



# Django form field custom widgets

Last Updated : 29 Dec, 2019

A widget is Django's representation of an HTML input element. The widget handles the rendering of the HTML, and the extraction of data from a GET/POST dictionary that corresponds to the widget. Whenever you specify a field on a form, Django will use a default widget that is appropriate to the type of data that is to be displayed. To find which widget is used on which field, see the documentation about [Built-in Field classes](#). This post revolves about the advanced use of widgets to modify the form structure and input type.

## Default Widget in Form Fields

Every field has a predefined widget, for example **IntegerField** has a default widget of **NumberInput**. Let's demonstrate this with help of our base project geeksforgeeks.

*Refer to the following articles to check how to create a project and an app in Django.*

- [How to Create a Basic Project using MVT in Django?](#)

---

Django Views Model Template Forms Jinja Python SQLite Flask Json Postman Interview Ques

Now let's create a demo form in "geeks/forms.py",

```
from django import forms

// creating a django form
class GeeksForm(forms.Form):
    title = forms.CharField()
    description = forms.CharField()
    views = forms.IntegerField()
```

```
available = forms.BooleanField()
```

Now to render this form we need to create the view and template which will be used to display the form to user. In `geeks/views.py`, create a view

```
from django.shortcuts import render
from .forms import GeeksForm

# creating a home view
def home_view(request):
    context = {}
    form = GeeksForm(request.POST or None)
    context['form'] = form
    return render(request, "home.html", context)
```

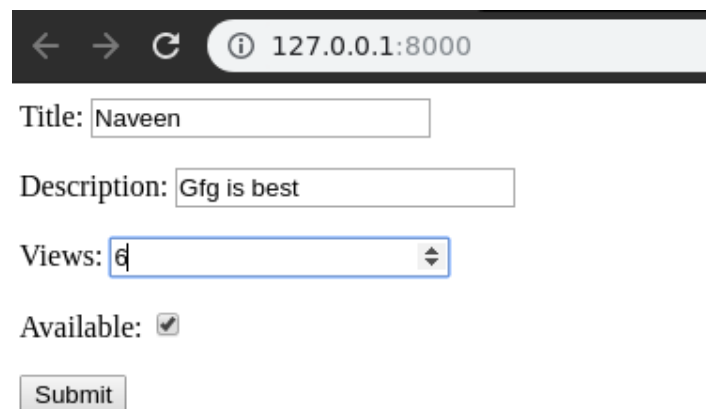
and in `templates/home.html`,

```
<form method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Submit">
</form>
```

Now let's display the form by running

Python `manage.py runserver`

visit <http://127.0.0.1:8000/>



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000`. The page contains a form with the following elements:

- Title:** A text input field containing the value "Naveen".
- Description:** A text input field containing the value "Gfg is best".
- Views:** A numeric input field with a spinner, containing the value "6".
- Available:** A checkbox that is checked.
- Submit:** A button labeled "Submit".

As seen in above screenshot, there is a different type of input field for IntegerField, BooleanField, etc. One can modify this using the following ways.

## Custom Django form field widgets

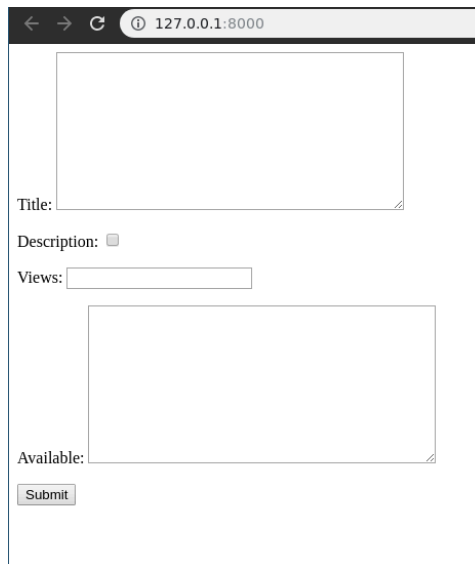
One can override the default widget of each field for various purposes. The list of widgets can be seen here – [Widgets | Django Documentation](#). To override the default widget we need to explicitly define the widget we want to assign to a field.

Make following changes to `geeks/forms.py`,

```
from django import forms

class GeeksForm(forms.Form):
    title = forms.CharField(widget = forms.Textarea)
    description = forms.CharField(widget = forms.CheckboxInput)
    views = forms.IntegerField(widget = forms.TextInput)
    available = forms.BooleanField(widget = forms.Textarea)
```

Now visit <http://127.0.0.1:8000/>,



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000`. The browser content shows a Django form with the following fields and widgets:

- Title:** A large text area widget.
- Description:** A checkbox widget.
- Views:** A text input widget.
- Available:** A large text area widget.

A **Submit** button is located at the bottom of the form.

Thus we can assign, any widget to any field using `widget` attribute. **Note** – The validations imposed on fields would still remain same, for example even if an `IntegerField` is made the same as `CharField`, it will only accept Integer inputs.

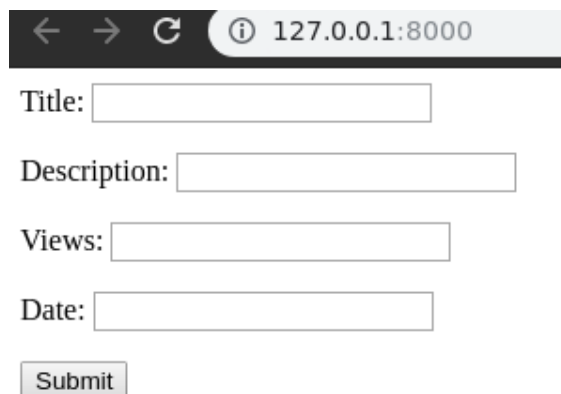
## Using Widgets to Customize DateField

widgets have a great use in Form Fields especially using Select type of widgets where one wants to limit the type and number of inputs form a user. Let's demonstrate this with help of modifying `DateField`. Consider `forms.py` as,

```
from django import forms

class GeeksForm(forms.Form):
    title = forms.CharField()
    description = forms.CharField()
    views = forms.IntegerField()
    date = forms.DateField()
```

By default, DateField as widget TextInput. It can be seen as



← → ↻ ⓘ 127.0.0.1:8000

Title:

Description:

Views:





Date:

Now let's change the widget for better and convenient input from user of a date. Add [SelectDateWidget](#) to DateField in forms.py,

```
from django import forms

class GeeksForm(forms.Form):
    title = forms.CharField()
    description = forms.CharField()
    views = forms.IntegerField()
    date = forms.DateField(widget = forms.SelectDateWidget)
```

Now input of date can be seen as very easy and helpful in the front end of the application. This way we can use multiple widgets for modifying the input fields.

    127.0.0.1:8000

Title:

Description:

Views:

Date: 

January ▼

1 ▼

2019 ▼

Submit

N Nave...



30

### Previous Article

Render HTML Forms (GET & POST) in Django

### Next Article

Initial form data - Django Forms

## Similar Reads

### Making a Django Admin Field Non-Required Without Custom Form Creation

The Django admin interface is a great tool for managing data in your Django application. Sometimes, you may need to make some fields optional in the admin...

2 min read

### Custom Field Validations in Django Models

This article revolves around how to add custom validation to a particular field. For example to add validation of an email to a CharField by specifying a particular...

3 min read

### Custom Field Validations in Django Forms

This article revolves around how to add custom validation to a particular field. For example to add validation of an email to a CharField by specifying a particular...

2 min read

## How to Add a Custom Field in ModelSerializer in Django

A key component of Django Rest Framework (DRF) is the Serializer, which converts complex data types like Django models into JSON, XML, or other conte...

4 min read

## Creating Custom Widgets in PyQt5

PyQt5 is one of the most advanced GUI library for Python. While being powerful, it's also well structured and makes it easy for you to build 'advanced' items....

2 min read

## help\_text - Django Form Field Validation

Built-in Form Field Validations in Django Forms are the default validations that come predefined to all fields. Every field comes in with some built-in validations...

4 min read

## initial – Django Form Field Validation

Built-in Form Field Validations in Django Forms are the default validations that come predefined to all fields. Every field comes in with some built-in validations...

4 min read

## label\_suffix - Django Form Field Validation

Built-in Form Field Validations in Django Forms are the default validations that come predefined to all fields. Every field comes in with some built-in validations...

4 min read

## label – Django Form Field Validation

Built-in Form Field Validations in Django Forms are the default validations that come predefined to all fields. Every field comes in with some built-in validations...

4 min read

## required - Django Form Field Validation

Built-in Form Field Validations in Django Forms are the default validations that come predefined to all fields. Every field comes in with some built-in validations...

4 min read

Article Tags :

PythonDjango-formsPython Django

Practice Tags :

python



Corporate & Communications Address:-  
A-143, 9th Floor, Sovereign Corporate  
Tower, Sector- 136, Noida, Uttar Pradesh  
(201305) | Registered Address:- K 061,  
Tower K, Gulshan Vivante Apartment,  
Sector 137, Noida, Gautam Buddh  
Nagar, Uttar Pradesh, 201305



Company

- About Us
- Legal
- In Media
- Contact Us
- Advertise with us
- GFG Corporate Solution
- Placement Training Program
- GeeksforGeeks Community

DSA

- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- Top 100 DSA Interview Problems
- DSA Roadmap by Sandeep Jain
- All Cheat Sheets

Languages

- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial
- Tutorials Archive

Data Science & ML

- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Web Technologies

- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- Bootstrap
- Web Design

Computer Science

- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths
- Software Development
- Software Testing

System Design

- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

School Subjects

- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar
- Commerce
- World GK

Python Tutorial

- Python Programming Examples
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question
- Django

DevOps

- Git
- Linux
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Inteivew Preparation

- Competitive Programming
- Top DS or Algo for CP
- Company-Wise Recruitment Process
- Company-Wise Preparation
- Aptitude Preparation
- Puzzles

GeeksforGeeks Videos

- DSA
- Python
- Java
- C++
- Web Development
- Data Science
- CS Subjects