Django Archive Documentation

Release 0.1.5

Nathan Osman

January 05, 2015

Contents

Django Archive provides a management command that will create a compressed archive of database tables and uploaded media.

CHAPTER 1

Contents

1.1 Installation

Django Archive is distributed as a Python package through PyPI.

1.1.1 PyPI Package

Installation on most platforms consists of running the following command:

```
pip install django-archive
```

1.1.2 Project Setup

Once the package is installed, it must be added to `INSTALLED_APPS`:

```
INSTALLED_APPS = (
    # ...
    'django_archive',
)
```

1.2 Usage

Interacting with Django Archive is done through a set of management commands.

1.2.1 Creating an Archive

To create an archive, use the `archive` management command:

```
python manage.py archive
```

This will create a compressed archive in the current directory containing a single fixture in JSON format and all uploaded media.

1.3 Settings

Django Archive provides a number of settings that can be used to customize its behavior. These settings are optional, but may be modified on a per-project basis in the project's `settings.py`.

1.3.1 ARCHIVE_DIRECTORY

Default: *empty*

Directory where the archive will be stored. The default behavior is to create the archive in the current directory.

1.3.2 ARCHIVE_FILENAME

Default: `'%Y-%m-%d--%H-%M-%S'`

String passed to `strftime()` to determine the filename of the archive.

1.3.3 ARCHIVE_FORMAT

Default: `'bz2'`

Format used for creating the compressed archive. The two options currently available include:

- `'bz2'`
- `'gz'`

1.3.4 ARCHIVE_EXCLUDE

Default:

```
(
    'contenttypes.ContentType',
    'sessions.Session',
    'auth.Permission',
)
```

List of models to exclude from the archive.

CHAPTER 2

Indices and tables

- *genindex*
- *modindex*
- *search*