfile, backend and frontend.

The dockerfile will setup a complete dashboard with sample users and sample configuration. This is one command and should be run on a system that runs docker, colima or another similar tool.

The backend is where all logic happens. List creation, result processing, session management and such. A few parts look like the frontend: the login page, the password reset page and spreadsheet upload page. All other pages are either redirects or interaction via JSON calls.

The frontend is the place where a users maintain lists, start scans, view and share reports. This is (mostly) the 'actual' website for day to day use.

1.5.2 Creating your own unique look and feel

The included default layout is an unbranded version of internet.nl, using the internet.nl styling. Only the logo's and references have been disabled. The setting for using your own template is called 'SITE_LAYOUT_NAME' and is exposed to the backend and frontend. The dashboard has not been optimized for custom branding yet, so your mileage to implement this for your organization may vary.

2 Dashboard User Management

2.1 Overview

The dashboard knows of three places that have to do with users:

- The API User
- The Account (which is a name for multiple users)
- The User

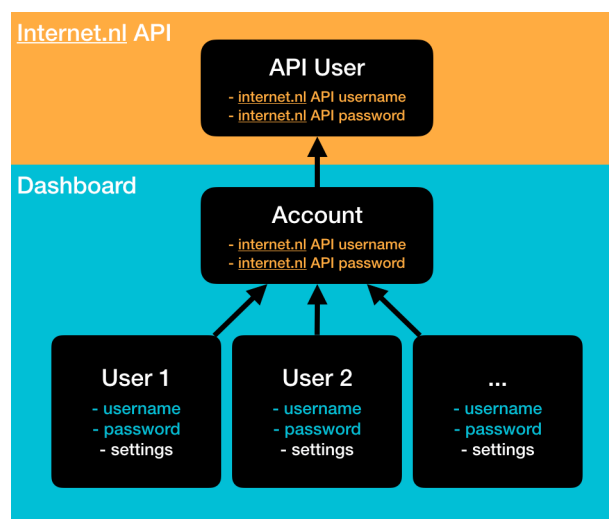
These connect to each other as shown in the following image.

Summarized:

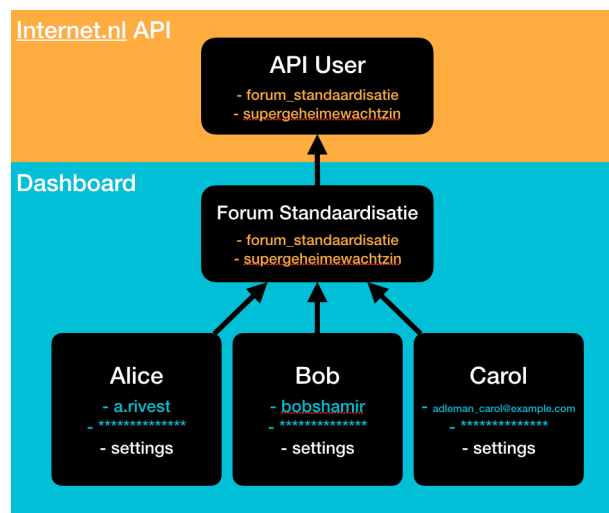
- An account can connect to the Internet.nl API using credentials.
- An account holds domains, report settings, scans and reports
- A User is part of an Account, there can be multiple users for an account
- A User has their own username, password and settings

We know that 'account' and 'user' is ambiguous.

2.1.1 Abstract



2.1.2 In practice

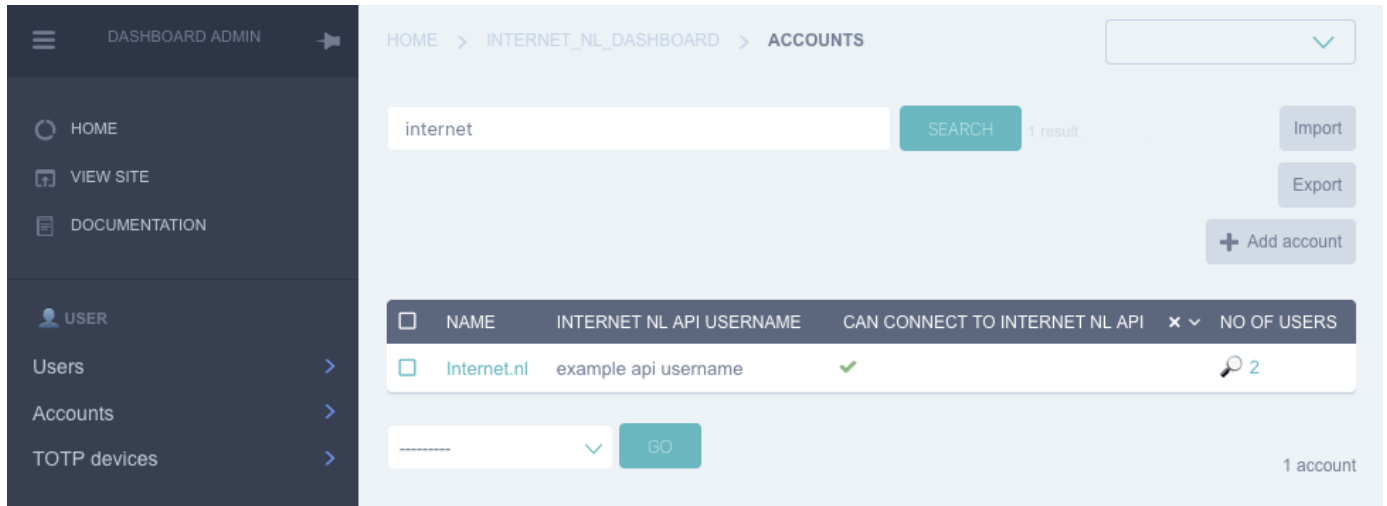


2.2 Management

User and account management in the dashboard is done via the Dashboard app and the Django Admin Interface. The latter is a bare bones data management application that allows editing data directly onto the database. The user experience is good enough for system administrators that need ultimate power and flexibility. With great power comes great responsibility.

