Clase 21 - Playground Intermedio (parte 3)

Formularios

Hasta el momento sabemos insertar datos por medio de la terminal (ejemplo: curso.save()) y por medio del panel de administración (ejemplo: por medio del add).

Básicamente, con esas dos alternativas estamos haciendo lo que se llama un “insert” en nuestra base de datos. Pero ninguna de estas dos opciones suelen ser las más utilizadas.

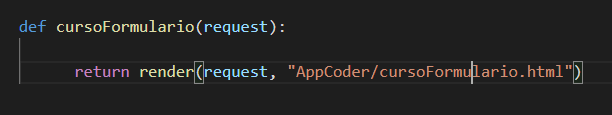
Lo más utilizado suele ser los formularios, que no es otra cosa más que una vista que permite poder agregar datos a nuestro model directamente desde nuestra web.

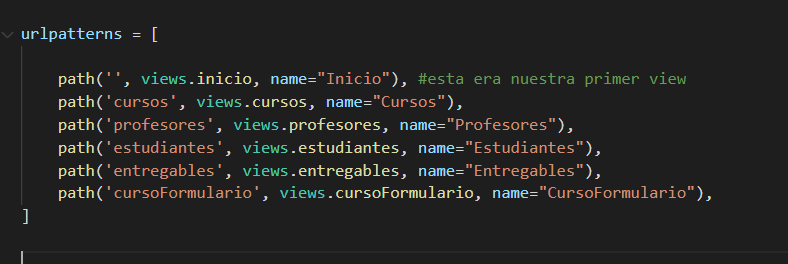
Los formularios se pueden hacer por medio de HTML con CSS (sabiendo algo de desarrollo web), pero también podemos hacer uso de facilidades que nos da Django para lograrlo.

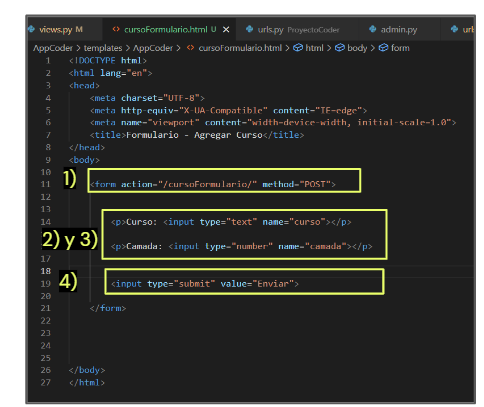
¿Cómo funcionan los formularios?

El html recibe nuestra información por medio de la vista y su template asociado. Al apretar un botón esa información viaja por medio de un método GET o POST y llega al servidor, donde esos datos se manipulan.

Creación de formularios (HTML)



1. Vamos a las vistas, views.py y creamos una nueva.
2. Luego agregamos la vista a las urls.py
3. Creamos el html. (donde haremos el formulario)



1) Action es el nombre de la url

2) Method es la forma en la que se envían los datos

3) Input son espacios para escribir información

4) Input Submit es el botón que envía la información.

Script:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Formulario - Agregar Curso</title>

</head>

<body>

    <form action="/cursoFormulario/"    method="post">

        <p> Curso: <input type = "text" name="curso">

        <p> Camada: <input type = "text" name="camada">

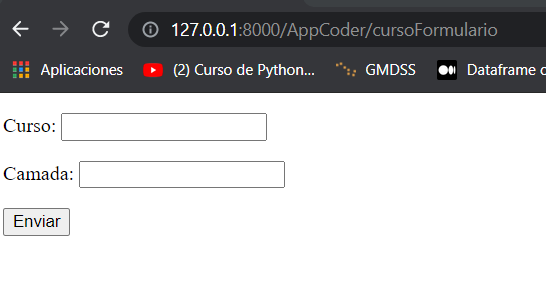
        <input type = "submit" value="Enviar">

     </form>

</body>

Creación de formularios

¡Nuestra vista!



Obviamente, al apretar el botón Enviar, aún no hace nada. Necesitamos que esos datos viajen al servidor y podamos hacer algo con ellos.

