

Python web con Django

Módulo 1

Otros tipos de respuesta

Otros tipos de respuesta

Generalmente, el contenido de la respuesta que se retorna desde una vista está en formato HTML, porque se está desarrollando un sitio web. Con lo visto hasta ahora, se podría, simplemente, crear el código HTML dentro de una vista y pasarlo como argumento a **HttpResponse**.

```
def index ( request ):  
    return HttpResponse( "<html>j<strong>Hola</strong>, <em>mundo</em>!</html>"  
    )
```

Para dar un paso más, se puede implementar una vista que arme una tabla HTML con nombres de cursos y la cantidad de inscriptos en cada uno.

Primero, se crea una base de datos de SQLite llamada **myproject/cursos.db** (junto al archivo **manage.py**) con una tabla de cursos con algunos datos como los siguientes:

Table:	cursos	
	nombre	inscriptos
1	Python	10
2	Java	12
3	PHP	8



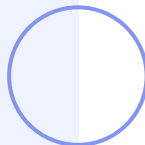
Luego la vista **cursos()** en **myapp/views.py** , que hace una consulta a esa tabla y arma un código HTML:

```
def cursos ( request ):  
    conn = sqlite3.connect( "cursos.db" )  
    cursor = conn.cursor()  
    cursor.execute( "SELECT nombre, inscriptos FROM cursos" )  
    html = ""  
    <html>  
    <title>Lista de cursos</title>  
    <table style="border: 1px solid">  
    <thead>  
    <tr>  
    <th>Curso</th>  
    <th>Inscriptos</th>  
    </tr>  
    </thead>
```

...

```
"""
for (nombre, inscriptos) in cursor.fetchall():
    html += f """
    <tr>
    <td> { nombre } </td>
    <td> { inscriptos } </td>
    </tr>
    """

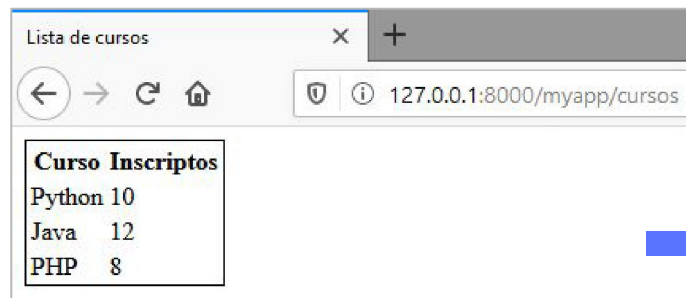
html += "</table></html>"
conn.close()
return HttpResponse(html)
```



Finalmente, se asocia esta vista con una dirección de URL en **myapp/urls.py**:

```
urlpatterns = [  
    path( "", views.index, name = "index" ),  
    path( "acerca-de", views.acerca_de, name = "acerca_de" ),  
    path( "cursos", views.cursos, name = "cursos" ),  
]
```

Al visitar <http://127.0.0.1:8000/myapp/cursos>, se observa:



También es posible retornar respuestas con otros formatos, como JSON o XML si se está diseñando un servicio web. He aquí, por ejemplo, una vista que retorna esta misma tabla en formato JSON:

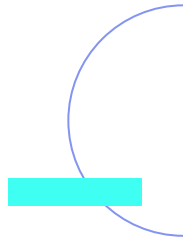
```
from django.http import HttpResponse, JsonResponse
# [...]
def cursos_json ( request ):
    conn = sqlite3.connect( "cursos.db" )
    cursor = conn.cursor()
    cursor.execute( "SELECT nombre, inscriptos FROM cursos" )
    response = JsonResponse(cursor.fetchall(), safe = False )
    conn.close()
    return response
```

Nótese que ahora la clase es **JsonResponse**, no **HttpResponse**.

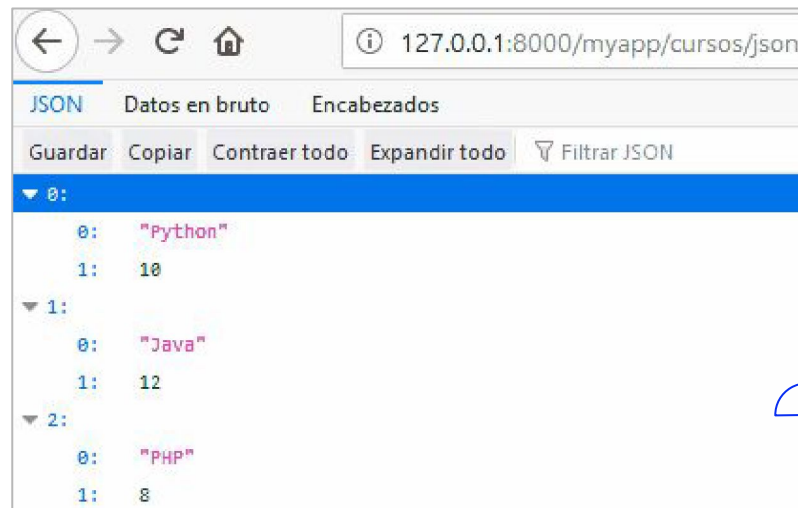


Luego se agrega la dirección de URL:

```
urlpatterns = [  
    path( "", views.index, name = "index" ),  
    path( "acerca-de" , views.acerca_de, name = "acerca_de" ),  
    path( "cursos" , views.cursos, name = "cursos" ),  
    path( "cursos/json" , views.cursos_json, name =  
        "cursos_json" )  
]
```

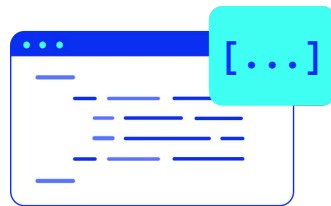


Si se ingresa a <http://127.0.0.1:8000/myapp/cursos/json>,
con Firefox se verá algo así:



O bien en la consola interactiva vía Requests:

```
>>> import requests
>>> r =
requests.get("http://127.0.0.1:8000/myapp/cursos/json")
>>> r.json()
[['Python', 10], ['Java', 12], ['PHP', 8]]
```



**¡Sigamos
trabajando!**