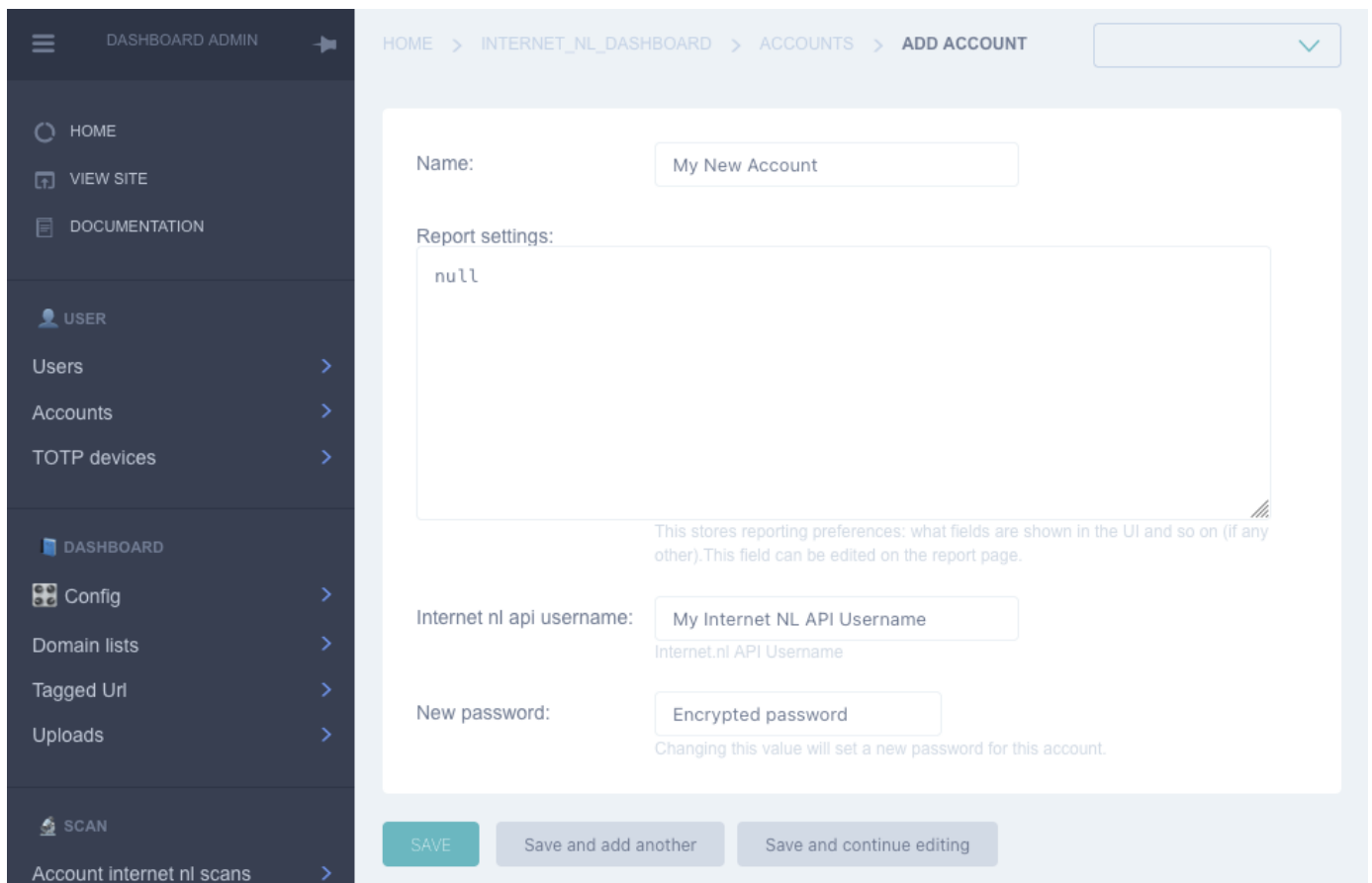
2.2.1 Account Management

Accounts can be created using the accounts menu. Accounts contain very little information. You will see report settings, which don't have to be altered. Just fill out the name and the Internet.nl API credentials.



The screenshot shows the 'ACCOUNTS' page in the Dashboard Admin interface. The left sidebar contains navigation links: HOME, VIEW SITE, DOCUMENTATION, USER, Users, Accounts, and TOTP devices. The main content area has a breadcrumb trail: HOME > INTERNET_NL_DASHBOARD > ACCOUNTS. A search bar contains the text 'internet' and shows '1 result'. There are buttons for 'Import', 'Export', and '+ Add account'. Below the search bar is a table with the following columns: NAME, INTERNET NL API USERNAME, CAN CONNECT TO INTERNET NL API, and NO OF USERS. The table contains one row: 'Internet.nl' with 'example api username' and a green checkmark in the 'CAN CONNECT' column, and '2' in the 'NO OF USERS' column. At the bottom right, it says '1 account'.

Here is an example of a filled in account. Notice how passwords here are not hashed, but encrypted. That's because they need to be sent to the API in a way that the API can read it. These credentials are sent over TLS to the API.



The screenshot shows the 'ADD ACCOUNT' page in the Dashboard Admin interface. The left sidebar is the same as the previous screenshot. The main content area has a breadcrumb trail: HOME > INTERNET_NL_DASHBOARD > ACCOUNTS > ADD ACCOUNT. The form contains the following fields: 'Name:' with the value 'My New Account'; 'Report settings:' with the value 'null'; 'Internet nl api username:' with the value 'My Internet NL API Username'; and 'New password:' with the value 'Encrypted password'. Below the password field, there is a note: 'Changing this value will set a new password for this account.' At the bottom, there are three buttons: 'SAVE', 'Save and add another', and 'Save and continue editing'.

2.2.2 User Management

User management is a bit more challenging. An account can have multiple users. That's what you see in the screenshot below: both users are in the 'Internet.nl' account.

The screenshot shows the 'USERS' page in the 'DASHBOARD ADMIN' interface. The breadcrumb trail is 'HOME > AUTHENTICATION AND AUTHORIZATION > USERS'. A search bar contains 'internet', and filters are set to 'By active' (dropdown), 'By staff status' (dropdown), and 'By superuser status' (dropdown). A 'SEARCH' button shows '2 results'. On the right, there are 'Import', 'Export', and '+ Add user' buttons. Below the filters is a table with the following data:

<input type="checkbox"/>	USERNAME x	IN ACCOUNT	FIRST NAME	LAST NAME	EMAIL ADDRESS	ACTIVE	STAFF STATUS
<input type="checkbox"/>	user_1	Internet.nl	First Name	Last Name		✓	✗
<input type="checkbox"/>	user_2	Internet.nl	First	Last	email@example.com	✓	✓

At the bottom, there is a 'GO' button and a status indicator '2 users'.

Editing a user reveals a long form. This form allows editing all the standard user information as well as user information specific to the dashboard experience.

In the top part of the form you can setup a username, password, first name, last name and e-mail.

Permissions only require the 'active' field. If you want to make somebody staff or superuser, they first have to set up their Second Factor authentication after logging in. No Groups of "User Permissions" need to be set up.

