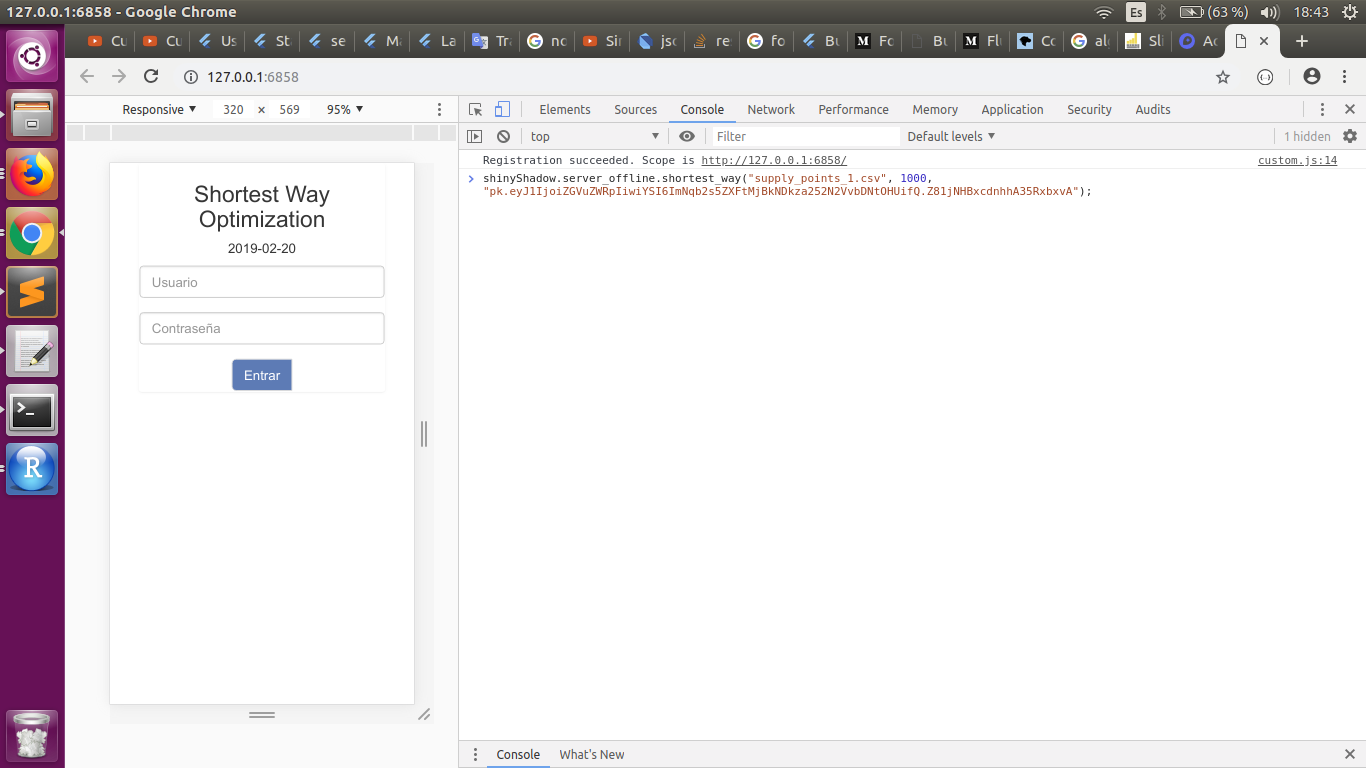
**DISTANCIA MAS CORTA**

La ruta del código esta en: "~/Rabies/shortest\_way\_optimization/www/petm\_offline.js"

En la carpeta "~/Rabies/shortest\_way\_optimization/" se ha dividido los supply\_points en 10 archivos, cada archivo consta de 7 puntos, esto para poder ejecutarlo en menos tiempo y con un usuario diferente.

Lo único que hace falta es crear usuarios en "<https://account.mapbox.com>" aqui te proporcionan un token, con ese tokens es que se tiene que ejecutar las consultas. Tiene que ser un usuario distinto por consulta para que no nos penalicen.

Lo siguiente que se tiene que hacer es ejecutar el archivo server.R, abrirlo en un navegador y presionar la tecla F12 para que podamos acceder a la consola del navegador donde podremos ejecutar nuestro código javascript.



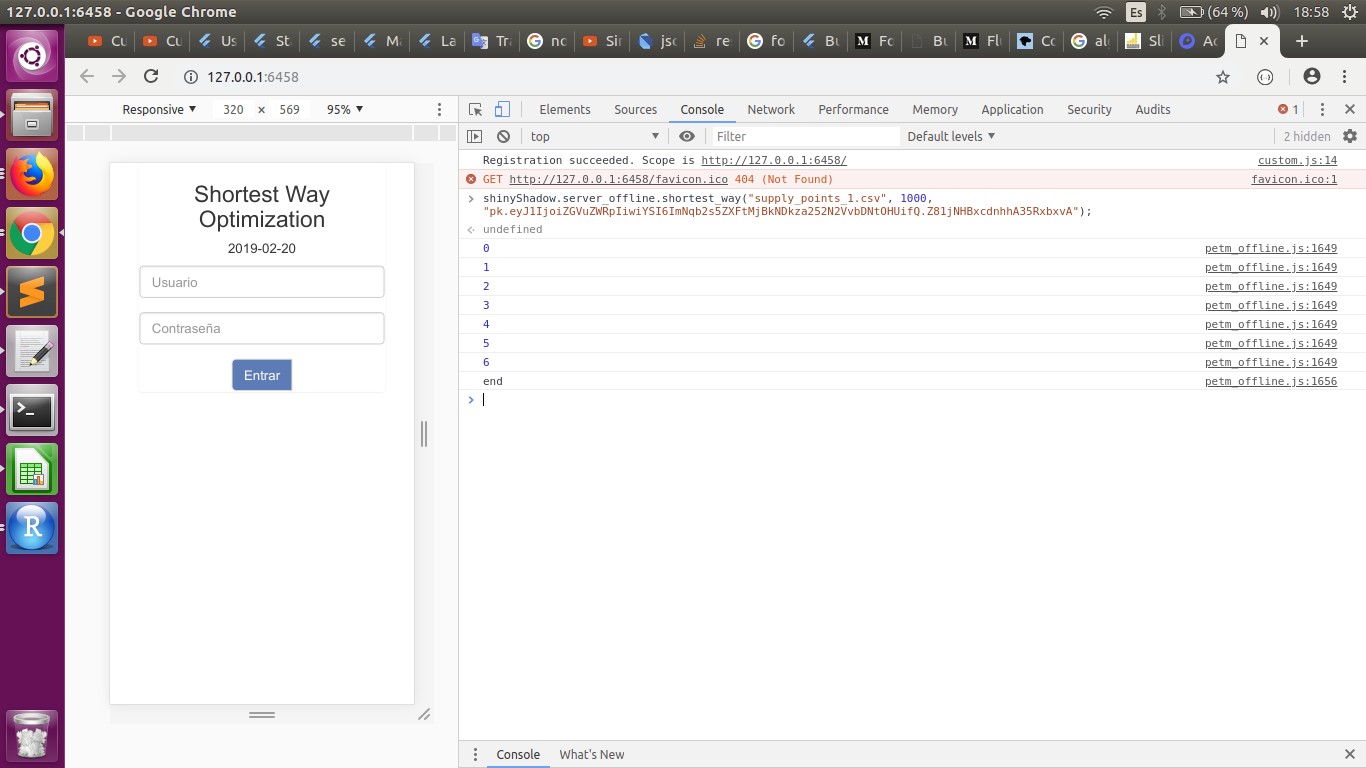
Luego para hacer funcionar nuestro código tenemos que poner en consola:

1.- *shinyShadow.server\_offline.shortest\_way("supply\_points\_1.csv", 1000, "tokens\_de\_cada\_usuario");*

* "supply\_points\_1.csv" => Aquí se pone el supply\_points que queremos ejecutar.
* 1000 => Es el tiempo en que va a realizar cada consulta
* "tokens\_de\_cada\_usuario" => Aqui se pone el token que se te da cuando creas un usuario

NOTA: Recomiendo realizar este procedimiento en cada navegador con distinto supply\_points y con distinto usuario, en mi maquina tengo instalado el navegador MOZILLA y CHROME y si abro los navegadores incognitos de cada uno tendria la opcion de correr hasta 4 supply\_points por máquina, el proceso demora aproximadamente 3 a 4 hora, así me demoro correr el supply\_points\_1.csv. Internamente ya está relacionado con los demands\_points que son aproximadamente 25mil registros.

2.- Cuando a finalizado la instrucción anterior en la consola empezará a aparecer numeros en consola desde 0 hasta 6 u al finalizar aparecerá "end", esto significa que la ejecución anterior a finalizado.



Cuando esto sucedase tiene que ejecutar el siguiente codigo en consola:

*shinyShadow.server\_offline.post\_shortest\_way();*

Este código hará que se cree el CSV con la matriz de distancias, similar al que yo pude crear con supply\_points\_1 de nombre "result\_shortest\_way\_2019-02-19\_08-14-51.csv". Al crearse dicho archivo el procedimiento habrá finalizado.

NOTA: Estas funciones estan definidas en el archivo “petm\_offline.js”

TOKENS OPCIONALES:

1. pk.eyJ1IjoiYWJjc2F4MzQ5OSIsImEiOiJja2Z5Z2gwOHcwOTdpMzRwajN3aWs3cTd2In0.C350P\_IhBq3M4BvUtPdg5A

2. pk.eyJ1IjoiYXJpc3NheDAzMDQ5OSIsImEiOiJja2Z5Z2tvejMwNDdhMndwaWlzOXUxcmw2In0.kOCkKYU5lKby1LF9GIHTzQ

**PARA EJECUTAR NUEVOS ARCHIVOS:**

Para este ejemplo voy a utilizar nombres de archivos que no existen:

* demandpoint va a ser el archivo: “test\_demandpoint.csv”
* supplypoint va a ser el archivo : “test\_suplypoint.csv”

1.- Revisar que el nuevo archivo “test\_demandpoint.csv” tenga las mismas columnas que los anteriores demandpoints (UNICODE, lat, long).

2.- Revisar que el nuevo archivo “test\_supplypoint.csv” tenga las mismas columnas que los anteriores supplypoints (VaccPoint, PointName, lat, long).

3.- Poner el archivo que corresponde al demanpoint en la carpeta: “Rabies/shortest\_way\_optimization/demandpoints/”

4.- Ir al archivo server.R en la linea 831 y poner el nuevo nombre del archivo demandpoint:

response$postas <- read.csv("demandpoints/test\_demandpoint.csv", sep = ",")

5.- Poner el archivo que corresponde al supplypoint en la carpeta : “Rabies/shortest\_way\_optimization/supplypoints/”

6.- Ejecutar el proyecto y continuar de forma que ya se conoce.