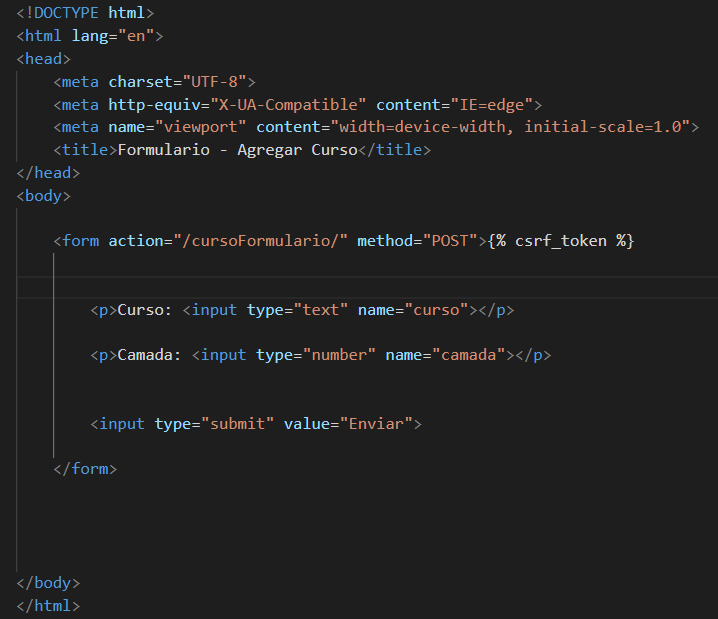
Para esto:

1- Debemos agregar el token de validación que nos exige Django

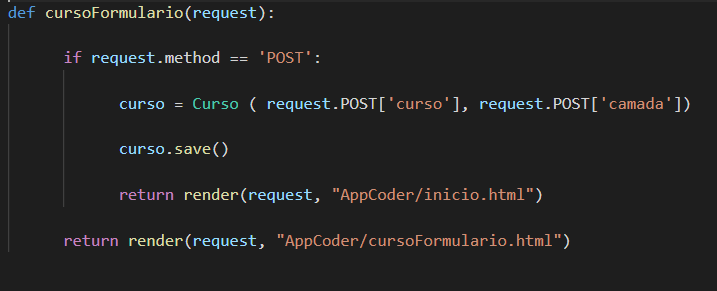
2- En la vista que creamos recibimos los datos.

De esta forma podemos guardar en la base de datos los datos recibidos por medio del form.

{% csrf\_token %}



Pero nos podría generar problemas de validación de los datos enviados, así que lo haremos con los formularios de Django.



Script:

def cursoFormulario(request):

      if request.method == 'POST':

            curso =  Curso(request.post['curso'],(request.post['camada']))

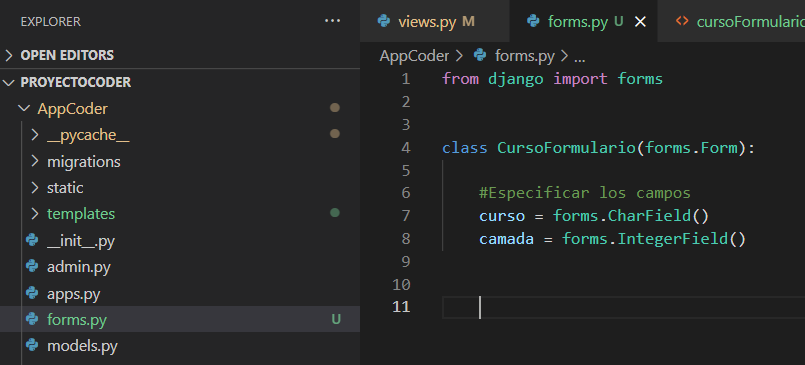
            curso.save()

            return render(request, "AppCoder/inicio.html")

      return render(request,"AppCoder/cursoFormulario.html")

Creación de formularios

(Api from Django)



1. Crear un archivo forms.py (en la APP).
2. Importamos los forms.
3. Creamos una clase para crear un formulario.

Script:

