Last Updated : 12 Feb, 2020

A Slug is basically a short label for something, containing only letters, numbers, underscores or hyphens. They're generally used in URLs. For example, in a typical blog entry URL:

```
https://www.geeksforgeeks.org/add-the-slug-field-inside-django-model/
```

Here, the last data `add-the-slug-field-inside-django-model` is the slug.

SlugField :

SlugField in Django is like a CharField, where you can specify `max_length` attribute also. If max_length is not specified, Django will use a default length of 50. It also implies setting **Field.db_index** to **True**.It is often useful to automatically prepopulate a SlugField based on the value of some other value.It uses [validate_slug](#) or [validate_unicode_slug](#) for validation.

Syntax

```
field_name = models.SlugField(max_length=200, **options)
```

SlugField has following optional arguments:

```
SlugField.max_length
```

The maximum length (in characters) of the field. The `max_length` is enforced at the database level and in Django's validation using [MaxLengthValidator](#).

```
SlugField.allow_unicode
```

If True, the field accepts Unicode letters in addition to ASCII letters. Defaults to False.

Django Model SlugField Explanation

Illustration of SlugField using an Example. Consider a project named
`geeksforgeeks` having an app named `geeks`.

*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 ?*

Enter the following code into `models.py` file of **geeks** app.

```python
from django.db import models
from django.db.models import Model
# Create your models here.

class GeeksModel(Model):
    geeks_field = models.SlugField(max_length = 200)
```

Add the geeks app to `INSTALLED_APPS`

```python
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'geeks',
]
```

Now when we run `makemigrations` command from the terminal,

```
Python manage.py makemigrations
```

A new folder named migrations would be created in `geeks` directory with a file
named `0001_initial.py`

```python
# Generated by Django 2.2.5 on 2019-09-25 06:00

from django.db import migrations, models

class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name ='GeeksModel',
            fields =[
                ('id',
                  models.AutoField(
                  auto_created = True,
                  primary_key = True,
                  serialize = False,
                  verbose_name ='ID'
                )),
                ('geeks_field',
                  models.SlugField(
                  max_length = 200,
                )),
            ],
        ),
    ]
```

Now run,

```
Python manage.py migrate
```

Thus, an `geeks_field` **SlugField** is created when you run migrations on the
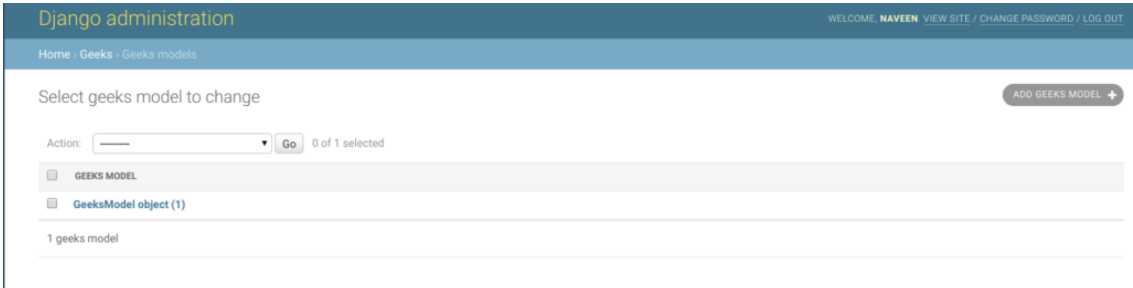project.

How to use SlugField ?

SlugField is used for storing basically storing URL paths after a particular URL.
To know more about how to properly add a SlugField to Django Project, refer
this article – **Add the slug field inside Django Model**

```python
# importing the model
# from geeks app
```

```
from geeks.models import GeeksModel

# creating a instance of
# GeeksModel
geek_object = GeeksModel.objects.create(geeks_field ="")
geek_object.save()
```

Now let's check it in admin server. We have created an instance of GeeksModel.

Field Options

Field Options are the arguments given to each field for applying some constraint or imparting a particular characteristic to a particular Field. For example, adding an argument `null = True` to SlugField will enable it to store empty values for that table in relational database.
Here are the field options and attributes that an SlugField can use.

Field Options	Description
Null	If **True**, Django will store empty values as **NULL** in the database. Default is **False**.
Blank	If **True**, the field is allowed to be blank. Default is **False**.
db_column	The name of the database column to use for this field. If this isn't given, Django will use the field's name.
Default	The default value for the field. This can be a value or a callable object. If callable it will be called every time a new object is created.

```
from geeks.models import GeeksModel
```

Field Options	Description
help_text	Extra "help" text to be displayed with the form widget. It's useful for documentation even if your field isn't used on a form.
primary_key	If True, this field is the primary key for the model.
editable	If **False**, the field will not be displayed in the admin or any other ModelForm. They are also skipped during model validation. Default is **True**.
error_messages	The error_messages argument lets you override the default messages that the field will raise. Pass in a dictionary with keys matching the error messages you want to override.
help_text	Extra "help" text to be displayed with the form widget. It's useful for documentation even if your field isn't used on a form.
verbose_name	A human-readable name for the field. If the verbose name isn't given, Django will automatically create it using the field's attribute name, converting underscores to spaces.
validators	A list of validators to run for this field. See the validators documentation for more information.
Unique	If True, this field must be unique throughout the table.

N   Nave...   G⋮   5

Previous Article

PositiveSmallIntegerField - Django Models

Next Article

SmallIntegerField - Django Models

Similar Reads

SlugField - Django Forms

SlugField in Django Forms is a slug field, for input of slugs for particular URLs or similar. This field is intended for use in representing a model SlugField in forms....

5 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-models   Python Django

Practice Tags :   python

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

Engineering Maths

Software Development

Software Testing

DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

Inteview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

OOAD

System Design Bootcamp

Interview Questions

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