

django自定义Color Picker控件

PYTHON, DJANGO,

0 Comments, March 21, 2013, 11:33 a.m.

版权所有, 转载请注明出处: <http://guangboo.org/2013/03/21/django-color-picker>

django提供了丰富的部件, 以满足我们对各种html控件的需求。并且如果有特别的要求, 我们还可以很容易的编写自己的控件, 本文主要内容是自定义一个颜色选择器的控件, 该控件是在input的基础上, 实现颜色的选择和展示。

如果你不知道如何实现自定义django的form控件的话, 可以查看django的源代码, 在文件django.forms.widgets.py中, 这就是开源的好处, 如果我们不知道该怎么入手, 可以看看它本身是如何实现的, “依葫芦画瓢”就是了。

因为我们的需求是实现颜色选择器的控件, 即ColorPicker, 其实这样的控件, google一下, 也能搜出一大堆的js库。为了方便我也在网上google到了一个较简单实用的color picker库, [jscolor](#)。该库就一个js文件, 没有其他的依赖库等, 并且为了方便, 我们也是直接引用该站点下的jscolor.js文件。

再则, 我们的model是使用charfield来表示color字段的, 在数据库中也是varchar类型, django默认的form控件是TextInput控件, 该控件需要我们手工在输入框中输入颜色的16进制值, 如#FF0000表示红色。而我们之所以使用jscolor库, 也是因为jscolor是使用的input[type='text']的控件, 因此这里我们在自定义django的form部件时, 从TextInput继承就可以了。代码如下:

```
# -*- coding:utf-8 -*-
from django import forms
from django.conf import settings
from django.utils import simplejson

class ColorPicker(forms.TextInput):
    def __init__(self, attrs = None, color_picker_attrs = None):
        if color_picker_attrs:
            cls = attrs.get('class', '')
            json = simplejson.dumps(color_picker_attrs)
            attrs.update({'class': '%s %s' % (cls, json).strip().replace(' ', '')})

        super(ColorPicker, self).__init__(attrs)

    def _media(self):
        js = [settings.COLOR_PICKER_JS_URL]
        return forms.Media(js = js)
    media = property(_media)
```

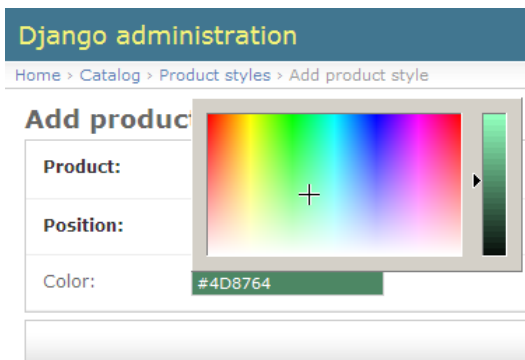
代码中的__init__构造函数, 与TextInput比起来多了一个color_picker_attrs参数, 该参数主要是为jscolor准备的, 主要是input控件的class属性值, 如: <input class="color {hash:true,caps:false}">, 具体可以参考[jscolor的文档](#)说明。另外更重要的是_media方法, 该方法返回一个forms.Media对象, 该对象用来往页面中输出js的引用代码, 并且需要添加media=property(_media), 来定义media属性。我们这里将js的定制定义在settings.py文件中, 命名为COLOR_PICKER_JS_URL, 并且值为http://jscolor.com/jscolor/jscolor.js。

那么该控件的使用, 就可以参考之前的文章《[django如何使ForeignKey字段显示树状结构](#)》中对CategoryAdmin类的定义方式, 重写formfield_for_dbfield方法, 来修改django默认生成的widget控件。代码如下:

```
class ColorStyleAdmin(admin.ModelAdmin):
    fields = ('product', 'position', 'color',)
    list_display = ('color',)

    def formfield_for_dbfield(self, db_field, **kwargs):
        if db_field.name == 'color':
            return db_field.formfield(widget = ColorPicker(attrs = {'class': 'color'}, color_picker_attrs = {'has': True, 'caps': False}))
        return super(ProductStyleAdmin, self).formfield_for_dbfield(db_field, **kwargs)
```

以上代码仅在此演示如何使用ColorPicker部件而用, 主要是formfield_for_dbfield的重写方式, 这里是将Model的color字段, 改变成color picker控件。并传递了hash, cap和required属性。运行效果如下图。



ZHANG, GUANGBO
APP SR. DEV
@STATE STREET
HANGZHOU, CHINA
@Weibo

LABELS

Office(5) Matlab(3)

UI(11) .NET(21)

ASP.NET(0)

PYTHON(28)

C/C++(1) XMPP(8)

DJANGO(10)

wxPYTHON(8)

Web(10) Others(10)

WCF(3) 互联网(5)

Google(3)

编程语言(2) WPF(2)

GMap.NET(1)

POST ARCHIVES

- Nov 2014(1)
- Oct 2014(1)
- Jun 2014(1)
- May 2014(3)
- Apr 2014(1)
- Mar 2014(2)
- Feb 2014(2)
- Jan 2014(17)
- Dec 2013(1)
- Oct 2013(5)
- Sep 2013(2)
- Aug 2013(1)
- Jun 2013(1)
- May 2013(11)
- Mar 2013(13)
- Feb 2013(8)
- Jan 2013(8)
- Dec 2012(3)
- Sep 2012(2)
- Aug 2012(2)
- Jul 2012(2)
- Jun 2012(5)

0

dongguangming ▾

Share  Favorite Post as dongguangming

Privacy

2/2