

Django Inicial

Clase 35

django

CONQUERBLOCKS

i18n

i18n

La internacionalización (i18n) en Django es el proceso de preparar tu aplicación para admitir varios idiomas, y es un tema importante para aplicaciones que requieren ser multilingües. Django ofrece un sólido soporte para la traducción de textos y plantillas, lo que permite traducir no solo mensajes estáticos en plantillas, sino también cadenas de texto dentro de vistas, formularios, y más.

Diferentes cosas a internacionalizar:

Plantillas, modelos y formularios

Instancias de nuestros modelos

i18n

```
# Habilitar internacionalización
USE_I18N = True

# Configurar la localización
USE_L10N = True

# Configurar el soporte para zonas horarias
USE_TZ = True

# Definir el idioma predeterminado
LANGUAGE_CODE = 'en-us' # Cambia a tu idioma predeterminado, como 'es' para español

# Definir las opciones de idiomas disponibles
LANGUAGES = [
    ('en', 'English'),
    ('es', 'Español'),
    # Puedes añadir más idiomas según sea necesario
]

# Definir la ruta donde se almacenarán los archivos de traducción
LOCALE_PATHS = [
    BASE_DIR / 'locale', # 'locale' será la carpeta donde se guardarán las traducciones
]
```

Textos en plantillas

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>{% trans "Mi sitio web" %}</title>
</head>
<body>
  <h1>{% trans "Bienvenido a nuestro sitio web" %}</h1>
</body>
</html>
```

Con variables

Usar variables dentro de {% blocktrans %}

Si tienes una variable dinámica dentro del texto, puedes utilizar {% blocktrans %}, que permite traducir bloques de texto que contienen variables.

html

 Copiar

```
<p>{% blocktrans %}Hola {{ username }}, bienvenido de nuevo!{% endblocktrans %}</p>
```

En este caso, {{ username }} es una variable que se mantendrá sin traducir, mientras que el resto del texto se traducirá según el idioma del usuario.

Traducción de textos en vistas y modelos

En las vistas, modelos, formularios y cualquier otro lugar en tu código Python, puedes traducir cadenas utilizando las funciones `gettext` y `gettext_lazy` que Django proporciona.

Traducción de textos en vistas y modelos

En las vistas, modelos, formularios y cualquier otro lugar en tu código Python, puedes traducir cadenas utilizando las funciones `gettext` y `gettext_lazy` que Django proporciona.

Importar las funciones de traducción:

python

```
from django.utils.translation import gettext as _  
from django.utils.translation import gettext_lazy as _lazy
```

Traducción de textos en vistas y modelos

Ejemplos de uso en vistas y modelos:

python

 Copiar

```
from django.utils.translation import gettext as _

# Uso en una vista
def my_view(request):
    mensaje = _("Hola, este es un mensaje traducido")
    return render(request, 'my_template.html', {'mensaje': mensaje})

# Uso en un modelo
from django.db import models
from django.utils.translation import gettext_lazy as _lazy

class Producto(models.Model):
    nombre = models.CharField(max_length=100)
    descripcion = models.TextField()
    precio = models.DecimalField(max_digits=10, decimal_places=2)

    class Meta:
        verbose_name = _lazy("Producto")
        verbose_name_plural = _lazy("Productos")
```

Una vez marcadas nuestras cadenas

Necesitamos generar los archivos de traducción
django-admin makemessages -l es

¿qué hace esto?

i18n

Tarea

◀ Despedida ▶

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez