



[Django Basics](#) [Django Projects](#) [Django Interview Question](#) [Flask](#) [Flask Projects](#) [Python API](#) [Torando](#) [Che](#)

Customizing Phone Number Authentication in Django

Last Updated : 24 Sep, 2024

As we know, Introducing phone number-based authentication in the Django Admin Panel is a strategic move to modernize and enhance your web application's security. In the traditional [Django](#) authentication system, users use a unique username and password to access their accounts. However, in today's digital landscape, where mobile phones are integral, implementing phone number authentication streamlines and secures the login process. This change aligns your application with current user preferences and strengthens security by reducing password-related risks. In this guide, we'll explore the key steps to transition from username-based to phone number-based authentication in the Django Admin Panel, providing a more convenient and secure user experience.

Django Admin Panel Username Authentication with Phone Number Authentication

To install Django follow these [steps](#).

Starting the Project Folder

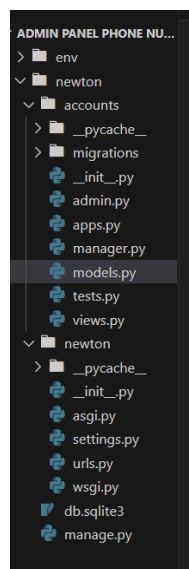
To start the project use this command

```
django-admin startproject newton  
cd newton
```

To start the app use this command

```
python manage.py startapp accounts
```

File Structure :



File Structure

Note: Phone number authentication is a critical feature in modern web applications. To implement this and other advanced features, the [Django Web Development Course](#) will help you build secure and user-friendly systems.

```
INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "accounts", //App name
]
```

Setting up necessary files

models.py: Here we are creating the [models](#) so that we can store information in our database. We define a custom User model that inherits from `AbstractUser`. This allows you to extend the default user model provided by Django.

Python



```
from django.db import models
from django.contrib.auth.models import AbstractUser
```

```

▶ from .manager import UserManager

class GFG(AbstractUser):
    phone = models.CharField(max_length=12, unique=True)
    USERNAME_FIELD = 'phone'
    REQUIRED_FIELDS = []
    objects = UserManager()

```

manager.py: We import BaseUserManager from django.contrib.auth.base_user to create a custom user manager. We define a custom UserManager class that extends BaseUserManager. This manager is responsible for creating and managing user instances. The create_user method is used to create a standard user with a phone number, an optional password, and any extra fields passed as keyword arguments. It checks if a phone number is provided and raises a ValueError if not.

Python

```

1 from django.contrib.auth.base_user import BaseUserManager
2 class UserManager(BaseUserManager):
3     use_in_migrations = True
4
5     def create(self, phone, password=None, **extra_fields):
6         if not phone:
7             raise ValueError('Phone number is required')
8         user = self.model(phone=phone, **extra_fields)
9         user.set_password(password)
10        user.save()
11        return user
12
13    def superuser(self, phone, password, **extra_fields):
14        extra_fields.setdefault('is_staff', True)
15        extra_fields.setdefault('is_superuser', True)
16        extra_fields.setdefault('is_active', True)
17        return self.create(phone, password, **extra_fields)

```

admin.py: Here we are registering the models.

Python

```
1 from django.contrib import admin
2 from .models import GFG
3 admin.site.register(GFG)
```

settings.py: Here we are adding the command to make changes in authentication process of Django.

Python

```
1 AUTH_USER_MODEL = "accounts.GFG" #Appname.model(classname)
```

Deployment of the Project

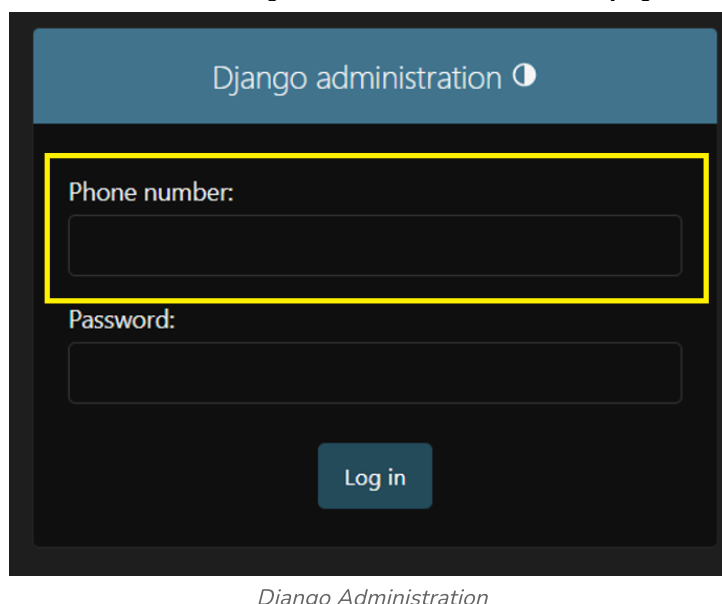
Run these commands to apply the [migrations](#):

```
python3 manage.py makemigrations
python3 manage.py migrate
```

