Last Updated : 21 Sep, 2021

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 format. There can be multiple ways on how to achieve custom validation. In this article, we are going to show it from the model itself so that you need not manipulate it elsewhere.
These Validation are run when you are trying to create an instance of a model. Technically, this validation is implemented after you run ModelName.objects.create(data = data).

Syntax –

```
field_name = models.Field(validators = [function 1, function 2])
```

Django Custom Field Validation Explanation

Illustration of **validators** 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. We will be using CharField for experimenting for all field options.

Python3

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

class GeeksModel(Model):
    geeks_mail = models.CharField(
                    max_length = 200,
                    )
```

Now we will apply a custom validation so that the above field is validated for google mail IDs only. Create a function that accepts an argument called **value**. One can apply any type of operation on value now. so let us check if our function value contains **@gmail.com** to be validated for google mail IDs only.

Python3

```
from django.db import models
# importing validationerror
from django.core.exceptions import ValidationError

# creating a validator function
def validate_geeks_mail(value):
    if "@gmail.com" in value:
        return value
    else:
        raise ValidationError("This field accepts mail id of google only")


# Create your models here.
class GeeksModel(models.Model):
    geeks_mail = models.CharField(max_length = 200)
```

Now let us add this function as a validator in our field. Note that one validator function can be used for multiple fields simultaneously.

Python3

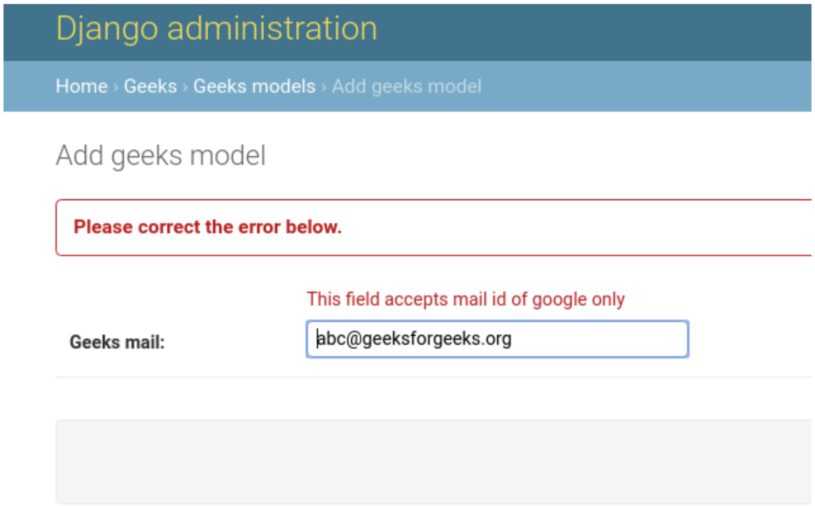
```
from django.db import models
# importing validationerror
from django.core.exceptions import ValidationError
```

```
# creating a validator function
def validate_geeks_mail(value):
    if "@gmail.com" in value:
        return value
    else:
        raise ValidationError("This field accepts mail id of google only")


# Create your models here.
class GeeksModel(models.Model):
    geeks_mail = models.CharField(
                     max_length = 200,
                     validators =[validate_geeks_mail]
                     )
```

Let us try to create an instance without gmail.com and check if our validation worked or not. Note that after every change in models.py one needs to run makemigrations and migrate commands.
In your browser go to http://localhost:8000/admin/geeks/geeksmodel/add/ and enter “abc@geeksforgeeks.org”.

Let us check if it gets saved in database.



Add geeks model

Please correct the error below.

This field accepts mail id of google only

Geeks mail: abc@geeksforgeeks.org

So the validation worked and this field can only accept email ids ending with @gmail.com. This way one can apply any kind of custom validation on value.

N Nave...

Previous Article

Change Object Display Name using __str__ function - Django Models | Python

Next Article

Meta Class in Models - Django

Similar Reads

Built-in Field Validations - Django Models

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

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

Django form field custom widgets

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...

3 min read

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 admi...

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

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

Article Tags : Python Django-models Python Django

Practice Tags : python

GeeksforGeeks
Sanchhaya Education Private Limited

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
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
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