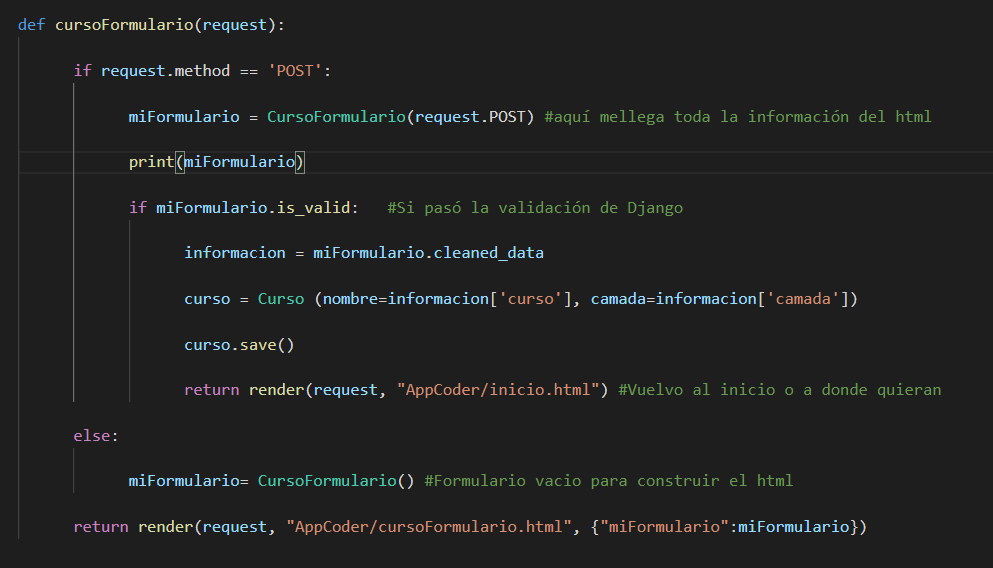
from django import forms

class CursoFormulario(forms.Form):

    curso = forms.CharField()

    camada = forms.IntegerField()

1. Recibimos en el HTML, el formulario para crear el <form>



Script:

from AppCoder.forms import CursoFormulario

def cursoFormulario(request):

      if request.method == "POST":

            miFormulario = CursoFormulario(request.POST) # Aqui me llega la informacion del html

            print(miFormulario)

            if miFormulario.is\_valid:

                  informacion = miFormulario.cleaned\_data

                  curso = Curso(nombre=informacion["curso"], camada=informacion["camada"])

                  curso.save()

                  return render(request, "AppCoder/inicio.html")

      else:

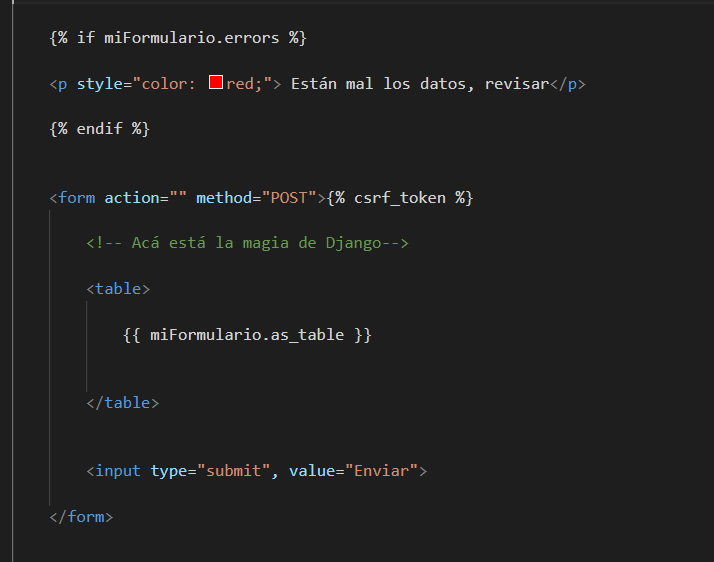
            miFormulario = CursoFormulario()

