
django-listing Documentation

Release 0.0.7

Eric Lapouyade

Jul 15, 2020

CONTENTS

1	Installation	3
2	Usage	5
3	Demo	7
4	Indices and tables	9

Django app for building HTML listings / tables

INSTALLATION

Install with pip:

```
pip install django_listing
```

Then declare the app in your settings.py

```
INSTALLED_APPS = [  
    ...  
    'django_listing',  
]
```


USAGE

The very basic setting is to use create a view from the `TemplateView` class.

in `views.py`:

```
from django.views.generic import TemplateView
from demo.models import Employee

class BasicUsageListingView(TemplateView):
    template_name = 'demo/basic_usage.html'
    extra_context = dict(employees_as_model=Employee) # See 'Employee' definition in
    ↪ "Read me first" at home page.
```

In this example, `Employee` is a model and is exported as `employees_as_model` into the template `demo/basic_usage.html`. If you want, you can also use `get_context_data()` method instead of `extra_context` attribute.

As usual, you attach a view to an url in `url.py`:

```
from .views import *

urlpatterns = [
    ...
    path('basic_usage/', BasicUsageListingView.as_view(),
        name='basic_usage'),
    ...
]
```

It is time to define the template `demo/basic_usage.html`:

```
{% load django_listing %}
<html>
<head>
...
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/
    ↪bootstrap.min.css"/>
{% include "django_listing/header.html" %}
...
</head>
...
{% render_listing employees_as_model per_page=5 %}
...
<script src="http://code.jquery.com/jquery-3.3.1.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js
    ↪"></script>
```

(continues on next page)

(continued from previous page)

```
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js">
↪</script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-datetimepicker/2.5.20/
↪jquery.datetimepicker.full.min.js"></script>
{% include "django_listing/footer.html" %}
</body>
</html>
```

You need to load `django_listing` tags at the very begining of the template, then include `django_listing/header.html` template in the `<HEAD>...</HEAD>` part of the template, and `django_listing/footer.html` template at the very end of the `<BODY>...</BODY>` part of the template. Then you can use the `{% render_listing ... %}` tag where you want to display the listing. First parameter must be a Django model, a queryset, an iterable or a `Listing` instance. Some parameters can be added : here the listing will have 5 rows per page.

Note : jquery, popper, bootstrap and datetimepicker javascripts are not mandatory for basic listings

CHAPTER THREE

DEMO

Actually, the detailed documentation is under construction. The best way to see how to use django-listing is to see the demo code here : [django-listing-demo](#)

If you have docker you can run the demo with this command:

```
docker run -p 8123:8123 elapouya/django-listing-demo
```

And then open your browser at this url : <http://localhost:8123>

To install docker on Linux, just use this command:

```
curl -sSL https://get.docker.com/ | sh
```

Otherwise, you can upload from here : <https://docs.docker.com/get-docker/>

INDICES AND TABLES

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