ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

PARSING HTTP REQUEST

LENGUAJES DE PROGRAMACIÓN

Tarik Saghouani Ben-Khalek

Pablo Sánchez Gómez

Eva María Hoyo de la Cruz

José Ignacio Manso LLanes

Contenido

Introducción

En esta práctica realizaremos las modificaciones necesarias en el servidor de la práctica 4 para que sea capaz de procesar una petición POST. Las peticiones POST vienen, por ejemplo, cuando alguien aprieta un botón submit en un formulario en un browser. El ejercicio 4 acabó con un formulario presentado en la pantalla del browser, en esta práctica el servidor deberá recoger los datos que se le envían a través de la petición POST generada al pulsar el botón submit en el formulario. Una vez recibidos los datos e introducidos en el diccionario correspondiente según la práctica 2, el servidor imprimirá por la salida estándar una línea con cada uno de los parámetros recibidos.

Entrega

Tal y como indica el enunciado de la práctica esta entrega consta de dos partes. En la primera, en un archivo adjunto a este documento se encuentra el código del servidor, que además consta en el ANEXO I más adelante.

Adicionalmente, y para una mejor comprensión del código desarrollado por este grupo, una explicación de este.

Entregas previas

Esta práctica es la suma de las practicas precedentes más un parte extra, por lo que daremos por explicado el código correspondiente a las prácticas anteriores y explicaremos las instrucciones correspondientes a la nueva funcionalidad.

Función adicional

Tal y como se ha expuesto en la introducción, la nueva funcionalidad consiste en imprimir por pantalla los datos obtenidos en la práctica anterior. Haciendo un refresco, estos datos provenían de rellenar el formulario web correspondiente a la dirección

http://127.0.0.1:5005/form.html

De nuevo los cambios se producen en dentro del bucle ***while*** que realiza la espera activa para que el proceso cliente realice su petición al servidor.

Por lo que se han insertado las siguientes instrucciones y realizado las siguientes modificaciones a nuestro código:

```
servidor.listen(2)
```

Ahora el servidor, no solo recibe una petición si no que está preparado para recibir como máximo 2.

```
ruta = '../prac5/html'
filename = ruta + respArray[1]
ruta2 = ruta + '/form.html'
tam = os.path.getsize(ruta2)
```

Se propone una mejora para el path, con lo que se debería optimar el acceso y la modificación de la ruta al archivo ***html***.

Se añade al código las siguientes instrucciones

```
textopost=cliente.recv(1024)
textopost=textopost.decode()
#print(textopost)
subtextopost = textopost.split('\n')
stp1=subtextopost[1].split(':')
stp2=subtextopost[2].split(':')
stp3=subtextopost[3].split(':')
stp4=subtextopost[4].split(':')
stp5=subtextopost[5].split(':')
stp6=subtextopost[6].split(':')


diccionariopost = {
    'User-Agent': stp1[1],
    'Accept': stp2[1],
    'Accept-Language': stp3[1],
    'Accept-Encoding': stp4[1],
    'Connection': stp5[1],
    'Upgrade-Insecure-Requests': stp6[1],

}
```

Estas instrucciones son las encargadas de añadir los datos introducidos en cada campo al diccionario. Además de recoger insertados en estos campos, en las variables ***stpX*** se recoge la información relativa a browser o navegador web a través del cual se ha conectado al servidor.

