



[Django](#) [Views](#) [Model](#) [Template](#) [Forms](#) [Jinja](#) [Python SQLite](#) [Flask](#) [Json](#) [Postman](#) [Interview Ques](#)

Django ModelForm – Create form from Models

Last Updated : 16 May, 2022

Django ModelForm is a class that is used to directly convert a model into a Django form. If you're building a database-driven app, chances are you'll have forms that map closely to Django models. For example, a User Registration model and form would have the same quality and quantity of model fields and form fields. So instead of creating a redundant code to first create a form and then map it to the model in a view, we can directly use ModelForm. It takes as an argument the name of the model and converts it into a Django Form. Not only this, ModelForm offers a lot of methods and features which automate the entire process and help remove code redundancy.

How to convert a model into a Django Form?

To explain the working of the project, we will use project **geeksforgeeks**, create a model and map it to Django forms.

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?](#)
- [How to Create an App in Django ?](#)

Now when we have our project ready, create a model in `geeks/models.py`,

Python3

```
# import the standard Django Model
# from built-in library
from django.db import models
```

```
# declare a new model with a name "GeeksModel"
class GeeksModel(models.Model):
    # fields of the model
    title = models.CharField(max_length = 200)
    description = models.TextField()
    last_modified = models.DateTimeField(auto_now_add = True)
    img = models.ImageField(upload_to = "images/")

    # renames the instances of the model
    # with their title name
    def __str__(self):
        return self.title
```

Before we create a model let's register our app in the main project. Go to `geeksforgeeks/settings.py` file and add `geeks` app in `INSTALLED_APPS` list. If we make migrations before this step it will display a message saying there are no changes made.

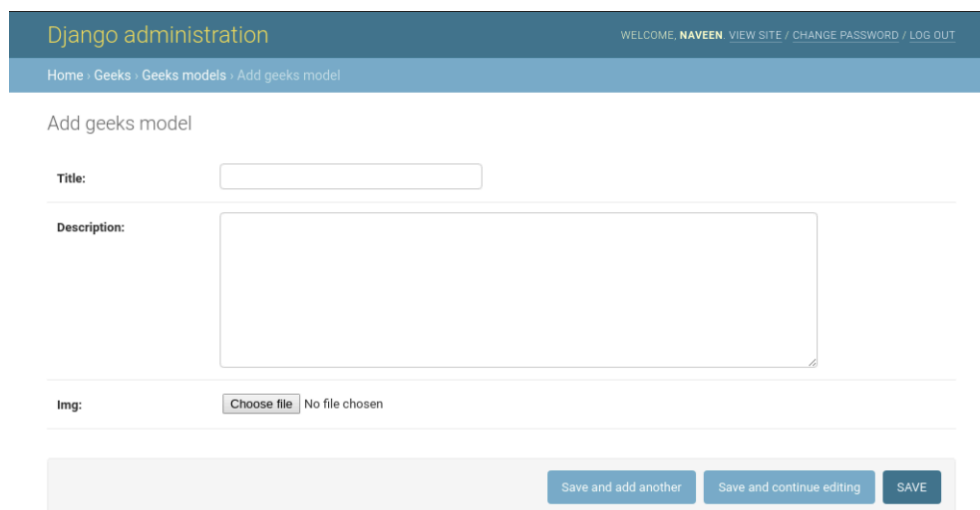
Now, run following commands to create the model,

```
Python manage.py makemigrations
```

```
Python manage.py migrate
```

We can check that model has been successfully created at

<http://127.0.0.1:8000/admin/geeks/geeksmodel/add/>,

The screenshot shows the Django administration interface. At the top, there's a header bar with 'Django administration' on the left and 'WELCOME, NAVEEN | VIEW SITE | CHANGE PASSWORD | LOG OUT' on the right. Below the header, a breadcrumb trail reads 'Home > Geeks > Geeks models > Add geeks model'. The main content area is titled 'Add geeks model'. It contains three form fields: 'Title:' with a text input, 'Description:' with a large text area, and 'img:' with a 'Choose file' button and the text 'No file chosen'. At the bottom right, there are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'.

To create a form directly for this model, dive into `geeks/forms.py` and Enter following code,

Python3

```
# import form class from django
from django import forms

# import GeeksModel from models.py
from .models import GeeksModel

# create a ModelForm
class GeeksForm(forms.ModelForm):
    # specify the name of model to use
    class Meta:
        model = GeeksModel
        fields = "__all__"
```

This form takes two arguments **fields** or **exclude**.

- **fields** – It is strongly recommended that you explicitly set all fields that should be edited in the form using the fields attribute. Failure to do so can easily lead to security problems when a form unexpectedly allows a user to set certain fields, especially when new fields are added to a model. Depending on how the form is rendered, the problem may not even be visible on the web page. Set the fields attribute to the special value ‘__all__’ to indicate that all fields in the model should be used.
- **exclude** – Set the exclude attribute of the ModelForm’s inner Meta class to a list of fields to be excluded from the form. For example:

```
class PartialAuthorForm(ModelForm):
    class Meta:
        model = Author
        exclude = ['title']
```

Finally, to complete our MVT structure, create a view that would render the form and directly save it to the database. In **geeks/views.py**,

Python3

```
from django.shortcuts import render
```

```
from .forms import GeeksForm

def home_view(request):
    context = {}

    # create object of form
    form = GeeksForm(request.POST or None, request.FILES or None)

    # check if form data is valid
    if form.is_valid():
        # save the form data to model
        form.save()

    context['form'] = form
    return render(request, "home.html", context)
```



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

- A "Title:" label followed by a text input field.
- A "Description:" label followed by a large text area.
- An "Img:" label followed by a "Choose file" button and the text "No file chosen".
- A "Submit" button at the bottom.

