
Django Migration Snapshots Documentation

Release 0.5.1

Lenders Cooperative

Dec 09, 2022

CONTENTS

1	Django Migration Snapshots	3
1.1	Documentation	3
1.2	Quickstart	3
1.3	Text Snapshot	4
1.4	Graphical Snapshot	5
1.5	Features	5
1.6	TODO's	6
1.7	Local Development	6
1.8	Deployment	6
1.9	License	6
2	Installation	7
3	Usage	9
4	Contributing	11
4.1	Types of Contributions	11
4.2	Get Started!	12
4.3	Pull Request Guidelines	13

Contents:

DJANGO MIGRATION SNAPSHOTS

Capture snapshots of your django project's migration history. These snapshots are represented as a directed graph using pygraphviz in both textual and graphical formats.

1.1 Documentation

The full documentation is at <https://django-migration-snapshots.readthedocs.io>.

1.2 Quickstart

Install Django Migration Snapshots:

```
pip install django-migration-snapshots
```

Add it to your INSTALLED_APPS:

```
INSTALLED_APPS = (  
    ...  
    "migration_snapshots",  
    ...  
)
```

1.2.1 1) Execute management command to create snapshot

```
# creates snapshot of entire migration history  
python manage.py create_snapshot  
  
# filter migrations before applied date (YYYY-MM-DD)  
python manage.py create_snapshot --date="2022-10-15"
```

1.2.2 2) Create object programmatically or from the admin panel

```
MigrationSnapshot.objects.create(output_format="pdf")
```

1.2.3 3) Automatically create migration snapshots with the *post_migrate* signal

```
from django.apps import AppConfig
from django.db.models.signals import post_migrate

def my_snapshot_callback(sender, **kwargs):
    # Create migration snapshot
    MigrationSnapshot.objects.create(output_format="pdf")

class MyAppConfig(AppConfig):
    ...

    def ready(self):
        # send signal only once after all migrations execute
        post_migrate.connect(my_snapshot_callback, sender=self)
```

1.3 Text Snapshot

```
digraph {
    "admin/0001_initial" -> "auth/0001_initial"
    "admin/0001_initial" -> "contenttypes/0001_initial"
    "admin/0002_logentry_remove_auto_add" -> "admin/0001_initial"
    "admin/0003_logentry_add_action_flag_choices" -> "admin/0002_logentry_remove_auto_add"
    "auth/0001_initial" -> "contenttypes/0001_initial"
    "auth/0002_alter_permission_name_max_length" -> "auth/0001_initial"
    ...
}
```

1.4 Graphical Snapshot



1.5 Features

- MigrationSnapshot data model
- Supported output formats
 - *BMP, CGIMAGE, DOT_CANON, DOT, GV, XDOT, XDOT12, XDOT14, EPS, EXR, FIG, GD, GIF, GTK, ICO, CMAP, ISMAP, IMAP, CMAPX, IMAGE_NP, CMAPX_NP, JPG, JPEG, JPE, JPEG_2000, JSON, JSON0, DOT_JSON, XDOT_JSON, PDF, PIC, PICT, APPLE_PICT, PLAIN_TEXT, PLAIN_EXT, PNG, POV_RAY, PS_PDF, PSD, SGI, SVG, SVGZ, TGA, TIF, TIFF, TK, VML, VMLZ, VRML, WBMP, WEBP, XLIB, X11*

- View migration history based on the migration's applied timestamp

1.6 TODO's

- Additional test coverage
- Setup tox
- Additional filters in management command (ie; per app, per model, etc.)
- More documentation

1.7 Local Development

```
make install  
make test
```

1.8 Deployment

```
make build  
make deploy
```

1.9 License

This project is provided under the [BSD License](#).

INSTALLATION

At the command line:

```
$ easy_install django-migration-snapshots
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv django-migration-snapshots  
$ pip install django-migration-snapshots
```


USAGE

To use Django Migration Snapshots in a project, add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (  
    ...  
    'migration_snapshots',  
    ...  
)
```

Add Django Migration Snapshots's URL patterns:

CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/Lenders-Cooperative/django-migration-snapshots/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

4.1.4 Write Documentation

Django Migration Snapshots could always use more documentation, whether as part of the official Django Migration Snapshots docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/Lenders-Cooperative/django-migration-snapshots/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *django-migration-snapshots* for local development.

1. Fork the *django-migration-snapshots* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/django-migration-snapshots.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv django-migration-snapshots
$ cd django-migration-snapshots/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 migration_snapshots tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.