WEBSOCKET DESIGN ASSIGNMENT

(Web Socket server implementation reads a CSV file and sends its contents as JSON to connected clients)

Aniverthy Amrutesh
Computer Science and Engineering
Dr. Ambedkar Institute of Technology
Banglore-56

WebSocket Server Documentation

This WebSocket server implementation reads a CSV file and sends its contents as JSON to connected clients every second.

Function: convert_csv_to_json(file_path)

- Input: file_path (string) - The path to the CSV file.
- Output: JSON representation of the CSV data.

This function reads the CSV file at the specified **file_path** and converts each row into a JSON object. It returns the JSON representation of the CSV data.

WebSocket Server Handler: websocket_server(websocket, path)

- Input: **'websocket'** (WebSocket object) - The WebSocket connection object.
- Input: **'path'** (string) - The URL path of the WebSocket connection.

This function is the handler for the WebSocket server. It prompts the user to enter the CSV file path. If the file path is valid and the file is in CSV format, the server continuously sends the CSV data as JSON to the client every second using the WebSocket connection.

Main Execution:

1. Sets up the WebSocket server to listen on an available port on the local machine.
2. Starts the WebSocket server and prints the server's address where it is listening for connections.
3. Runs the server indefinitely.

WebSocket Client Documentation

This WebSocket client implementation connects to the WebSocket server and receives the JSON data sent by the server.

WebSocket Client Handler: websocket_client()

This function is the handler for the WebSocket client. It establishes a WebSocket connection to the server and continuously receives JSON data from the server. The received JSON data is printed to the console.

Main Execution:

1. Establishes a WebSocket connection to the server.
2. Receives JSON data from the server and prints it to the console.
3. Continues listening for data until termination or disconnection.

Output:

- WebSocket Server:
  - Prints the address where the WebSocket server is listening, e.g., "WebSocket server started. Listening on 0.0.0.0:59050."
  - Prompts the user to enter the CSV file path.
- WebSocket Client:
  - Prints the received JSON data from the server to the console, e.g., [{"11": "66", "22": "77", "33": "88", "44": "99", "55": "1010"}, {"11": "1111", "22": "1212", "33": "1313", "44": "1414", "55": "1515"}]
  - Continues printing the received JSON data as it arrives from the server.

CODE

Websocket_server.py

```
import asyncio
import csv
import json
import os
import websockets
import nest_asyncio

nest_asyncio.apply()

# Function to read the CSV file and convert its rows into JSON objects
def convert_csv_to_json(file_path):
rows = []
with open(file_path, 'r') as csvfile:
csvreader = csv.DictReader(csvfile)
for row in csvreader:
rows.append(row)
return json.dumps(rows)

# WebSocket server handler
async def websocket_server(websocket, path):
# Prompt the user to enter the file path
file_path = input("Enter the CSV file path: ")
# Check if the file exists and is a CSV file
if not os.path.isfile(file_path) or not file_path.endswith('.csv'):
print("Invalid file path or file is not in CSV format.")
return

# Continuously send CSV data as JSON to the client every second
while True:
csv_json = convert_csv_to_json(file_path)
await websocket.send(csv_json)
await asyncio.sleep(1)
```

```
# Start the WebSocket server
start_server = websockets.serve(websocket_server, "", 0)
server = asyncio.get_event_loop().run_until_complete(start_server)
server_address = server.sockets[0].getsockname()
print(f"WebSocket server started. Listening on {server_address[0]}:{server_address[1]}.")

# Run the server indefinitely
asyncio.get_event_loop().run_forever()
```

OUTPUT:

```
C:\Users\user\PycharmProjects\Websocket\venv\bin\python.exe
C:/Users/user/PycharmProjects/Websocket/websocket_server.py
WebSocket server started. Listening on 0.0.0.0:59050.
Enter the CSV file path: Book1.csv
```

NOTE:

Book1.csv

11	22	33	44	55
66	77	88	99	1010
1111	1212	1313	1414	1515

Web_socket_client.py

```
import asyncio
import nest_asyncio
import websockets

nest_asyncio.apply()

# WebSocket client handler
async def websocket_client():
    uri = "ws://localhost: 59050"  # Need to update with the correct server port if necessary
    async with websockets.connect(uri) as websocket:
        While True:
```

```
        # Receive JSON data from the server
        json_data = await websocket.recv()
        print(json_data)


# Run the WebSocket client
asyncio.run(websocket_client())
```

OUTPUT:

```
C:\Users\user\PycharmProjects\Websocket\venv\bin\python.exe C:/Users/user/PycharmProjects/Websocket/web_socket_client.py
[{"11": "66", "22": "77", "33": "88", "44": "99", "55": "1010"}, {"11": "1111", "22": "1212", "33": "1313", "44": "1414", "55": "1515"}]
[{"11": "66", "22": "77", "33": "88", "44": "99", "55": "1010"}, {"11": "1111", "22": "1212", "33": "1313", "44": "1414", "55": "1515"}]
[{"11": "66", "22": "77", "33": "88", "44": "99", "55": "1010"}, {"11": "1111", "22": "1212", "33": "1313", "44": "1414", "55": "1515"}]
[{"11": "66", "22": "77", "33": "88", "44": "99", "55": "1010"}, {"11": "1111", "22": "1212", "33": "1313", "44": "1414", "55": "1515"}]
[{"11": "66", "22": "77", "33": "88", "44": "99", "55": "1010"}, {"11": "1111", "22": "1212", "33": "1313", "44": "1414", "55": "1515"}]
[{"11": "66", "22": "77", "33": "88", "44": "99", "55": "1010"}, {"11": "1111", "22": "1212", "33": "1313", "44": "1414", "55": "1515"}]
…
```

Please let me know If any changes needs to be done

Thank you