The last part of the form are the settings the user can edit when logged into the dashboard. Here you can specify under which account the user operates. You can leave the other fields empty, they can set that up themselves.

2 Dashboard User Management

HOME > AUTHENTICATION AND AUTHORIZATION > USERS > USER_1

< >

History

Username:*

user_1

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:

algorithm: pbkdf2_sha256 iterations: 216000 salt: ?k***** hash: tLc *****

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using [this form](#).

PERSONAL INFO

First name:

First Name

Last name:

Last Name

Email address:

PERMISSIONS

☒ Active

Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☐ Staff status

Designates whether the user can log into this admin site.

☐ Superuser status

Designates that this user has all permissions without explicitly assigning them.

Groups:

+

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions:

actstream | action | Can add action

actstream | action | Can change action

Specific permissions for this user. Hold down "Control", or "Command" on a Mac, to select more than one.

IMPORTANT DATES

Last login:

2021-09-07

11:13:53

Date joined:*

2020-01-30

18:44:47

+ Add another Dashboard...

Internet.nl/user_1

DASHBOARD USER: INTERNET.NL/USER_1

Account:*

Internet.nl

Mail preferred mail address:

reports@example.com

This address can deviate from the account mail address for other account features.

Mail preferred language:*

Netherlands

☒ Mail send mail after scan finished

After a scan is finished, an e-mail is sent informing the user that a report is ready.

Mail after mail unsubscribe code:

This is autofilled when sending an e-mail. The user can use mail_send_mail_after_scan_finished to false without logging out.

Notes:

3 Email Templates

The dashboard can send out mail when scans are finished. An email is composed using a template. These templates are rich in functionality, and can be adjusted without the need of re-building and re-deploying this software.

3.1 General information

3.1.1 Where can i find templates

Templates are located at `/admin/django_mail_admin/emailtemplate/` on your installation. They are part of the admin system and require the right permissions to be edited. An admin can edit them easily.

3.1.2 What templates are available

Currently there is one feature that uses an e-mail template, which is *scan_finished*. When a scan is finished, this mail template is used to draft a mail.

The scan finished template uses two sub-templates, which are:

- scan_finished
- - detailed_comparison_regression
 - detailed_comparison_improvement

The sub templates are tables that list regressions and improvements.

3.1.3 How do i define templates in multiple languages

Templates are postfixed with a language identifier. The default is `'_en'`, for English. The dashboard currently supports two languages, Dutch and English: `nl` and `en`.

If a template for `_nl` (or any other language) is missing, it will use `_en`.

(Technically there are many solutions to make sure translations are multilingual, most of which require re-compilation or maintenance of external translation files that require a build. When there are many languages in use, there will be a need to switch.)

3.1.4 What template system is used

The Django template system is used, with one exception: all fields from the dashboard are strings.

The Django template system is a mix of special tags and HTML, Javascript and CSS. It is also used in building front end systems which are rendered on the server side.

The Django Template system is documented here: <https://docs.djangoproject.com/en/3.1/ref/templates/language/>

3.1.5 What do i need to watch out for

Syntax mistakes can prevent a mail from being composed or sent. In this case the software will crash. This is the largest drawback of the current approach. When editing a template, be careful.

3.1.6 What are the pros and cons of the current approach

Pros:

- No need to recompile / rebuild / deliver the software when templates change
- Multi language in a simple and straightforward way (allowing changes per language)
- Uses well documented systems of django, html, javascript and css.

Cons:

- No syntax highlighting or checking of templates in the editor, use an external editor(!)

- A template per language is annoying when doing maintenance

3.1.7 How to set up e-mail sending

Email sending uses Django Mail Admin: https://github.com/Bearle/django_mail_admin/

You need to configure an outbox with proper values. You can test if this works via the command line with the command:

```
dashboard send_testmail
```

3.2 Scan Finished mail

3.2.1 When is this mail sent

This mail is only sent when the user has set that in their profile. You can review that for example here: </admin/auth/user/1/change/>

The fields that are used to send this mail are:

- Mail preferred mail address: an e-mail address
- Mail preferred language: a language from the list (only supporting NL or EN)
- Mail send mail after scan finished: a boolean value if the mail can be sent
- Mail after mail unsubscribe code: automatically generated

If all preconditions are correct, a mail will be queued as part of the scanning process, just before the scan is finished.

3.2.2 How to test this mail

A test mail can be sent at will on the admin page of AccountInternetNLScan, here: /admin/internet_nl_dashboard/accountinternetnlsan/

A mail can only be sent if a scan is finished and the mail preconditions and setup are correct.

When this mail has been queued, either wait for the periodic task that sends mail is performed (once a minute), or perform this via the command line (during development):

```
dashboard send_queued_mail --processes=1 --log-level=2
```

3.2.3 scan_finished tags

- `{{unsubscribe_code}}`

Allows an unsubscribe from a specific feed of mails. This code can be used without a login. The url this code is used for is for example:

```
{{dashboard_address}}/spa/#/unsubscribe?feed=scan_finished&unsubscribe_code={{unsubscribe_code}}
```

- `{{recipient}}`

The recipient of the email, which is, in order of fallback A) the first name, B) the last name, C) the username.

- `{{user_id}}`

The id of a user, which might be useful at some point.

- `{{list_name}}`

The name of the domain list that is being scanned.

- `{{report_id}}`

The mail is about a report. Using this number a link can be built to the report. For example:

```
{{dashboard_address}}/spa/#/report/{{report_id}}
```

3 Email Templates

- `{{report_average_internet_nl_score}}`

The average score in the report.

- `{{report_number_of_urls}}`

The total number of urls in this report.

- `{{scan_id}}`

The number of the scan performed, which might be useful for context and tracking purposes.

- `{{scan_started_on}}`

The date and time when the scan started, in ISO format.

- `{{scan_finished_on}}`

An approximation of when the scan is finished. This mail is sent as part of the scanning process, which is thus not yet finished. It might be off by a minute or two. This is also in ISO format.

- `{{scan_duration}}`

Number of seconds it took to complete a scan. Also an approximation.

- `{{scan_type}}`

Either web or mail. Can be used in sentences like:

The `{{scan_type}}` scan on `{{list_name}}` is finished.

- `{{previous_report_available}}`

If there is a previous report for this list. The value will be “True” if that is the case. Otherwise it will be “False”. Note that this is a string value, not a boolean value.

- `{{previous_report_average_internet_nl_score}}`

The average score of the previous report. This is used for easy overall comparison.

- `{{compared_report_id}}`

The id of the previous report, can be used to build a link with a comparison, such as:

`{{dashboard_address}}/spa/#/report/{{report_id}}/{{compared_report_id}}`

- `{{comparison_is_empty}}`

A string boolean containing either “True” or “False”. The comparison is empty when all values compared to the previous and current report are the same. There has been no change, at all.

If the comparison is empty, there is no need to show any details of course.

- `{{improvement}}`

The number of improvements made in the current report, compared to the last report.

- `{{regression}}`

The number of regressions in the current report, compared to the last report.

- `{{neutral}}`

The number of neutral values in the current report, compared to the last report. Neutral is either unchanged, or a comparison against an error, not-testable or other hard to compare value.

- `{{comparison_report_available}}`

A simple value to check if a comparison is available. Can be used to enable or disable sections of the email.

- `{{comparison_report_contains_improvement}}`

Set to “True” if there are improvements in the comparison. There might be only improvements and no regressions and vice versa.

- `{{comparison_report_contains_regression}}`

3 Email Templates

Set to "True" if there are regressions available.

- `{{days_between_current_and_previous_report}}`

The number of days between the current and previous report.

- `{{comparison_table_improvement}}`

This is a rendered section of html, based on the `detailed_comparison_improvement(_en)` template. To use pre-rendered html, use the following in your e-mail, using the word "safe":

`{{comparison_table_improvement|safe}}`

- `{{comparison_table_regression}}`

See `comparison_table_improvement`.

- `{{domains_exclusive_in_current_report}}`

A comma separated string of domains that are available in the current report, but not in the previous report. These are new domains that have been added to the list, usually. There are also edge cases where the domain could not be scanned last time, but it could this time.

- `{{domains_exclusive_in_other_report}}`

A comma separated string of domains that are only available in the previous report. Probably those domains have been deleted from the list of domains during the new scan.

- `{{dashboard_address}}`

The web address of the dashboard. This is configured in the settings at: `/admin/constance/config/`

3.2.4 Example template

`scan_finished_en` takes into account a multitude of situations where there are no scan results.