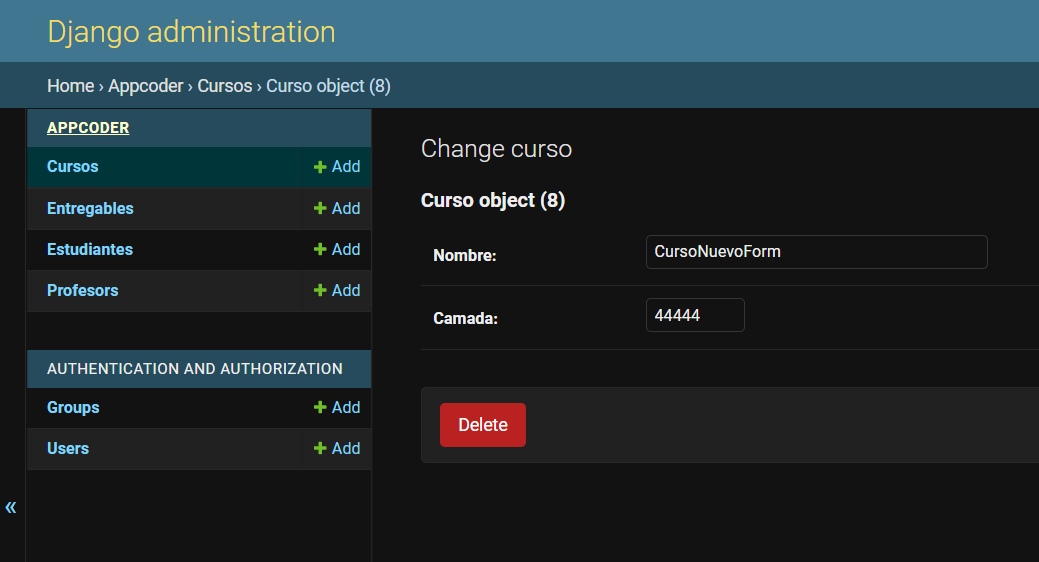
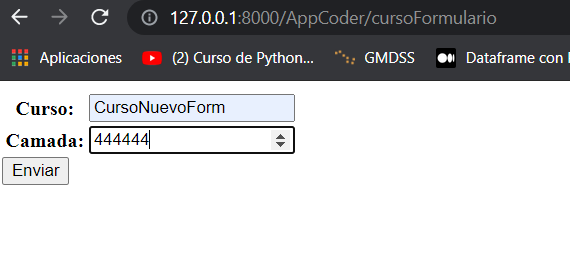
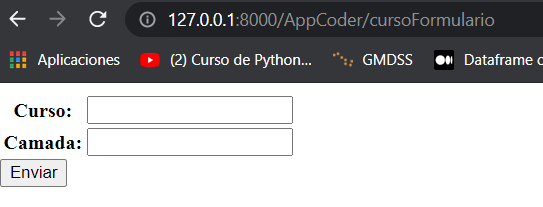
      return render(request, "AppCoder/cursoFormulario.html", {"miFormulario": miFormulario})

1. Modificamos la vista para que ya no reciba el HTML, sino el api forms.

**from AppCoder.forms import CursoFormulario.**

Vemos que queda muy sencillo y cada vez nos desligamos más de html.

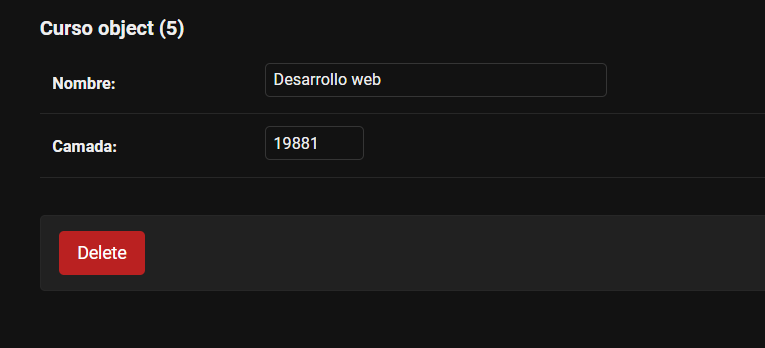




Búsqueda con Form

Otra cosa que suele ser útil es usar un formulario para buscar algún dato en nuestra base de datos.

Por ejemplo: si queremos saber si tenemos un curso que corresponda a una determinada camada, hay dos alternativas, o existe o no existe. En nuestra caso existe en la BD.

