

Arquitecturas Empresariales

Cientes y servicios

Juan Camilo Posso Guevara

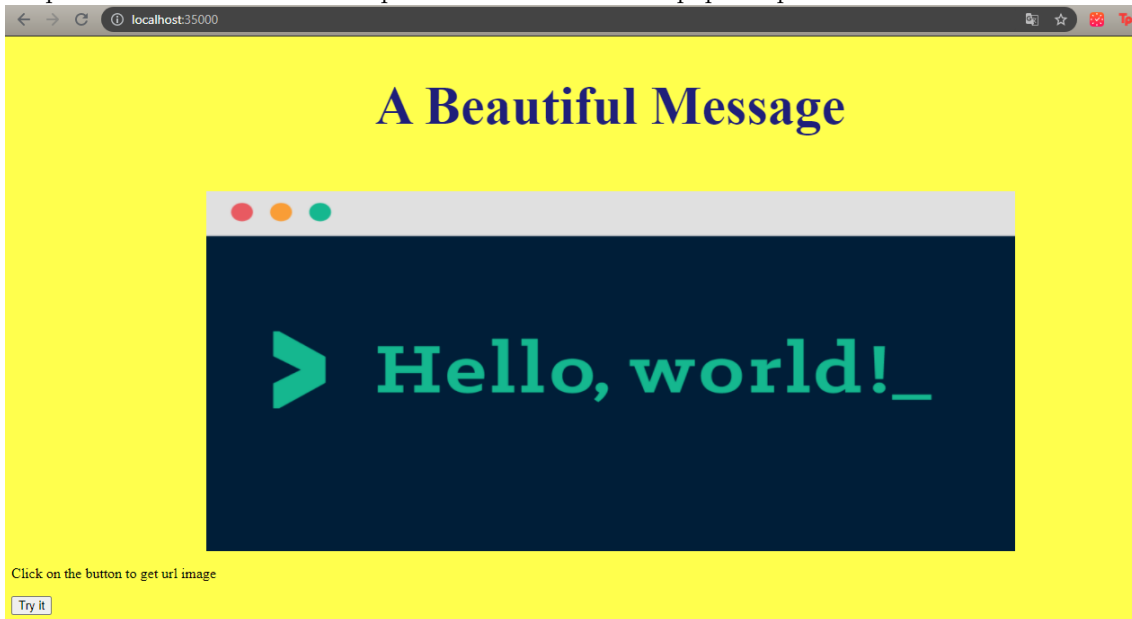
September 8, 2021

1 Introduction

Para este ejercicio se plantea construir un reto, el cual consisten en escribir un servidor web el cual soporte multiples solicitudes sseguidas (no concurrentes) El servidor debe retornar todos los archivos solicitados, incluyendo páginas html e imágenes. Construya un sitio web con javascript para probar su servidor. Despliegue su solución en Heroku. NO use frameworks web como Spark o Spring use solo Java y las librerías para manejo de la red. Para ello hablaremos del diseño del reto planteado y de su funcionamiento.

2 Diseño

Se planea construir una clase que reciba solicitudes http para que atienda solicitudes del cliente.



2.1 HttpServer

Su función será actuar como un servidor el cual cuenta con los siguientes métodos:

2.1.1 Método: start

Su función principal será la de configurar el funcionamiento principal y manejo de la red al igual del uso de los distintos recursos java para poder mantener conexión con el cliente y por otro lado escuchar solicitudes de manera constante una vez recibidas las solicitudes llamaremos al siguiente método.

2.1.2 Método: serverConnection

Su funcion será la de procesar las distintas solicitudes hechas por el cliente para poder ordenarla y enviarla a otro metodo para que genere una respuesta adecuada al cliente.

2.1.3 Método: getResource

Su funcion principal es recibir la solicitud hecha previamente procesada por serverConnection , para ello buscará si lo que se necesita está en la ruta principal (index.html) de no ser asi llamará a una funcion secundaria generatePath la cual nos buscará la ruta del archivo y acto seguido genera el header de la respuesta con la funcion generateHeader para conocer el tipo de archivo que se estará tratando , una vez hecho esto se genera la respuesta final de la solicitud del cliente.

3 Evaluacion

Para probar la conexion se busco primero que hiciera lectura por el puerto designado en localhost del archivo principal index.html y que retornara su contenido luego se probó con cada uno de los tipos de archivos javascript, css y las imagenes en la ruta del navegador para lo cual generó correctamente el contenido de los archivos creados en html-public. Finalmente se almacenó en heroku y se comprobó que generara los recursos estaticos para lo cual se generaron correctamente.

4 Conclusiones

A traves de este ejercicio se pudo conocer como generar solicitudes a un servidor a traves de un cliente y retornar respuestas de manera correctamente segun el tipo de archivo o documento solicitado.