

Let's Understand

Validators in Django Models

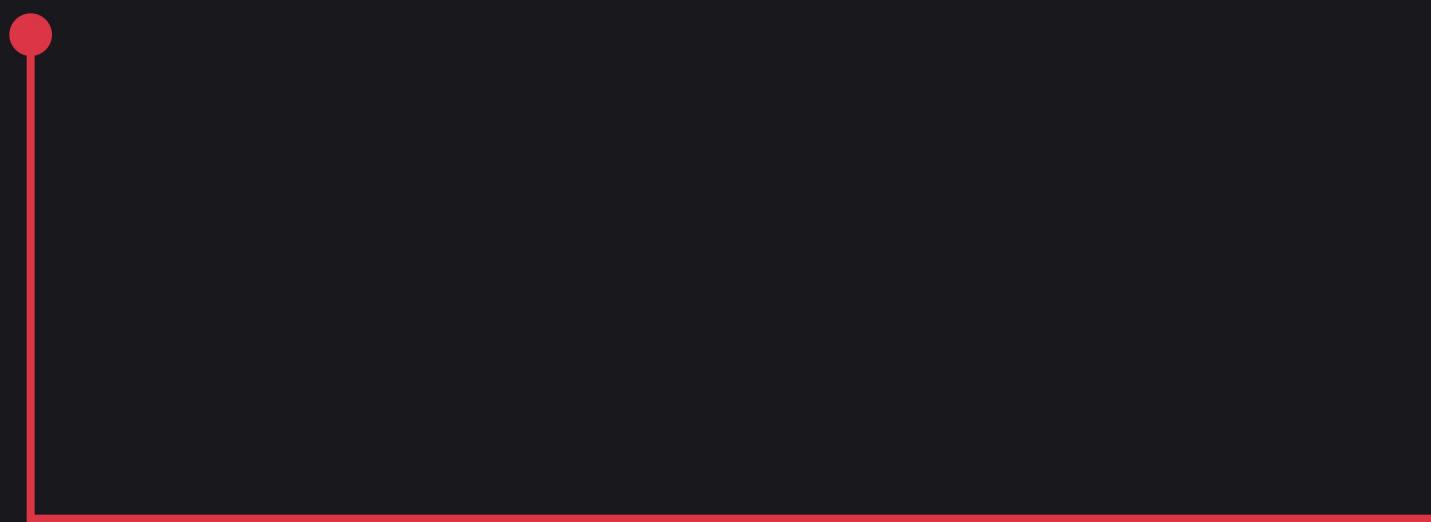


Validators

A validator are the functions that will raises a ValidationError, before it is saved to database, if the field value doesn't meet the criteria.

How to use Validators

To use validators, simply we just need to use **validators** attributes to our models field



Validators Example

Let's say, we have a Student model and it consists roll_number field

```
from django.core.validators import MinValueValidator  
  
class Student(models.Model):  
    roll_number = models.PositiveIntegerField(  
        validators=[MinValueValidator(1)]  
)
```

Built-in validators

So, here we have added this

MinValueValidator build-in validators

and pass the minimum value to add

the validation to our roll_number field.

Built-in Validators in Django Models

Django provides several built-in validators that you can use to validate model fields. These validators are available in the `django.core.validators` module. Some common built-in validators include:

MinValueValidator

It ensures that the value is not less than a specified minimum value.

MaxValueValidator

It ensures that the value is not greater than a specified maximum value

MaxLengthValidator

It ensures length of a string value does not exceed a specified maximum length.

MinLengthValidator

It ensures that the length of a string value is at least a specified minimum length.

EmailValidator

It ensures that the value is a valid email address.

RegexValidator

It ensures that the value matches a specified regular expression pattern.

How to write custom validators

To write a custom validator, simply we just need to create a function, that will take a field value.

```
from django.core.exceptions import ValidationError
from django.utils.translation import gettext_lazy as _

def validate_even(value):
    if value % 2 != 0:
        raise ValidationError(
            _('Value must be an even number'),
            code='invalid'
        )

class MyModel(models.Model):
    even_field = models.IntegerField(validators=[validate_even])
```

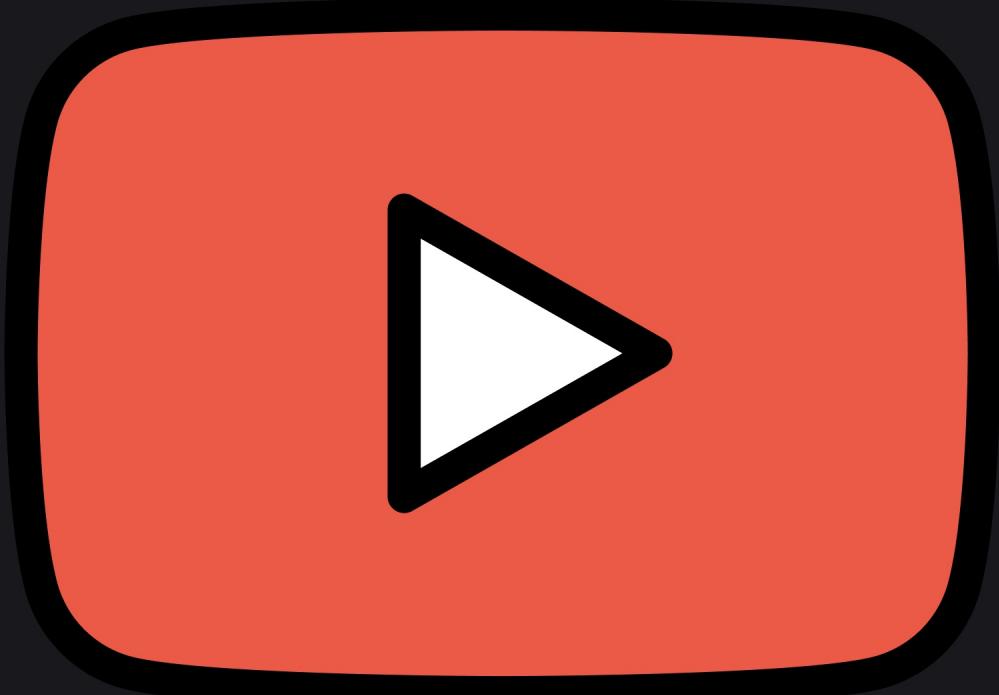
This is a function, that will take a value and raise a validation error, if criteria doesn't match