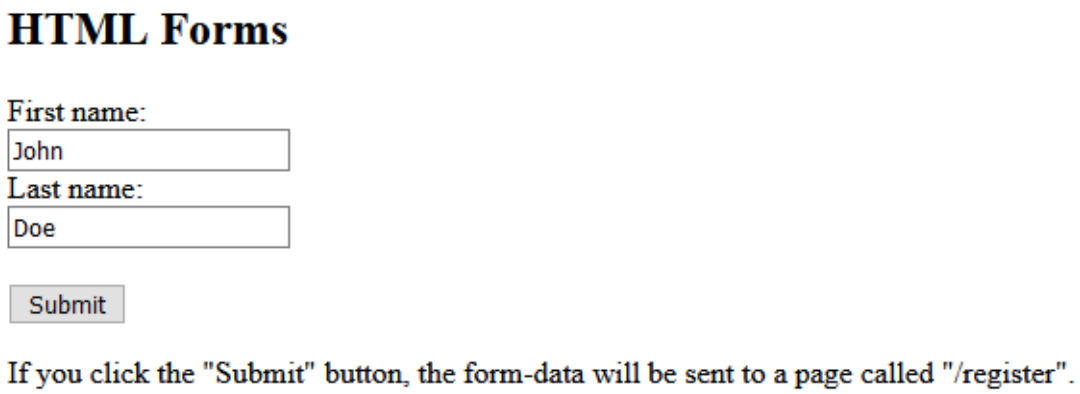
Luego se crea un diccionario en el que se asigna un campo texto a uno de estos valores.

Por último, se recorre la colección de datos (diccionario) imprimiendo su campo clave y los valores asociados a este por la pantalla del entorno de desarrollo. Obteniendo como resultado el solicitado en el enunciado de la práctica.

Además se ha añadido un icono con formato favicon.ico para que se quite un error que saldría del formulario si no se añade, también para quitarlo se podrian quitar unas íneas de código del html pero se ha optado por añadir el icono, dentro de la carpeta html del proyecto.

Capturas de las salidas

Salida correspondiente al navegador que hemos usado para probar la práctica, en este caso Mozilla Firedox.



Salida correspondiente con al entorno de navegación, en nuestro caso PyCharm

ANEXO I

```
import os
import socket

servidor = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
ip = 'localhost'
port = 5005
print("starting up on %s port %d" % (ip, port))
servidor.bind((ip, port))
servidor.listen(2)

ap = [" ", " "]
ap1 = [" ", " "]
while True:
    print("Waiting for a connection.")
    cliente, direccion = servidor.accept()
    print("Conection from: %s %d" % (direccion[0], direccion[1]))
    resp = cliente.recv(44)
    respuesta = resp.decode()
    print("Received: %s" % respuesta)
    respArray = respuesta.split()
    conex = (respArray[2], "200", "OK")
    respArray[0] = ' '.join(conex)
    ruta = '../prac5/html'
    filename = ruta + respArray[1]
    ruta2 = ruta + '/form.html'
    tam = os.path.getsize(ruta2)
    file = open(ruta2, 'rb')
    text = file.read(tam)

    if "fname=" in respuesta:
        nombre = respuesta.split('fname=')
        ap = nombre[1].split('&lname=')
        ap1 = ap[1].split(' HTTP')

    contenido = {
        'Cache-Control': 'no-store',
        'Content-Length': tam,
        'Content-Type': 'text/html; charset=utf8',
        'Connection': 'close',
        'Contenido del fichero': ap[0] + " " + ap1[0],
    }
    respCont = ''.join('%s: %s\n' % (k, v) for k, v in contenido.items())
    print(" ")

    print(respCont)
    print(text)
    file.close()
    cliente.send(respArray[0].encode())
    cliente.send(respCont.encode())
    cliente.send('\n'.encode())
    cliente.send(text)



    textopost=cliente.recv(1024)
    textopost=textopost.decode()
```

```
    #print(textopost)
    subtextopost = textopost.split('\n')
    stp1=subtextopost[1].split(':')
    stp2=subtextopost[2].split(':')
    stp3=subtextopost[3].split(':')
    stp4=subtextopost[4].split(':')
    stp5=subtextopost[5].split(':')
    stp6=subtextopost[6].split(':')


    diccionariopost = {
        'User-Agent': stp1[1],
        'Accept': stp2[1],
        'Accept-Language': stp3[1],
        'Accept-Encoding': stp4[1],
        'Connection': stp5[1],
        'Upgrade-Insecure-Requests': stp6[1],

    }

    print("Inicio datos del formulario ")
    print(diccionariopost)
    print(ap[0] + " " + ap1[0])
    print("Fin datos del formulario ")
    cliente.close()
```