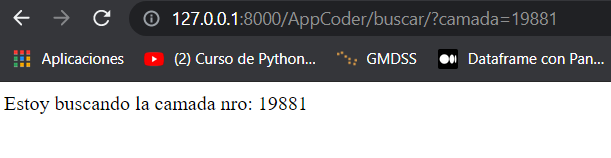
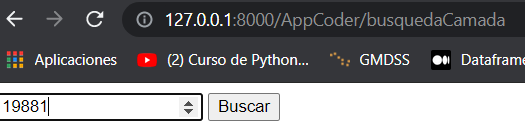


¡Ahora busquemos esa camada!

1. Creamos una vista, busquedaCamada(request)
2. Registramos el url
3. Creamos busquedaCamada.html
4. Creamos la vista buscar.



¡Anunciamos que lo hemos logrado!

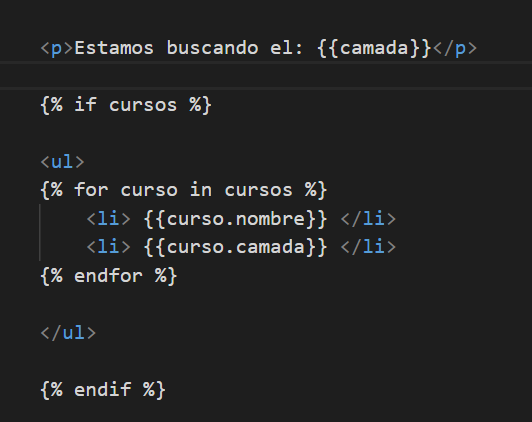


Los datos llegaron, pero aún no buscamos en la BD, pasemos a eso ahora, principalmente usaremos una búsqueda por filtros en la base de datos con objects.filter:



Mientras que nuestro **resultadosPorBusqueda.html** quedaría así.

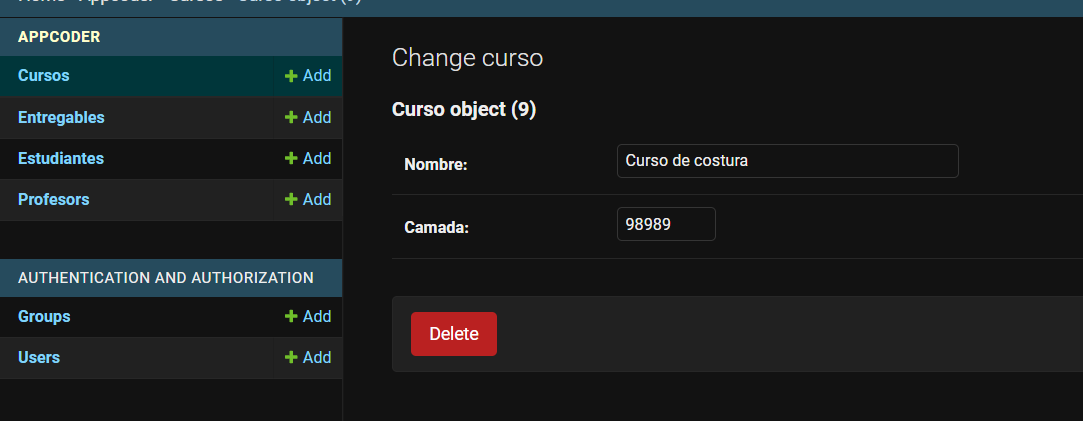
Notar que en nuestro caso teníamos más de un curso con esa camada, filter trae todos los registros, no solo uno.



Ordenamiento de template

Ordenamiento de template

Con los formularios aprendimos a insertar nuevos datos y buscar datos en la base de datos. Próximamente, aprenderemos a modificar datos y eliminar datos. Pero, hicimos muchos html y muy poco estéticos, ahora ordenaremos todo para tener forms dentro html heredados de la planilla padre.



¿Cómo lo hice?

¡Simple! Cambiando el url y la vista.

La vista que se llamaba cursoFormulario, pasó a llamarse cursos y me lleva a la página de cursos.html con un formulario.

