How to Insert Data into a Database from an HTML form in Django

In this article, we show how to insert data into a database from an HTML form in Django.

If you use a ModelForm in Django, you wouldn't need to worry about inserting data into the database, because this would be done automatically. This is the case with Django ModelForms.

However, there are times where you may not want to use Django ModelForms and instead just want to code the form directly in HTML and then insert the data that the user has entered into the database. This is what this article addresses. We will show how to insert the data that a user enters into an HTML form into the database.

So let's create a form in which a user creates a post. This post form will simply take in 2 values, the title of the post and the content of the post.

We will then insert this into the database.

So the first thing we have to do is create our database (the model) in the models.py file.

models.py File

So the first thing we have to do is create our database.

We will call our database, Post.

It will only have two fields: title and content.

```
#models.py File
```

```
#models.py File

from django.db import models


class Post(models.Model):
    title= models.CharField(max_length=300, unique=True)
    content= models.TextField()
```

Okay, this is a very basic database. We simply have a title and a content field.

After this, we save the file and then, within the command line, run the command, py manage.py makemigrations, and then run the command, py manage.py migrate.

createpost.html Template File

Now we'll create our template file. We'll call it, createpost.html

Within this template file, we are going to have the form where a user can submit a post.

It is a simple form that only contains 2 fields: title and content.

This is shown in the code below.

```
<html>
<head>
<title>Create a Post </title>
</head>

<body>
<h1>Create a Post </h1>
<form action="" method="POST">
{% csrf_token %}
Title: <input type="text" name="title"/><br/>
Content: <br/>
<textarea cols="35" rows="8" name="content">
</textarea><br/>
<input type="submit" value="Post"/>
</form>
</body>

</html>
```

So this template contains a very basic form that has 2 fields: one which is title and the other which is content.

We need a name attribute with each form field because this is how we will extract the data that the user enters into the field.

views.py File

Lastly, we have our views.py file.

In this file, we will take the data that the user has entered into the form fields and insert the data into a database.

The following code in the views.py file does this.

```
from django.shortcuts import render
from .models import Post


def createpost(request):
        if request.method == 'POST':
            if request.POST.get('title') and request.POST.get('content'):
                post=Post()
                post.title= request.POST.get('title')
                post.content= request.POST.get('content')
                post.save()

                return render(request, 'posts/create.html')

        else:
                return render(request,'posts/create.html')
```

So this is the heart of our code in which we extract the data from the form fields that the user has entered and insert the data into a database.

The first thing we want to do is to make sure that the user has clicked the 'Post' button on the template file. We check this with, if request.method == 'POST':

We want to make sure that the fields aren't blank. So we use the if statement, if request.POST.get('title') and request.POST.get('content'):, to make sure both fields are filled in.

After this, we create a variable named post and set it equal to Post()

This sets the variable equal to the Post database. Remember that you must import the database at the top of the views.py page.

Now in order to insert data into the database use the database name followed with a ., and then the form field- we then set this equal to request.POST.get('attribute')

To save data to the title database field, we use the statement, post.title, and set this equal to request.POST.get('title'). This takes the title database field and sets it equal to whatever data the user entered into the title form field.

To save data to the content database field, we use the statement, post.content, and set this equal to request.POST.get('content'). THis takes the content database field and sets it equal to whatever data the user entered into the content form field.

We then save the data. Without the statement, post.save(), the data isn't saved.

You then would return any template file that you want to.

And this is how we can insert data into a database from an HTML form in Django.

Related Resources

[How to Randomly Select From or Shuffle a List in Python]

Comments

Name

Enter your comment here

Comment Add Image

Ritesh Gupta · May 14, 2021

Sir data saved but in which database and table how to see about this

Like · Flag

Soham Desai · Apr 29, 2021

thanks a lot mate

Like · Flag

Unnati · Apr 2, 2021

Thanks a lot

Like · Flag

Priyanka gite · Feb 3, 2021 1 👍

Thank you

Like · Flag

Karol · Sept 17, 2020

Thanks

Like · Flag

raghu · Aug 21, 2020

Very good. saved my lot of time..

Like · Flag

karan · Aug 14, 2020 5 👍

can we make big web app using django

Like · Flag

ankit · July 13, 2020

its helpfull

Like · Flag

banga · July 10, 2020 1 👍

understood

Like · Flag

Anonymous · June 22, 2020 2 👍

same error sanja

Like · Flag

Showing 1 to 10 →