and here we can simply register the validator

For more django
related content

Subscribe

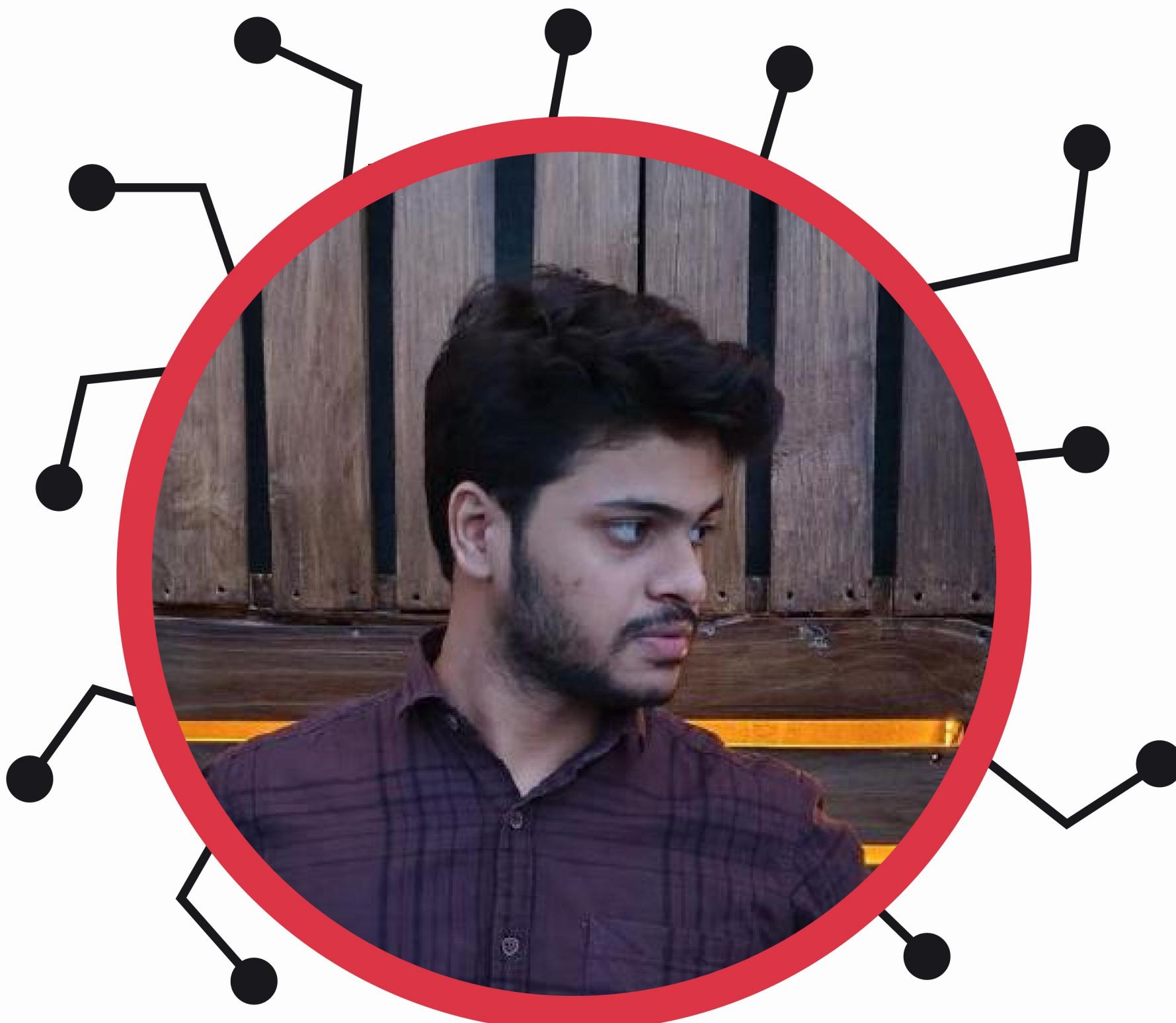
To My Youtube
Channel



Link in bio

DID YOU FIND THIS HELPFUL

Let me know in the comment



Aashish Kumar

Software Engineer

Follow for more ❤