Everything is set, Now visit <http://127.0.0.1:8000/>, Now you can see that every model field is mapped into a form field and displayed correspondingly. Field mappings are discussed later in this article. So now let's try entering data into the form and check if it gets saved into the database.

127.0.0.1:8000

Title:

Description:

GFG is Best

Img:

Choose file

django mode...rm form.png

Submit

Hit submit and Bingo the form gets saved automatically to database. We can verify it at <http://localhost:8000/admin/geeks/geeksmodel/>.

Django administration

WELCOME, **NAVEEN** [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Geeks > Geeks models

Select geeks model to change

ADD GEEKS MODEL +

Action: Go 0 of 1 selected

☐ GEEKS MODEL

☒ **Naveen**

1 geeks model

Field types

The generated Form class will have a form field for every model field specified, in the order specified in the fields attribute. Each model field has a corresponding default form field. For example, a CharField on a model is represented as a CharField on a form. A model ManyToManyField is represented as a MultipleChoiceField. Here is the full list of conversions:

Model Field	Form Field
AutoField	Not represented in the form
BigAutoField	Not represented in the form

Model Field	Form Field
BigIntegerField	IntegerField with min_value set to -9223372036854775808 and max_value set to 9223372036854775807.
BinaryField	CharField, if editable is set to True on the model field, otherwise not represented in the form.
BooleanField	BooleanField, or NullBooleanField if null=True.
CharField	CharField with max_length set to the model field's max_length and empty_value set to None if null=True.
.DateField	DateField
DateTimeField	DateTimeField
DecimalField	DecimalField
DurationField	DurationField
EmailField	EmailField
FileField	FileField
FilePathField	FilePathField
FloatField	FloatField
ForeignKey	ModelChoiceField
ImageField	ImageField
IntegerField	IntegerField

Model Field	Form Field
IPAddressField	IPAddressField
GenericIPAddressField	GenericIPAddressField
ManyToManyField	ModelMultipleChoiceField
NullBooleanField	NullBooleanField
PositiveIntegerField	IntegerField
PositiveSmallIntegerField	IntegerField
SlugField	SlugField
SmallAutoField	Not represented in the form
TextField	CharField with widget=forms.Textarea
TimeField	TimeField
URLField	URLField

Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our [Mastering Django Framework - Beginner to Advanced Course](#) on GeeksforGeeks, designed for aspiring developers and experienced programmers. This comprehensive course covers everything you need to know about Django, from the basics to advanced features. Gain practical experience through **hands-on projects** and real-world applications, mastering essential Django principles and techniques. Whether you're just starting or looking to refine your skills, this course will empower you to build sophisticated web applications efficiently. Ready to enhance your web development journey? Enroll now and unlock your potential with Django!

Previous Article

[Django ModelFormSets](#)

Next Article

[{{ form.as_table }} - Render Django Forms as table](#)

Similar Reads

How to Filter ForeignKey Choices in a Django ModelForm

A ForeignKey field allows one model to relate to another in Django. When this field is represented in a form (e.g., a Django ModelForm), it typically displays a...

7 min read

Django Models | Set - 1

Prerequisites : Django Creating apps Models - According to Django Models, A model is the single, definitive source of information about your data. It contains...

2 min read

Django Models | Set - 2

Model Fields - Model fields define the datatype which will be stored in the provided variable. To store price, the integer type is more suitable. To store heig...

2 min read

Python | Relational fields in Django models

Prerequisite: Django models Django models represent real-world entities, and it is rarely the case that real-world entities are entirely independent of each other...

4 min read

BigIntegerField - Django Models

BigIntegerField is a 64-bit integer, much like an IntegerField except that it is guaranteed to fit numbers from -9223372036854775808 to...

4 min read

AutoField - Django Models

According to documentation, An AutoField is an IntegerField that automatically increments according to available IDs. One usually won't need to use this directl...

4 min read

BigAutoField - Django Models

BigAutoField is a 64-bit integer, much like an AutoField except that it is guaranteed to fit numbers from 1 to 9223372036854775807. One can create a...

4 min read

BooleanField - Django Models

BooleanField is a true/false field. It is like a bool field in C/C++. The default form widget for this field is CheckboxInput, or NullBooleanSelect if null=True. The...

4 min read

CharField - Django Models

CharField is a string field, for small- to large-sized strings. It is like a string field in C/C++. CharField is generally used for storing small strings like first name, last...

4 min read

DateField - Django Models

DateField is a field that stores date, represented in Python by a datetime.date instance. As the name suggests, this field is used to store an object of date...

5 min read

Article Tags : [Python](#) [Django-forms](#) [Django-models](#) [Python 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

Languages

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

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

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

Python Tutorial

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

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design

DevOps

Git
Linux
AWS
Docker
Kubernetes

Engineering Maths
Software Development
Software Testing

Azure
GCP
DevOps Roadmap

System Design

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

Inteview Preparation

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

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

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

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved