diafo Documentation

Release

Ashish Kumar

Mar 03, 2018

Contents

1 Requirements 3

2 Installation 5

3 Usage 7

4 Support 9

5 Bugs 11

Diafo is a Django package for generating dynamic forms during the runtime.

CHAPTER 1

Requirements

- **Python**: 3.4, 3.5, 3.6
- **Django**: 1.11
- **Django bootstrap form**: 1.4

CHAPTER 2

Installation

1. Install using pip:

```
pip3 install diafo
pip3 install django-bootstrap-form
```

2. Then add `'diafo'` to your `INSTALLED_APPS`.

```
INSTALLED_APPS = [
    ...
    'diafo',
    'bootstrapform',
]
```

3. Include the polls URLconf in your project urls.py like this:

```
url(r'^diafo/', include('diafo.urls')),
```

4. Run *python manage.py migrate* to create the diafo models.

CHAPTER 3

Usage

Diafo can be used for generating questionnaires consisting various questions during runtime. One can specify the question type like CharField, TextFied, ChoiceField, MultipleChoiceField, etc. Requirement can also be mentioned whether the question is compulsary or not.

Let see its usage through a simple survey webapp.

```
from diafo.models import Questionnaire

class Survey(models.Model):
    title = models.CharField(max_length=200)
    #link the form to your model
    questionnaire = models.OneToOneField(Questionnaire, null=True)
```

And then in your view you could do:

```
def new_survey(request):
    ...
    #get the title by using form or any how..
    questionnaire = Questionnaire.objects.create(name=tilte)
    survey = Survey.objects.create(title=title, questionnaire = questionnaire)
    return HttpResponseRedirect(reverse('diafo:admin_view', kwargs={'pk': survey.form.
↪pk}))
```

So now that you have questionnaire linked to your model,you can add questions and edit the questionnaire thru the diafo:admin_view, and let the user fill the questionnaire through diafo:user_view.

The diafo main URLs are as follows:

```
urlpatterns = [
    # for adding question,editing questionnaire detail,seeing responses,etc
    url(r'^admin/(?P<pk>[0-9a-f-]+)/$', views.admin_view, name='admin_view'),
    # for normal users to fill the questionnaire(no edit rights)
    url(r'^user/view/(?P<view_id>[0-9a-f-]+)/$',views.user_view, name = 'user_view'),
]
```

The Questionnaire model is as follows:

```
class Questionnaire(models.Model):
    name = models.CharField(max_length=200, null=True)
    # id used for admin purpose.. used as pk in admin view
    #id and pk are same since primary key is True
    id = models.UUIDField(primary_key=True, default=uuid.uuid4, editable=False)
    # view_id - used for normal user (can only see and fill the questionnaire)
    view_id = models.CharField(max_length=50,unique=True, default=uuid.uuid4)
    #specific settings (can be edited thru diafo admin view for given questionnaire)
    requires_sign_in = models.BooleanField(default = False, blank = True)
    collect_identity = models.BooleanField(default = False, blank = True)
```

This is how you can use use the diafo in your views or templates.

```
class yourview(request,your_parameters):
    ...
    # get the questionnare object.
    if want_to_go_on_admin_view:
        # use questionnare.pk and diafo:admin_view
        #for use in template href={% url 'diafo:admin_view' questionnaire.pk %}
        return HttpResponseRedirect(reverse('diafo:admin_view', kwargs={'pk
↪':questionnaire.pk}))
    elif want_a_rendered_form_for_user:
        # use questionnaire.view_pk and diafo:user_view
        #for use in template href={% url 'diafo:user_view' questionnaire.view_id %}
        return HttpResponseRedirect(reverse('diafo:admin_view', kwargs={'view_id
↪':questionnaire.view_id}))
```

CHAPTER 4

Support

If you have questions about usage or development you can contact me.

CHAPTER 5

Bugs

Really? Oh well.. Please Report. Or better, fix :)