
Django Actual

Release b0.94.6

Sipmann

Dec 30, 2020

TOPICS:

1	Overview	1
2	Installation	3
2.1	Scaffolding	3
2.1.1	Field types	3
2.1.2	Example	4
2.2	License	5
2.3	Need Help	5
3	Indices and tables	7

OVERVIEW

This project is a follow up of the old project django-common from Tivix. The project served as a base to this one but it doesn't mean that we'll keep things like it was before.

Django-actual consists of the following things:

- A middleware that makes sure your web-app runs either on or without 'www' in the domain.
- A SessionManagerBase base class, that helps in keeping your session related code object-oriented and clean! See session.py for usage details.
- An EmailBackend for authenticating users based on their email, apart from username.
- Some custom db fields that you can use in your models including a UniqueHashField and RandomHashField.
- Bunch of helpful functions in helper.py
- A render_form_field template tag that makes rendering form fields easy and DRY.
- A dry response class: XMLResponse in the django_actual.http that can be used in views that give xml responses.

INSTALLATION

First run the pip install command like the following

```
pip install django-actual-helpers
```

Then add the *django_actual* app to your *INSTALLED_APPS* inside django *settings.py*.

2.1 Scaffolding

After installing and configuring the *django_actual* app, you can start scaffolding apps. First let's see the supported field types.

2.1.1 Field types

- char - CharField (default)
- text - TextField
- int - IntegerField
- decimal - DecimalField
- datetime - DateTimeField
- foreign - ForeignKey
- email - EmailField
- bool - BooleanField

To scaffold an app, you have the following syntax. Where you have an APPNAME (existing or not) and set the MODELNAME (must be a new one) followed by its fields.

```
python manage.py scaffold APPNAME --model MODELNAME [fields]
```

The fields must follow a strict pattern. Must be FIELDNAME:TYPE. Each field type has its own extended properties like *max length*, *required*, and *digits|precision*. The *char* type is the default one, so you can just write the field name (see the example below).

Two fields foreign and decimal requires additional parameters:

- “foreign” as the third argument takes foreign model, example:

```
blog:foreign:Blog, post:foreign:Post, added_by:foreign:User
```

NOTICE: All foreign key models must already exist in the project. User and Group models are imported automatically.

- decimal field requires two more arguments *max_digits* and *decimal_places*, example:

```
total_cost:decimal:10:2
```

- char field also have a length parameter on the third position
- all fields have a required as the last argument

```
total_cost:deciman:10:2:True name:char:50:True
```

2.1.2 Example

Let's create a simple forum app. We need Forum, Topic, and Post model.

Forum model Forum model needs just one field name:

```
python manage.py scaffold forum --model Forum name
```

Topic model Topics are created by site users so we need: created_by, title and Forum foreign key (update_date and create_date are always added to models):

```
python manage.py scaffold forum --model Topic created_by:foreign:User title_  
↪forum:foreign:Forum
```

Post model Last one are Posts. Posts are related to Topics. Here we need: title, body, created_by and foreign key to Topic:

```
python manage.py scaffold forum --model Post title body:text created_by:foreign:User_  
↪topic:foreign:Topic
```

All data should be in place!

Now you must add forum app to INSTALLED_APPS and include app in urls.py file by adding into urlpatterns:

```
urlpatterns = [  
    ...  
    path('forum', include('forum.urls')),  
]
```

Now syncdb new app and you are ready to go:

```
python manage.py makemigrations  
python manage.py migrate
```

Run your server:

```
python manage.py runserver
```

And go to forum main page:

<http://localhost:8000/forum/>

All structures are in place. Now you can personalize models, templates and urls.

At the end you can test the new app by running test:


```
python manage.py test forum

Creating test database for alias 'default'...
.....
-----
Ran 7 tests in 0.884s

OK
```

Happy scaffolding!

2.2 License

Copyright (c) 2011 Sipmann, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2.3 Need Help

INDICES AND TABLES

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