This template will probably be quickly outdated, but shows how to build a nice template with the fields above.

```
Hi {{recipient}},<br>
<br>
The {{scan_type}} scan on '{{list_name}}' has finished and your report is ready. The average
<br>
View the report at this link: <br>
<a href="{{dashboard_address}}/spa/#/report/{{report_id}}">
    {{dashboard_address}}/spa/#/report/{{report_id}}</a><br>

{% if previous_report_available == "False" %}
<br>
This is the first report for '{{list_name}}'. The next time this list is scanned, a comparis
{% endif %}

{% if previous_report_available == "True" and comparison_is_empty == "True" %}
<br>
A previous report, #{{compared_report_id}}, is available but contains no changes compared to
{% endif %}

{% if previous_report_available == "True" and comparison_is_empty == "False" %}
<br>
<h3>Changes compared to previous report</h3>
Below a summary is given compared to the previous report, #{{compared_report_id}}. The previ
<br>
You can view the comparison in detail on the dashboard at <a href="{{dashboard_address}}/spa
<br>
<h4>Summary of changes:</h4>
<table>
    <tr style='font-weight: bold; text-align: center;'>
```

3 Email Templates

```
        <td>{{improvement}}</td><td>{{regression}}</td><td>{{neutral}}</td>
    </tr>
    <tr>
        <td>Improvements</td><td>Regressions</td><td>Neutral</td>
    </tr>
</table>
{% endif %}
```

```
{% if previous_report_available == "True" and comparison_report_contains_improvement != "True" %}
<br>
Only neutral changes have been observed, therefore no detailed overview of changes is included.
{% endif %}
```

```
{% if comparison_report_contains_improvement == "True" or comparison_report_contains_regression == "True" %}
{% if comparison_report_contains_improvement == "True" %}
<br>
<h4>Overview of improvements:</h4>
<table style="">
    <tr>
        <th>Domain</th>
        <th>Score</th>
        <th>Improvement(s)</th>
        <th>Metrics improved</th>
    </tr>
    {{comparison_table_improvement|safe}}
</table>
{% endif %}
```

```
{% if comparison_report_contains_regression == "True" %}
<br>
<h4>Overview of regressions:</h4>
<table>
    <tr>
        <th>Domain</th>
        <th>Score</th>
        <th>Regeression(s)</th>
        <th>Metrics regressed</th>
    </tr>
    {{comparison_table_regression|safe}}
</table>
{% endif %}
```

```
{% endif %}
```

```
{% if domains_exclusive_in_current_report != "" %}
<br>
This report includes new domains, which are not included because they could not be compared:
{% endif %}
```

```
{% if domains_exclusive_in_other_report != "" %}
<br>
The following domains have disappeared in the new report, and are thus not included above:
{% endif %}
```

```
<br>
Regards,<br>
The internet.nl dashboard<br>
<br>
[
```

3 Email Templates

```
<a href="{{dashboard_address}}/spa/#/unsubscribe?feed=scan_finished&unsubscribe_code={{unsubscribe_code}}">unsubscribe</a>
-
<a href="{{dashboard_address}}/spa/#/account">preferences</a>
]

<style>
table th, table td{
    padding: 5px;
}
</style>
```

detailed_comparison_regression_en:

```
{% for record in data %}
<tr style='background-color: {% cycle 'rgba(0,0,0,.05)' 'inherit' %};'>
    <td style="vertical-align: baseline">
        {{ record.url }}
    </td>
    <td style="vertical-align: baseline">
        <a href="{{ record.new.report }}" target="_blank">
            
            {{ record.new.score }}%
        </a>
    </td>
    <td style="vertical-align: baseline">
        {{ record.changes.regression }}
    </td>
    <td>
        <ul>
            {% for metric in record.changes.regressed_metrics %}
                <li>{{ metric }}</li>
            {% endfor %}
        </ul>
    </td>
</tr>
{% endfor %}
```

detailed_comparison_improvement_en:

```
{% for record in data %}
<tr style='background-color: {% cycle 'rgba(0,0,0,.05)' 'inherit' %};'>
    <td style="vertical-align: baseline">{{ record.url }}</td>
    <td style="vertical-align: baseline">
        <a href="{{ record.new.report }}" target="_blank">
            
            {{ record.new.score }}%
        </a>
    </td>
    <td style="vertical-align: baseline">
        {{ record.changes.improvement }}
    </td>
    <td>
        <ul>
            {% for metric in record.changes.improved_metrics %}
                <li>{{ metric }}</li>
            {% endfor %}
        </ul>
    </td>
</tr>
{% endfor %}
```

Indices and tables

- `genindex`
- `modindex`
- `search`