

Django Form表单内置字段详解

在上一节《[Django Form表单Field属性与方法](#)》，我们对 Field 基类的常用属性以及重点方法做了介绍，而在前述章节用到的 Form 表单的字段比如 CharField 即为 Field 的子类，而这些子类可以继承基类的所有属性及方法，不仅如此它们还定义了一些额外的属性用来限制对字段本身的赋值。下面就开始介绍 Django 表单系统的常用内置字段类型。

1. 常用表单字段类型

1) CharField

这个字段在 Django Form 表单中代表字符串类型的表单字段，是最常用的表单字段。它的 HTML 控件类型属性 widget 默认使用 TextInput 即表示普通文本输入框。它的源码构造函数如下所示：

```
1. class CharField(Field):
2.     def __init__(self, *, max_length=None, min_length=None, strip=True, empty_value='', **kwargs):
3.         self.max_length = max_length
4.         self.min_length = min_length
5.         self.strip = strip
6.         self.empty_value = empty_value
7.         super().__init__(**kwargs)
```

使用 max_length 与 min_length 分别可以限制字段值的最大值与最小值，对于不满足的条件字段值，将抛出 ValidationError 异常。strip 属性默认执行 Python 的 strip() 方法，可以除去字符串两端的空格。如果传递的是空值的时候，将会使用 empty_value 属性设定的值，如下所示：

```
1. In [1]: from django import forms
2. In [2]: title=forms.CharField(empty_value='C语言中文网')
3. In [3]: title.clean('')
4. Out[3]: 'C语言中文网'
```

2) IntegerField

它代表整数类型的表单字段，widget 控件默认使用 NumberInput。Django 中它的部分源码如下所示：

```

1. class IntegerField(Field):
2.     #控件类型
3.     widget = NumberInput
4.     default_error_messages = {
5.         'invalid': _('Enter a whole number.'),
6.     }
7.     #正则匹配模式
8.     re_decimal = re.compile(r'¥.0*¥s*¥')
9.     #构造函数
10.    def __init__(self, *, max_value=None, min_value=None, **kwargs):
11.        self.max_value, self.min_value = max_value, min_value
12.        if kwargs.get('localize') and self.widget == NumberInput:
13.            # Localized number input is not well supported on most browsers
14.            kwargs.setdefault('widget', super().widget)
15.        super().__init__(**kwargs)

```

max_value 与 min_value 用来限定取值范围，如果给出的值不在这两者的范围之内，那么就会抛出异常。如下所示：

```

1. In [5]: x=forms.IntegerField(max_value=5,min_value=1)
2. In [6]: x.clean('4')
3. #字符串强转为整数型4
4. Out[6]: 4
5. In [7]: x.clean('6')
6. ....
7. #抛出异常
8. ValidationError: ['确保该值小于或等于5。']

```

3) BooleanField

表示布尔类型的表单字段，该字段控件默认使用 CheckboxInput。默认为必填字段，若不填则会抛出异常，可以通过它的 required=False 属性来变为可选，即使不提供也不会抛出异常。

4) ChoiceField

表示选择类型的表单字段，默认空间使用 Select。它的构造函数如下所示：

```

1. class ChoiceField(Field):
2.     widget = Select
3.     default_error_messages = {
4.         'invalid_choice': _('Select a valid choice. %(value)s is not one of the available choices.'),
5.     }
6.
7.     def __init__(self, *, choices=(), **kwargs):
8.         super().__init__(**kwargs)
9.         self.choices = choices

```

它需要一个二元数组或者能够返回可迭代二元组的函数对象作为参数，它的使用方法如下所示：

```
1. In [28]: Gender= (('M', 'male'), ('F', 'female'))
2. In [29]: sex=forms.ChoiceField(choices=Gender)
3. In [30]: sex
4. Out[30]: <django.forms.fields.ChoiceField at 0x1435430>
```

在页面中 sex 字段对应的选择框中有两个可选值即 male 和 female。如果表单被提交，那么只有二元组的第一个元素，也就是 M 和 F 会被提交到后台服务器进行处理。

5) EmailField

它继承自子类 CharField，可以用验证电子邮件地址的格式是否正确。它的源码如下所示：

```
1. class EmailField(CharField):
2.     #默认控件
3.     widget = EmailInput
4.     default_validators = [validators.validate_email]
5.     def __init__(self, **kwargs):
6.         super().__init__(strip=True, **kwargs)
```

widget 默认使用 EmailInput。由于需要输入电子邮件的表单比较常见，如用户注册表单、身份验证表单等，所以，这个字段类型用在这些场景中会非常方便。这种验证特定结构的字段还有 UUIDField、GenericIPAddressField、URLField、SlugField 等，它们分别用来验证 UUID（通用唯一识别码）、IP 地址、URL 以及字符串。

6) DateTimeField

它是用于表示时间的表单字段，默认的 Widget 控件是 DateTimeInput，它接受一个可选参数 input_formats，这个参数是一个列表，列表中的元素可以转换为 Python 方法 datetime.datetime 的时间格式，默认可以接收的时间格式如下所示：

```
['%Y-%m-%d %H:%M:%S',      # '2019-1-5 11:15:57'
'%Y-%m-%d %H:%M',          # '2019-3-25 13:30'
'%Y-%m-%d',                 # '2019-12-5'
'%m/%d/%Y %H:%M:%S',        # '10/25/2006 14:30:59'
'%m/%d/%Y %H:%M',           # '10/25/2006 14:30'
'%m/%d/%Y',                 # '10/25/2006'
'%m/%d/%Y %H:%M:%S',        # '1/15/09 14:30:59'
'%m/%d/%Y %H:%M',           # '1/15/09 14:30'
'%m/%d/%Y',                 # '1/15/09']
```

DateTimeField 的 clean 方法接受的值类型可以是 datetime.datetime、datetime.date 或符合特定格式的字符串，最终会返回 datetime.datetime 对象或抛出异常。实例如下：

```
1. In [1]: from django import forms
2. In [2]: x=forms.DateTimeField(input_formats=[' 2019-1-5 11:12:12'])
3. In [3]: x
4. Out[3]: <django.forms.fields.DateTimeField at 0x6721b50>
```

7) ComboField

字段 ComboField 是一个验证字段。我们使用它可以规定字段的验证规则，它默认的 Widget 控件类型为 TextInput，它接收一个必选参数：fields，是一个用于验证字段值的字段列表，并按提供的参数顺序进行验证。实例如下：

```
1. In [18]: from django.forms import ComboField, CharField, EmailField
2. In [19]: f = ComboField(fields=[CharField(max_length=10), EmailField()])
3. In [20]: f.clean('caoxuesong@163.com')
4. ....
5. #报错异常
6. ValidationError: ['确保该变量包含不超过 10 字符（目前字符数 18）。']
```

Django 表单系统提供了多个诸如此类的验证字段，比如 MultipleChoiceField、SplitDateTimeField 等。在这里不做介绍，建议参阅官方文档了解即可。

本节介绍了 Django 表单系统的常用字段，这些字段自带的参数可以实现验证该字段输入值的目的，但是上述字段并没有涵盖所有的字段类型，如果大家感兴趣可以查阅官方文档《[Form fields](#)》进行学习，熟练掌握这些内容对于后续表单系统知识的学习将有很大的帮助。