Connecting websocket to server in python:

import asyncio

import websockets

import json

import time

async def echo\_message(websocket, message):

    for char in message:

        await websocket.send(json.dumps({"type": "echo", "content": char}))

        await asyncio.sleep(0.1)

    await websocket.send(json.dumps({"type": "echo", "content": "\n"}))  # End of message

async def reverse\_message(websocket, message):

    reversed\_message = message[::-1]

    for char in reversed\_message:

        await websocket.send(json.dumps({"type": "reverse", "content": char}))

        await asyncio.sleep(0.1)

    await websocket.send(json.dumps({"type": "reverse", "content": "\n"}))  # End of message

async def count\_last\_character(websocket, message):

    if not message:

        count = 0

    else:

        last\_char = message[-1]

        count = message[:-1].count(last\_char)

    await websocket.send(json.dumps({"type": "count", "content": count}))

async def handler(websocket, path):

    async for message in websocket:

        data = json.loads(message)

        action = data.get("action")

        content = data.get("message")

        if action == "echo":

            await echo\_message(websocket, content)

        elif action == "reverse":

            await reverse\_message(websocket, content)

        elif action == "count":

            await count\_last\_character(websocket, content)

start\_server = websockets.serve(handler, "localhost", 8765)

asyncio.get\_event\_loop().run\_until\_complete(start\_server)

asyncio.get\_event\_loop().run\_forever()

Connecting websocket server to client in python:

import asyncio

import websockets

import json

async def test\_echo(websocket):

    await websocket.send(json.dumps({"action": "echo", "message": " The quick brown fox jumped over the lazy dog o"}))

    async for message in websocket:

        data = json.loads(message)

        if data["type"] == "echo":

            print("Echo:", data["content"])

        else:

            break

async def test\_reverse(websocket):

    await websocket.send(json.dumps({"action": "reverse", "message": " The quick brown fox jumped over the lazy dog o"}))

    async for message in websocket:

        data = json.loads(message)

        if data["type"] == "reverse":

            print("Reverse:", data["content"])

        else:

            break

async def test\_count(websocket):

    await websocket.send(json.dumps({"action": "count", "message": " The quick brown fox jumped over the lazy dog o"}))

    message = await websocket.recv()

    data = json.loads(message)

    print("Count:", data["content"])

async def test():

    uri = "ws://localhost:8765"

    async with websockets.connect(uri) as websocket:

        print("Testing Echo:")

        await test\_echo(websocket)

        print("\nTesting Reverse:")

        await test\_reverse(websocket)

        print("\nTesting Count:")

        await test\_count(websocket)

asyncio.get\_event\_loop().run\_until\_complete(test())

Request and Response:

