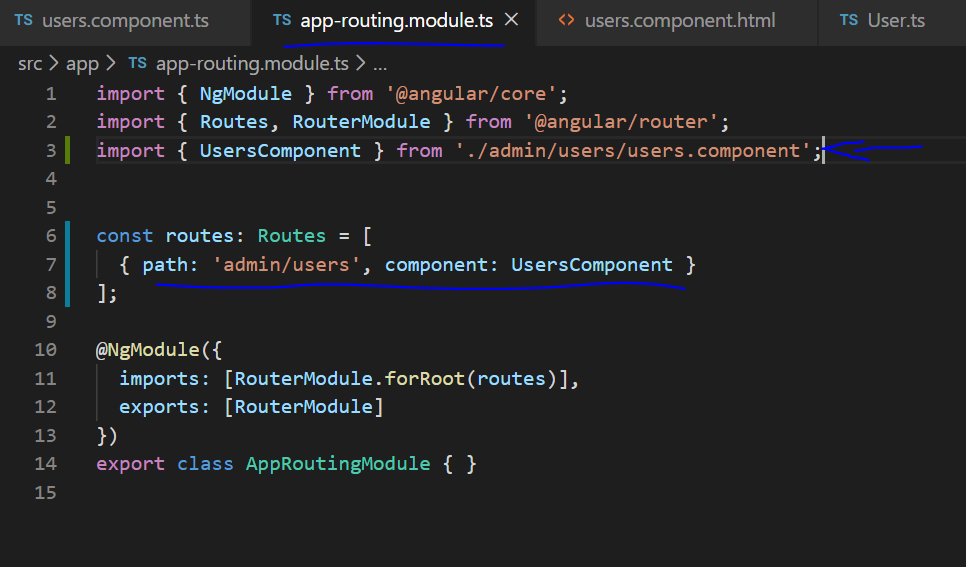
# Crear db en MySQL

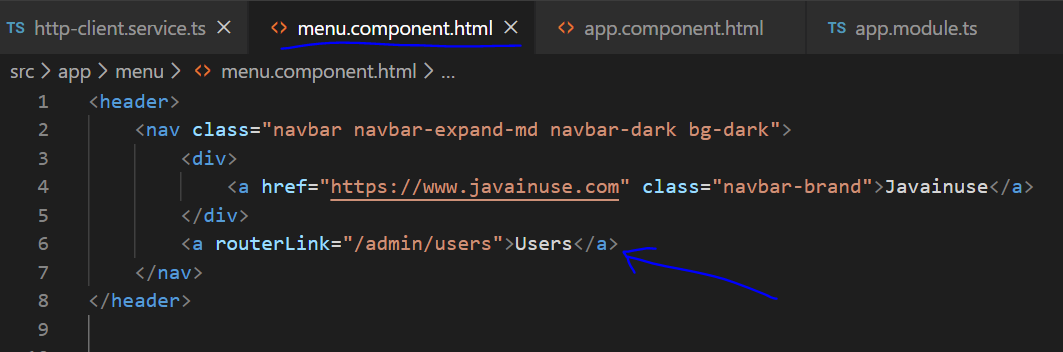
Mirar la documentación doc-accessing-data-mysql

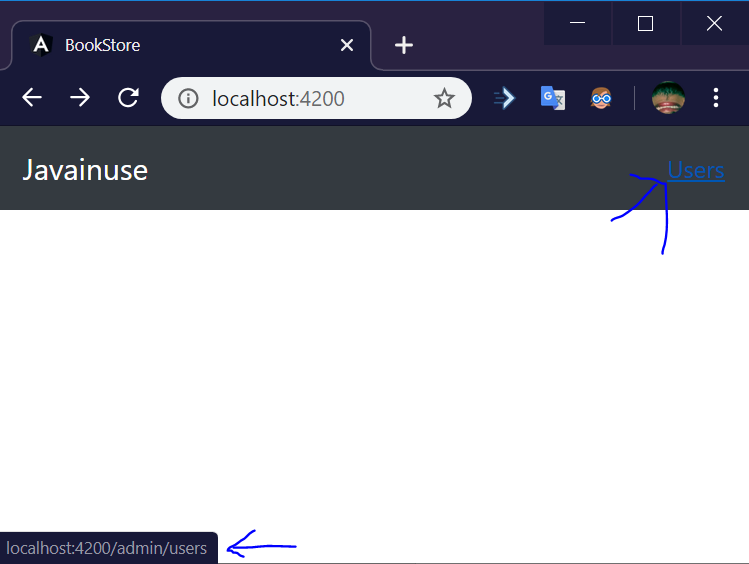
# Router en Angular

Añadimos el componente que vamos a enrutar en la clase “app-routing.module.ts”

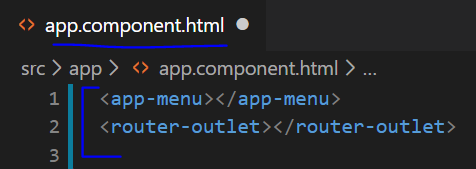


Ponemos en la página principal que se nos muestra el enlace para que llame a nuestro componente y nos muestre su html renderizado.



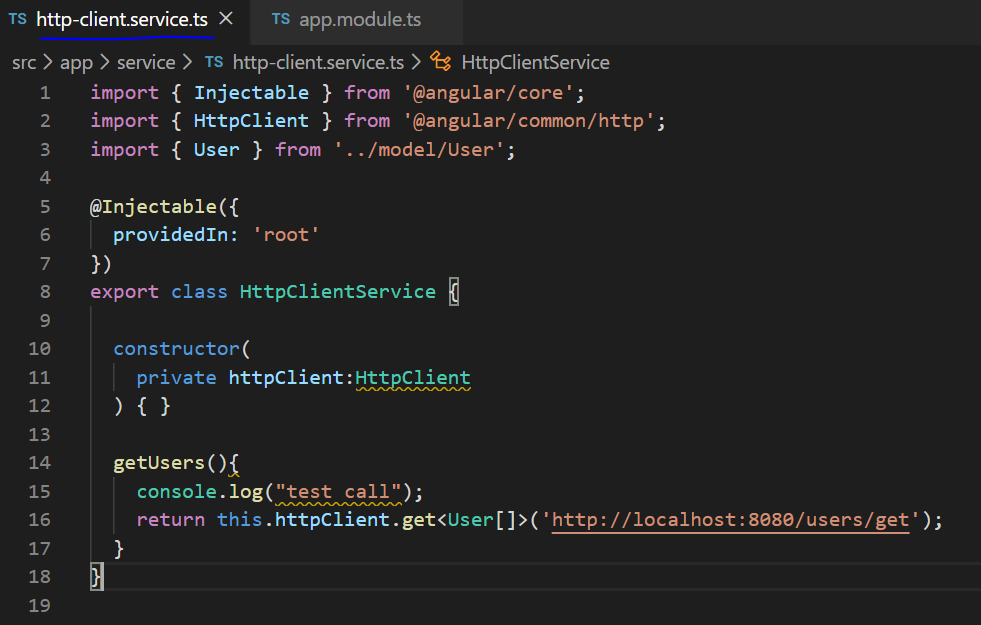


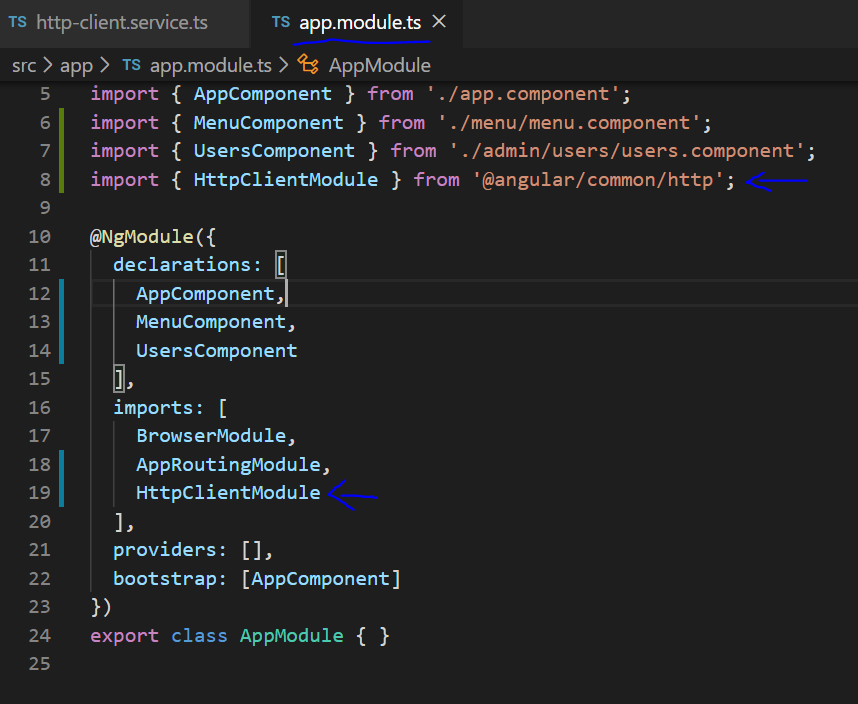
Si nos fijamos hemos añadido el enlace en el html del componente “menú.componente.html”, este componente lo hemos creado nosotros y lo llamamos en el html del componente principal de todos que es el “app.component.html”



# Crear servicio HttpClientService en Angular







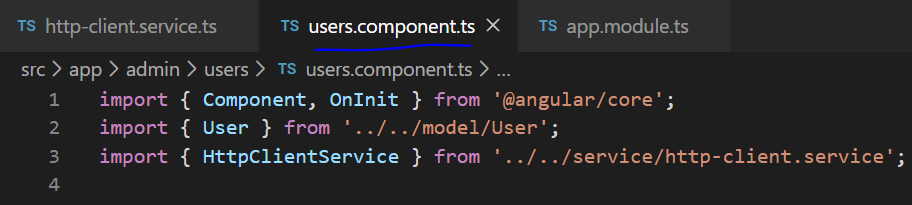
Nota: Si nos saltamos este paso obtendremos el siguiente error por consola:

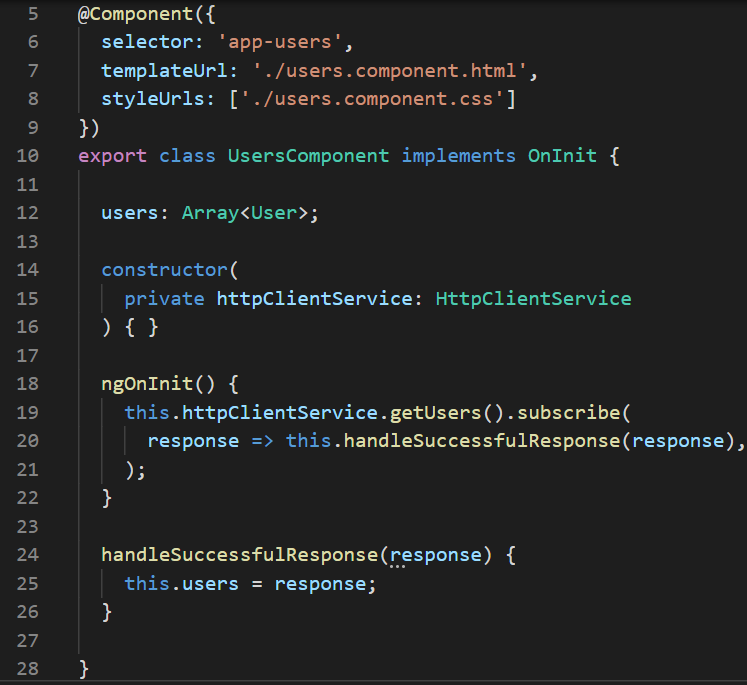


Para solucionarlo tenemos que añadir el módulo HttpClient en la clase ‘app.module.ts’ como se ha realizado arriba.

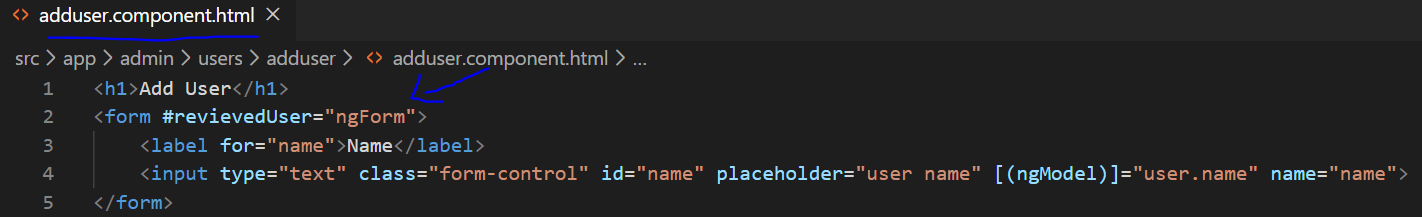
Por último, escribimos el controlador que usará nuestra clase HttpClientService:

Users.component.ts (el controlador que llama al servicio y pone el resultado en la plantilla Users.component.html)

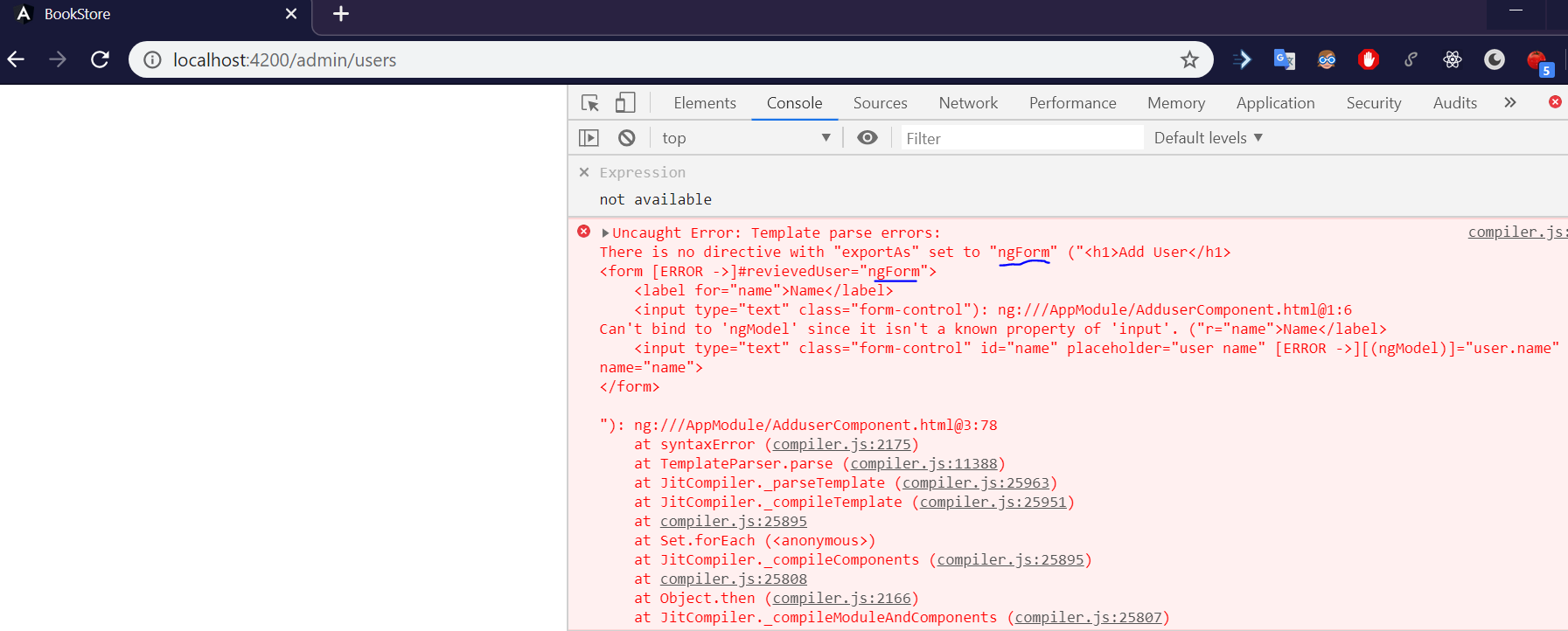




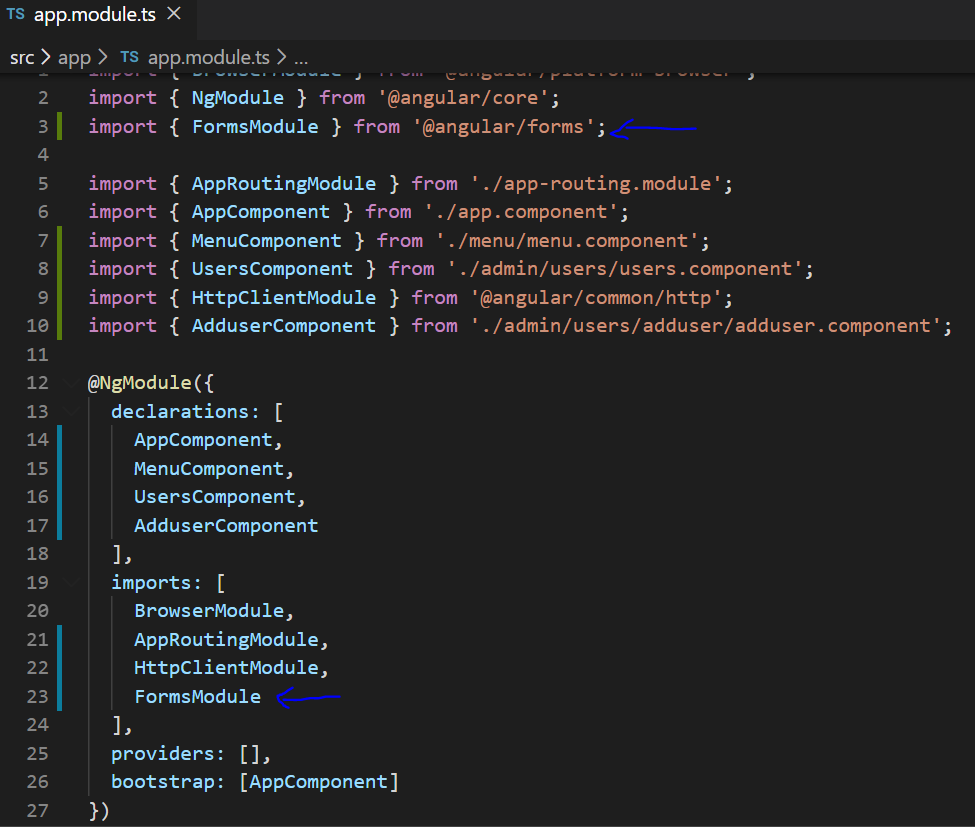
# Usar ngForm en una plantilla html



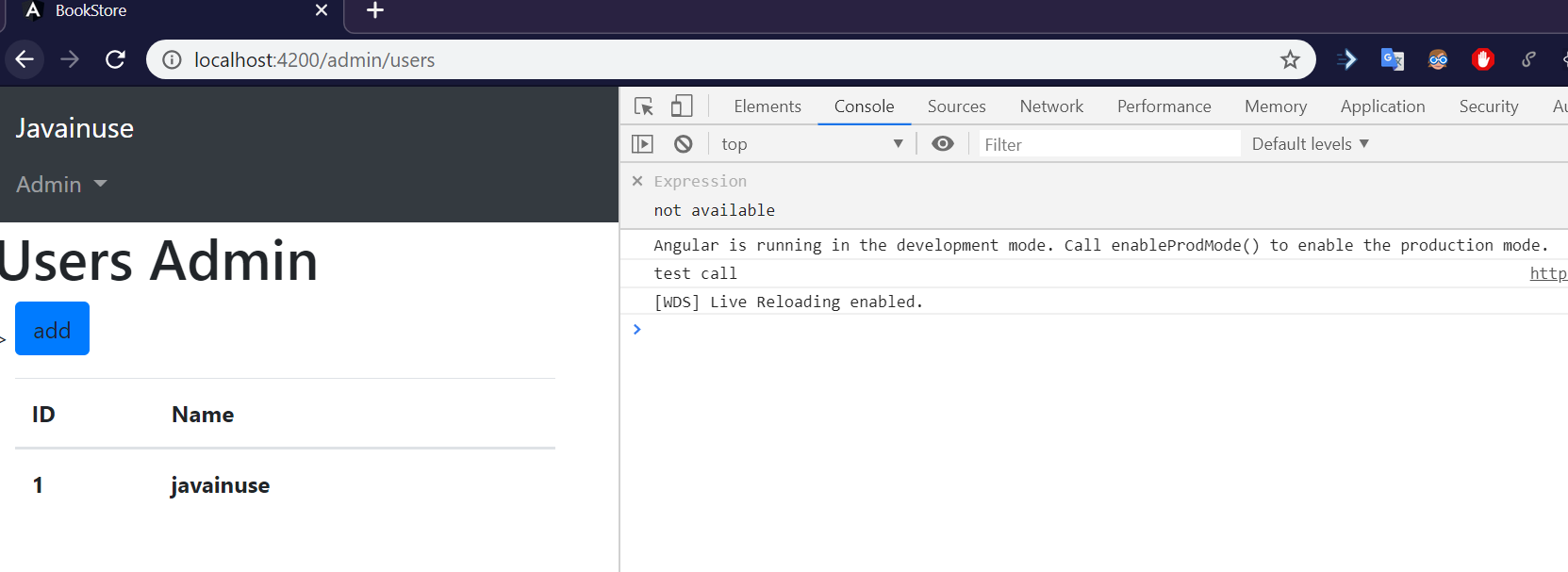
Si miramos la página renderizada en Chrome veremos que no aparece nada y un error por consola



Esto es producido porque no hemos añadido el módulo que nos permite utilizar la palabra “ngForm”

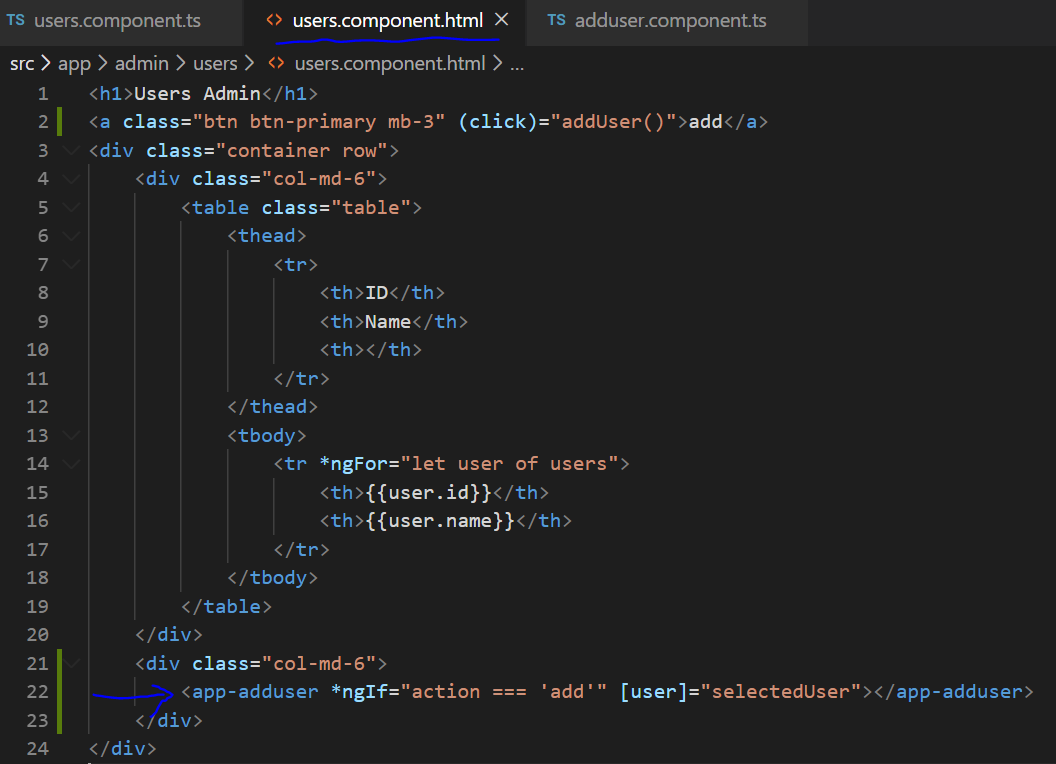


Ya estaría solucionado el error



# Pasar objetos de componente padre a hijo

Entendemos como componente hijo al componente que se llama desde la plantilla html del padre. En este caso tenemos el componente de users que en su plantilla llama al componente adduser por lo tanto el componente adduser pasa a ser el componente hijo del componente padre users.

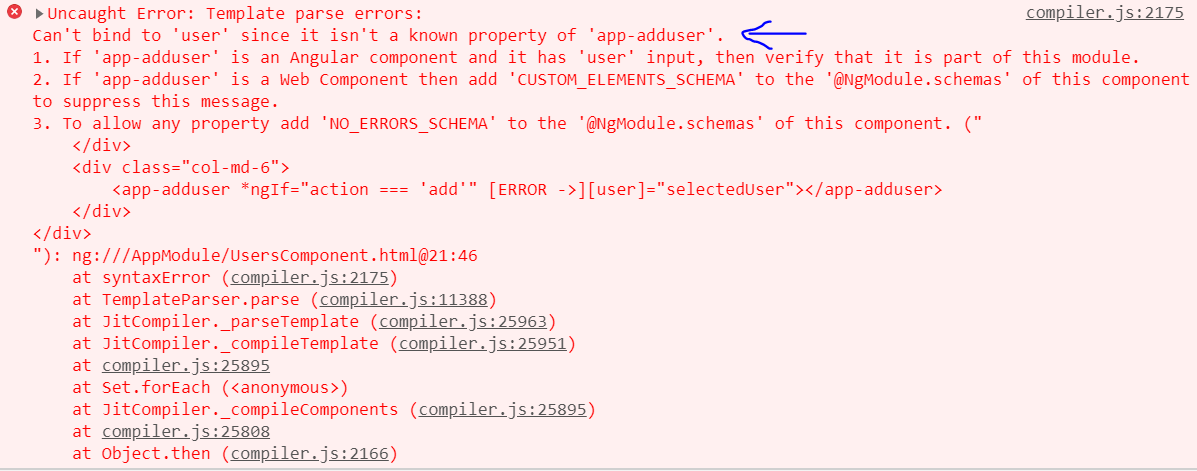


Si vemos la imagen anterior, [user]=”selectedUser”:

[user]: objeto de app-adduser

“selectedUser”: objeto de app-users

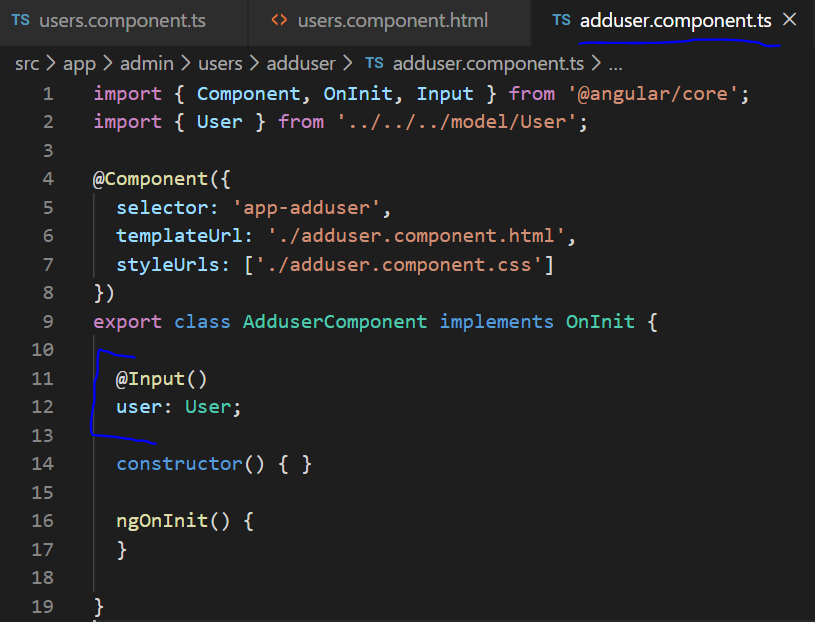
Estamos pasando el objeto selectedUser a su componente hijo. Si vemos el navegador nos aparecerá el siguiente error



Esto es porque:



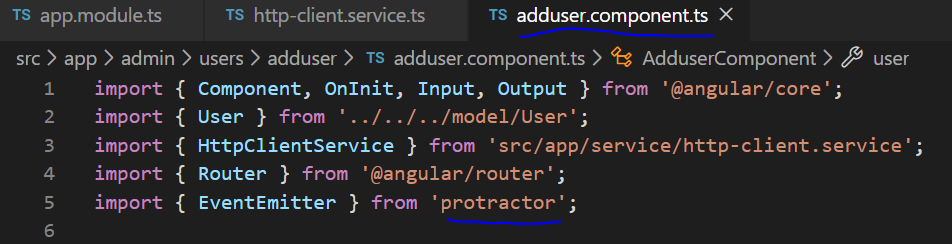
Para poder pasar un objeto de una clase padre a una clase hija, en la clase hija se tiene que definir la propiedad que queremos pasar con la anotación @Input



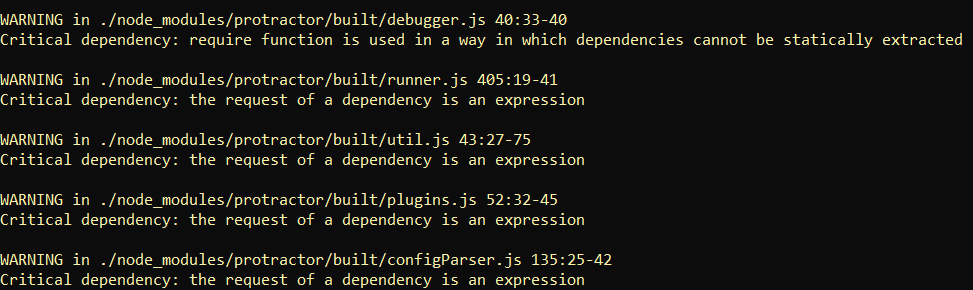
# Errores

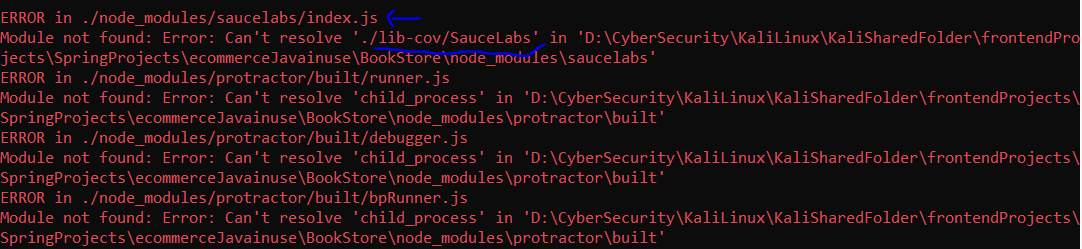
## Error1

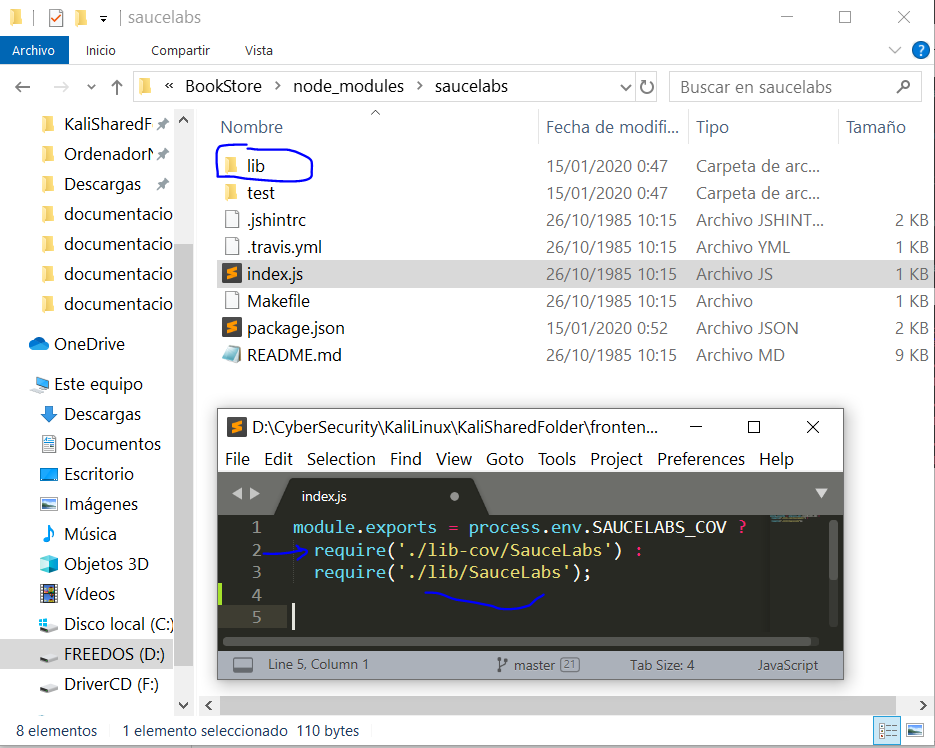
Hemos usado la librería protractor pero no la hemos instalado con ‘npm install protractor’



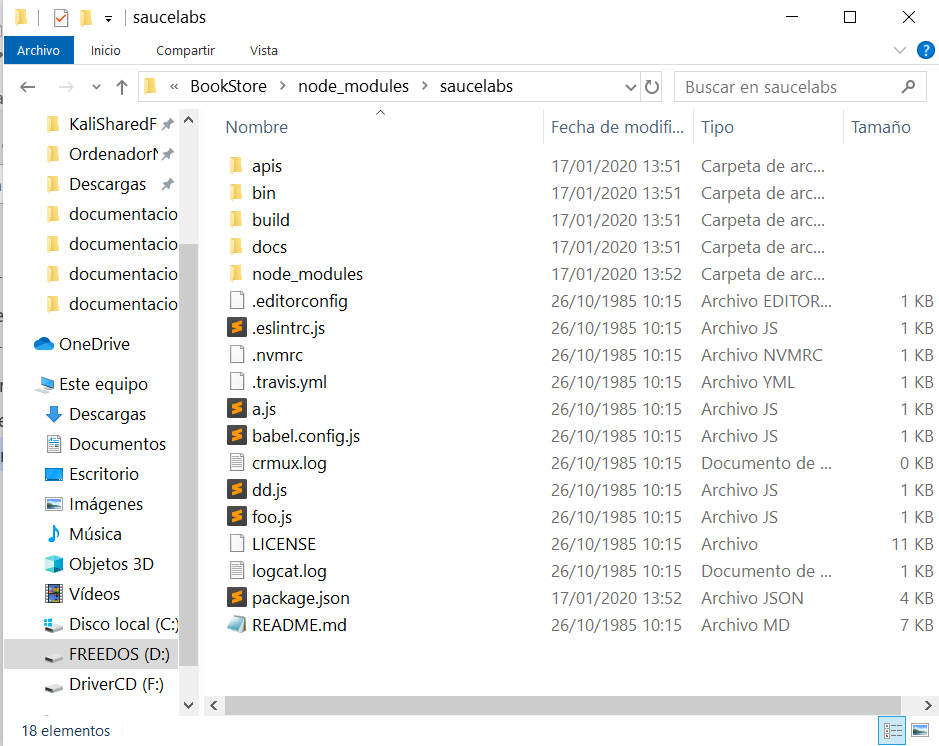
Una vez instalada, me sigue dando problemas



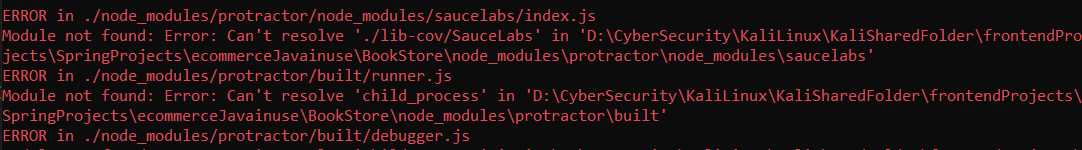




La carpeta ‘lib-cov’ no existe y eso puede ser lo que esté fallando. La que existe es la carpeta ‘lib’ y no sé por qué el código que se ejecuta es el que pide la carpeta que no existe en vez de la carpeta que sí existe. Vamos a probar a volver a instalar la dependencia ‘saucelabs’ con el comando ‘npm install saucelabs’.



La carpeta ha cambiado completamente y sigue dando el mismo error al arrancar el servidor

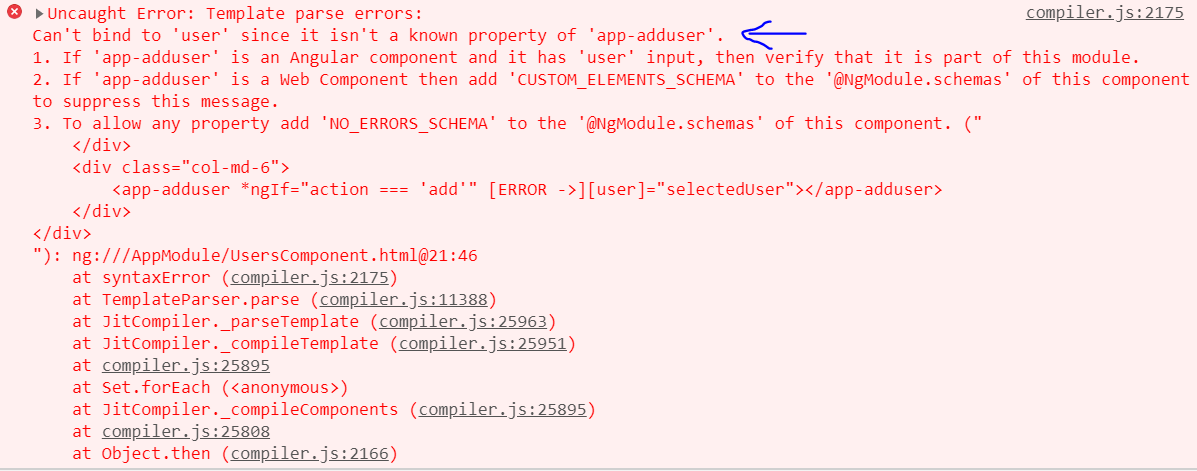


## Error2



Explicado en el punto “Crear servicio HttpClientService”.

## Error3



Explicado en el punto “Pasar objetos de componente padre a hijo”