django-not-eav Documentation

Release 0.0.1

James Pic

September 13, 2012

CONTENTS

This app attempts the impossible: implement a bad idea the right way.

CHAPTER

ONE

CONCEPT

django-not-eav provides a special model: Attribute.

When such a model is created:

- a model field is generated and added to the corresponding model class dynamically,
- a column is created, using south API,
- the admin is reloaded.

not_eav.autodiscover() is in charge of iterating over all Attribute to add the model fields to the model class dynamically.

You can add or remove fields in the admin, but you may not change a field name or type.

The advantage of this pattern over classic eav heavy solutions is that dynamic fields can be treated like regular fields.

The inconvenient is that it gives the website administrator the opportunity to mess up with his database.

In the end, this app enables the website administrator to become the developer of his own application.

This app will be the base requirement for the 3 upcomming apps, enabling the website administrator to:

- create his forms,
- create his list views,
- create his details views,

CHAPTER

TWO

REQUIREMENTS

- Python 2.7
- South
- Django 1.4+

CHAPTER

THREE

RESOURCES

You could subscribe to the mailing list ask questions or just be informed of package updates.

- Mailing list graciously hosted by Google
- Git graciously hosted by GitHub,
- Documentation graciously hosted by RTFD,
- Package graciously hosted by PyPi,
- Continuous integration graciously hosted by Travis-ci

CHAPTER

FOUR

DEMO

See test_project/README

CHAPTER FIVE

WHEN THINGS GO WRONG

If you need help, open an issue on the github issues page.

But make sure you've read how to report bugs effectively and how to ask smart questions.

Also, don't hesitate to do pull requests !

CHAPTER

SIX

FULL DOCUMENTATION

6.1 django-not-eav demo

Run the demo project and try to add dynamic attributes to Django.

The demo project lives in the `test_project` subdirectory of the repository.

6.2 test_project: basic features and examples

Virtualenv is a great solution to isolate python environments. If necessary, you can install it from your package manager or the python package manager, ie.:

```
sudo easy_install virtualenv
```

6.2.1 Install last release

Install packages from PyPi and the test project from Github:

```
rm -rf django-not-eav not_eav_env/

virtualenv not_eav_env
source not_eav_env/bin/activate
git clone https://github.com/yourlabs/django-not-eav.git
cd django-not-eav/test_project
pip install -r requirements.txt
./manage.py runserver
```

6.2.2 Or install the development version

Install directly from github:

```
NOT_EAV_VERSION="master"

rm -rf not_eav_env/

virtualenv not_eav_env
source not_eav_env/bin/activate
pip install -e git+git://github.com/yourlabs/django-not-eav.git@$NOT_EAV_VERSION#egg=not_eav
cd not_eav_env/src/not-eav/test_project
```

```
pip install -r requirements.txt
./manage.py runserver
```

6.2.3 Usage

- Run the server,
- Connect to */admin/*, ie. http://localhost:8000/admin/,
- Login with user "test" and password "test",
- Create an attribute on the book content type,
- Open add page for Book model,
- Use your new field

6.2.4 Database

A working SQLite database is shipped, but you can make your own ie.:

```
cd test_project
rm -rf db.sqlite
./manage.py syncdb --noinput
```

6.3 Quick start

The purpose of this documentation is to get you started as fast as possible, because your time matters and you probably have other things to worry about.

6.3.1 Quick install

Install the package:

```
pip install django-not-eav
# or the development version
pip install -e git+git://github.com/yourlabs/django-not-eav.git#egg=django-not-eav
```

Add to INSTALLED_APPS:

```
'not_eav'
```

Add before admin.autodiscover():

```
import not_eav
not_eav.autodiscover()
```

This will iterate over all `Attribute` objects, adding the dynamic fields to the corresponding models.

CHAPTER

SEVEN

INDICES AND TABLES

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