Run the server with the help of following command:

```
python3 manage.py runserver
```

Output



The image shows a screenshot of the Django administration login interface. At the top, there is a blue header bar with the text 'Django administration' and a small circular icon. Below this, the login form is displayed on a dark background. The form contains two input fields: 'Phone number:' and 'Password:'. The 'Phone number:' field is highlighted with a yellow rectangular border. Below the password field is a blue 'Log in' button. The text 'Django Administration' is centered below the form.

Conclusion :

Changing Django Admin Panel Username Authentication to Phone Number Authentication is a significant modification to the default authentication system. This alteration involves customizing Django's authentication backend, forms, and views to accommodate phone numbers as the primary identifier for administrators.

In conclusion, implementing phone number authentication in the Django Admin Panel enhances security and user-friendliness. However, it requires careful consideration of user privacy, validation, and potential challenges related to international phone number formats and verification methods. Ensure you follow best practices and consider regulatory requirements when implementing such changes.

Looking to dive into the world of programming or sharpen your Python skills? Our [Master Python: Complete Beginner to Advanced Course](#) is your ultimate guide to becoming proficient in Python. This course covers everything you need to build a solid foundation from fundamental programming concepts to advanced techniques. With **hands-on projects**, real-world examples, and

expert guidance, you'll gain the confidence to tackle complex **coding challenges**. Whether you're starting from scratch or aiming to enhance your skills, this course is the perfect fit. Enroll now and master Python, the language of the future!



prath...



4

Previous Article

Generate random numbers and background colour in Django

Next Article

Similar Reads

Customizing Object Level Permissions - Django REST Framework

In this article, we will discuss how to customize Object Level Permissions in Django REST Framework. To customize permission classes in Django REST...

5 min read

Python | Extending and customizing django-allauth

Prerequisite: Django-allauth setup and Configuration Let's deal with customizing django-allauth signup forms, and intervening in registration flow to add custom...

4 min read

Customizing Filters in Django REST Framework

Prerequisite: Adding Filtering in APIs – Django REST Framework [link needed article on published yet] Django filters facilitate filtering the queryset to retrieve...

4 min read

Do You Know That Google's Latest Phone is Not a Smart Phone?

Believe it or not, you are probably addicted to your smartphone!!! In fact, there's a high chance that your smartphone is currently in your hand or at least in a 1-...

4 min read

Properly Store and Validate Phone Numbers in Django Models

Choosing the best way to store phone numbers in Django models depends on your application's requirements and complexity. Using Django CharField with...

3 min read

Customizing Styles in Matplotlib

Here, we'll delve into the fundamentals of Matplotlib, exploring its various classes and functionalities to help you unleash the full potential of your data...

12 min read

Customizing Swagger UI

FastAPI is a state-of-the-art, high-performance web framework that uses normal Python type hints to develop APIs with Python 3.7+. FastAPI's ability to...

6 min read

Python Django | Google authentication and Fetching mails from scratch

Google Authentication and Fetching mails from scratch mean without using any module which has already set up this authentication process. We'll be using...

12 min read

Implement Token Authentication using Django REST Framework

Token authentication refers to exchanging username and password for a token that will be used in all subsequent requests so to identify the user on the server...

2 min read

JWT Authentication with Django REST Framework

JSON Web Token is an open standard for securely transferring data within parties using a JSON object. JWT is used for stateless authentication mechanisms for...

2 min read

Article Tags :

[Python](#)

[Django](#)

[Geeks Premier League](#)

[Geeks Premier League 2023](#)

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

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

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

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

Interview 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

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved