django-dataexporter

unknown

May 15, 2021

CONTENTS

django-dataexporter is a extensible helper to export Django QuerySets and other data to CSV and Excel.

CHAPTER

ONE

FEATURES

- Exporter class to generate CSV and Excel files out of QuerySets and other iterables.
- Factory to generate Django ModelAdmin actions to trigger an export out of Django's famous admin interface.

CHAPTER

TWO

REQUIREMENTS

django-dataexporter supports Python 3 only and requires at least Django 2. In addition, the Python package `openpyxl` needs to be installed.

CHAPTER

THREE

PREPARE FOR DEVELOPMENT

A Python 3.6 interpreter is required in addition to poetry.

```
$ poetry install
```

Now you're ready to run the tests:

```
$ poetry run pytest
```

CHAPTER

FOUR

RESOURCES

- Documentation
- Bug Tracker
- Code

Contents:

4.1 Installation

- Install with pip:

```
pip install django-dataexporter
```

4.2 Usage

The API of the Exporter class is subject to change and therefore not yet documented.

4.2.1 Admin actions

The library provides admin action factories for csv and xlsx formats.

```
from django.contrib import admin
from django_dataexporter.admin import export_csv_action_factory, export_excel_action_
↪factory

from .models import User


class UserAdmin(admin.ModelAdmin):
    export_fields = ('id', 'first_name', 'last_name')
    actions = [
        export_csv_action_factory(fields=export_fields, header=True, label='Export as CSV
↪'),
        export_excel_action_factory(fields=export_fields, header=True, label='Export as␣
↪XLSX'),
    ]
```

The dataexporter classes can be also used inside an admin function. It is especially handy when defining a custom dataexporter class.

```
from django.contrib import admin
from django_dataexporter.csv import CsvExporter
from django_dataexporter.excel import ExcelExporter

from .models import User


class CustomExporter(CsvExporter):
    export_name = 'Custom'
    fields = ['first_name', 'last_name']
    field_header_verbose_names = {'last_name': 'Surname'}
    filename_extension = 'txt'

    def get_data_value(self, record, field):
        record = super().get_data_value(record, field)
        return record or 'Sorry, no data here.'


class UserAdmin(admin.ModelAdmin):
    export_fields = ('id', 'first_name', 'last_name')
    actions = ['csv_export', 'excel_export', 'custom_export']

    def csv_export(self, request, queryset):
        exporter = CsvExporter(fields=self.export_fields, header=True)
        return exporter.get_http_response(request, queryset)
    csv_export.short_description = 'Export surveys as CSV'

    def excel_export(self, request, queryset):
        exporter = ExcelExporter(fields=self.export_fields, header=True)
        return exporter.get_http_response(request, queryset)
    excel_export.short_description = 'Export surveys as XLSX'

    def custom_export(self, request, queryset):
        exporter = CustomExporter()
        return exporter.get_http_response(request, queryset)
    custom_export.short_description = 'Export name and surname only'
```

4.3 Changelog

4.3.1 1.0.0 (2021-05-14)

- Add support for Python 3.7, 3.8, 3.9
- Drop support for Django < 2.2

4.3.2 0.0.3 (2021-02-12)

- Add support for Django 3

4.3.3 0.0.2 (2019-05-03)

- Improve documentation
- Fix bug when using the provided admin factories twice in the same ModelAdmin

4.3.4 0.0.1 (2019-02-05)

- Initial release of *django-dataexporter*

Api documentation:

4.4 API Reference

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search