django-reporting-endpoints Documentation

Release 0.9.4

Julius Seporaitis

Nov 06, 2023

CONTENTS:

1 Installation 3

2 Configuring django-reporting-endpoints 5

3 Example Project 7

4 Changelog 9

5 Indices and tables 11

Django app for receiving browser's Reporting API reports.

CHAPTER

ONE

INSTALLATION

1.1 Requirements

Python 3.8 to 3.12 supported.

Django 3.2 to 5.0 supported.

Step (5) below assumes that you are using django-csp.

1.2 Installation

1. Install with **pip**:

```
python -m pip install django-reporting-endpoints
```

2. Add django-reporting-endpoints to your `INSTALLED_APPS`:

```
INSTALLED_APPS = [
    ...,
    "reporting_endpoints",
    ...,
]
```

3. Configure your `REPORTING_ENDPOINTS`:

```
REPORTING_ENDPOINTS = {
    "csp-violations": "/_/csp-reports/",
}
```

4. Add the middleware:

```
MIDDLEWARE = [
    ...,
    "reporting_endpoints.middleware.ReportingEndpointsMiddleware",
    ...,
]
```

4. Add the view to your urlconfig:

```
urlpatterns = [
    ...,
    path(
        "_/csp-reports/", reporting_endpoint, kwargs={"endpoint_name": "csp-endpoint
↪"}
    ),
]
```

5. Configure your `report-to` directives for the packages you use. For example, if you use `django-csp`:

```
CSP_REPORT_TO = "csp-violations"
```

CHAPTER

TWO

CONFIGURING DJANGO-REPORTING-ENDPOINTS

Reporting-API is a working draft specification. Different browsers have different versions implemented. `django-reporting-endpoints` purpose was to cover the minimum configuration required to get it going.

Note: The chrome team's announcement of Reporting API has useful details on how to set up and debug reporting endpoints.

2.1 Policy Settings

These settings affect `Reporting-Endpoints` and `Report-To` headers respectively. They default to *empty* and are not sent, unless explicitly set in the settings.

Note: At the time of writing, reporting endpoints feature is available on Chromium based browsers only (e.g. Google Chrome). If you want to support multiple browsers, the recommendation is to configure both options.

REPORTING_ENDPOINTS

Set it to a concrete string `endpoint1-name="https://example.com/reports",endpoint2-name="/local/reports"`, the " (double-quotes) are significant. Alternatively, use a `dict`:

```
REPORTING_ENDPOINTS = {
    "endpoint1-name": "https://example.com/reports",
    "endpoint2-name": "/local/reports",
}
```

With the above example, you could use *endpoint1-name* and *endpoint2-name* in your *report-to* directives.

REPORT_TO_ENDPOINTS

This configuration option can take a string, a dict, or a list. The general format of a report-to endpoint group is the following:

```
{
    "group": "csp-violations",
    "max_age": 86400,
    "endpoints": [
       {"url": "https://example.com/reports"},
       {"url": "/local/reports"},
    ]
}
```

To set multiple groups, you can use a `list`:

```
REPORT_TO_ENDPOINTS = [
 {
     "group": "csp-violations",
     "max_age": 86400,
     "endpoints": [
       {"url": "/local/reports"},
     ]
 },
 {
     "group": "external-reporting",
     "max_age": 86400,
     "endpoints": [
       {"url": "https://example.com/reports"},
     ]
 },
]
```

Alternatively, if you have just a single group, you can pass a single `dict` on its own. The configuration option can take a string value - the serialized value of the header.

With the above example configuration options set, you could use *csp-violations* and *external-reporting* in your Content Security Policy *report-to* directives.

CHAPTER

THREE

EXAMPLE PROJECT

The django-reporting-endpoints repository contains an example project, demonstrating use of django-reporting-endpoints.

To run it locally:

1. You will need to install ngrok or similar.
2. Checkout the source code & install dependencies
3. Run migrations and setup a Django superuser.

```
python manage.py migrate
python manage.py createsuperuser
```

3. Run django server & proxy it via https.

```
python manage.py runserver
ngrok http 8000
```

4. Open the https address that ngrok gives you.

   The homepage is designed to trigger CSP violations and report them.

5. Open Django admin on 127.0.0.1:8000 and investigate the reports.

CHAPTER

FOUR

CHANGELOG

4.1 pending

- Make *disposition* an optional input field.

4.2 0.9.1-0.9.4 (2023-09-27)

- Improved test coverage.
- Improved documentation.
- Documentation build fix.

4.3 0.9.0 (2023-09-27)

- Initial release.

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search