

How to
→

encrypting

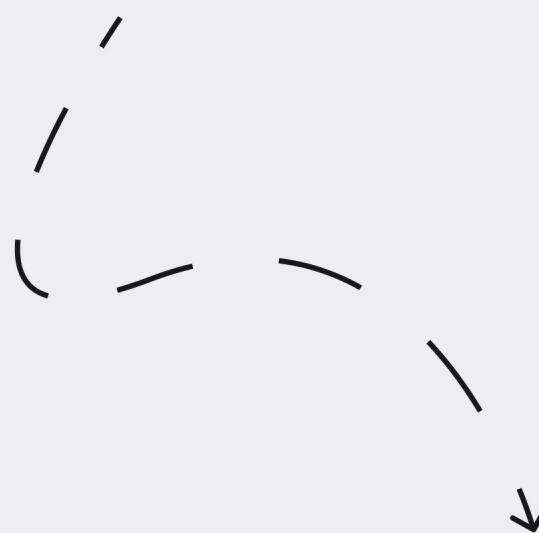
Sensitive Data in Django



Introduction

To encrypting data in django, we will use
cryptography python package.

Simply use this command to install
crpytography package.



```
$ pip install cryptography
```

1. Generate a Strong Secret Key

Next, we need to generate a strong secret key, that we will use for encrypt & decrypt our data in django.

```
from cryptography.fernet import Fernet  
  
key = Fernet.generate_key()  
  
# output  
b'WVqef5yYVcuny4UG0JLlE0VfcQVDkmdconHV5Z3B_H8='
```

Store this secret key in .env file

Define an Encryption Helper

Next, just create helpers.py file inside your app folder, and here will define the two functions that will encrypt and decrypt the data

```
● ● ● helpers.py

from cryptography.fernet import Fernet

# LOAD THE SECRET KEY
SECRET_KEY = """GET THE SECRET KEY FROM .ENV FILE"""

# INSTANCE OF FERNET CIPHER
fernet = Fernet(SECRET_KEY)

# ENCRYPT THE DATA
def encrypt_data(data):
    encrypted_data = fernet.encrypt(data.encode())
    return encrypted_data

# DECRYPT THE DATA
def decrypt_data(encrypted_data):
    decrypted_data = fernet.decrypt(encrypted_data).decode()
    return decrypted_data
```

encrypt the django model data

Let's create a model and encrypt the data
before just save.

```
from django.db import models
from django.contrib.auth.models import User
from .helpers import encrypt_data

models.py

class BankDetails(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    account_number = models.CharField(max_length=100)

    def save(self, *args, **kwargs):
        self.account_number = encrypt_data(data=self.account_number)
        return super().save(*args, **kwargs)
```



Here we just override the save method
and encrypt the account_number data
before it just save

decrypt the django model data

Last slide, we just encrypt the account_number data, now let's decrypt the account_number data in our view function

```
views.py
from django.shortcuts import render
from .helpers import decrypt_data
from .models import BankDetails

def bank_details(request):
    user_bank_details = BankDetails.objects.get(
        user=request.user
    )
    account_number = decrypt_data(
        user_bank_details.account_number
    )

    context = {"account_number": account_number}
    return render(request, "bank-details.html", context)
```

Use this templates file

Let's use the decrypt data in
templates file

```
<!DOCTYPE html>                                         bank-details.py
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Bank Info</title>
  </head>
  <body>
    <div class="container border p-5 text-center mt-5">

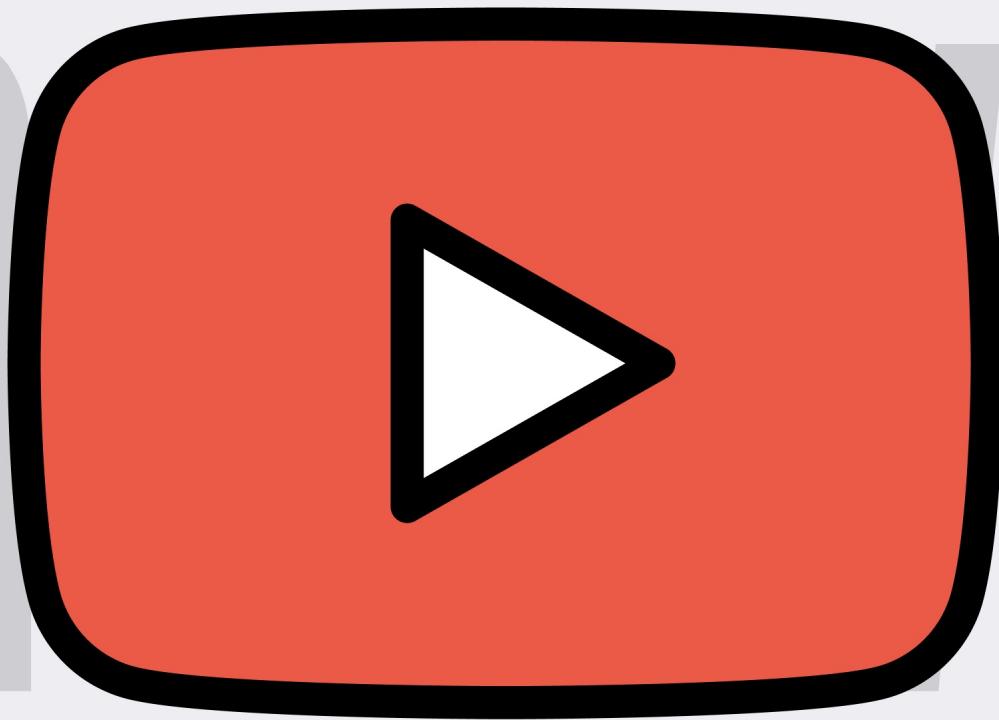
      <h3>Bank Details</h3>

      <p>Account number : {{account_number}}</p>
    </div>
  </body>
</html>
```

In today digital world, the security is very crucial, and this is one of the effective way to enhance the data security through encryption

For more django related content

Subscribe To My Youtube Channel!



Link in bio

DID YOU FIND THIS HELPFUL



Follow for more

#pythonworld