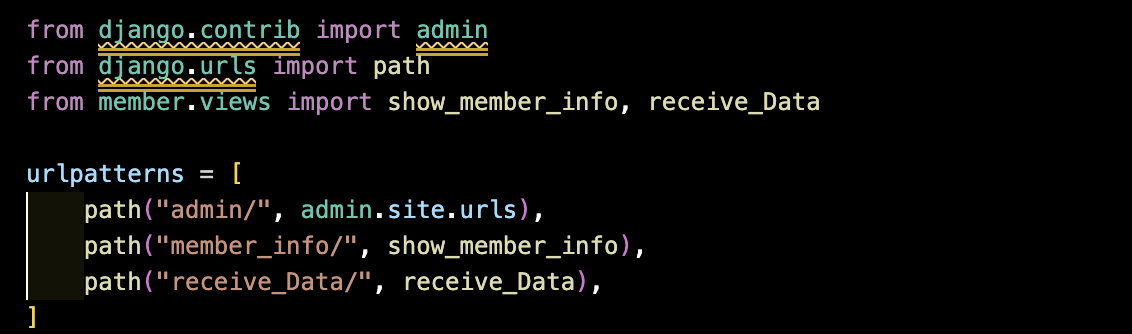
這次測試目的事先將settings.py裡有csrf的那行東西註解掉，藉此證明說在傳輸資料到server時可以先不用準備csrf token

server端程式碼：

如果要新增一個獨立的功能函式出來的話就需要在url中新增相對應的位址：

客戶端程式碼：

從client傳送資料到收到response的過程是：

第一步先啟動server，然後在client中url定義要傳送資料終點的位址，接著在myobj定義要傳送的內容有哪些，headers是定義傳送資料的主體為json格式，x為定義傳送資料的參數，分別有：位址、將myobj定義為json格式傳送、及再次將headers的格式樣式寫入，最後傳送出x.text。

利用print測試程式碼執行狀況

server端接收到後會先分辨傳送方法是否為POST，如果不是就傳回本次方法名稱，如果是POST的話就會開始解碼，解碼後會先印出當下的資料類型後，利用json轉換和讀取檔案內容，將其內容存在name這個變數裡，接著再用HttpResponse回傳題目內容，將name替代為內容物：John，所以最後會回傳 ” Hello, John”

這次作業最大的問題是在測試中途，儲存程式碼的當下會出現” Unexpected error in LSP server, see server's logs for details”這些字，雖然上網查了多資料，但還是不確定他的起因到底是什麼，但因為之前好像有發生過，大概記得有可能是因為用了anaconda 3裡的python當解譯器，所以才會發生錯誤，在前幾次作業我雖然還是用電腦本身的python來用，但也不確定為何，這次一開始用那個python的時候出現了錯誤，才讓我再次用回anaconda的版本，因為他是目前所有python中的最新版，但就是不知道為何，anaconda的python好像有點問題，這次和上次都是因為他的關係，出現了類似錯誤，我再次換回電腦原本安裝的版本後就又都可以正常執行了。

我最後在換回電腦版本的python解譯器之前，因為真的不確定到底哪裡出了問題，所以就又把程式碼丟給chatgpt問問看，看是不是程式碼的問題，只是也看不出來問題點在哪就是了，這次請chatgpt幫忙後最大的改變是server端的name = reg.get(‘Name’, ‘Stranger’)和return = HttpResponse(f”Hello, {name}”)，以及client端的print(sennd\_to\_server(‘John’))，他是說要在server端那行加那串是為了防止後續有空輸入的產生，讓他只要發現有空輸入，就會直接用Stranger來替代原本要輸入到name的部分，再來就是在HttpResponse裡寫入f，讓變數能夠直接再字串裡替換，使得整段回傳程式變的更淺顯易懂也更簡單，最後是client的部分，說來慚愧，我一開始還真的沒想到可以單純寫一串這個來測試本程式，他寫出來之後我才悟到這件事