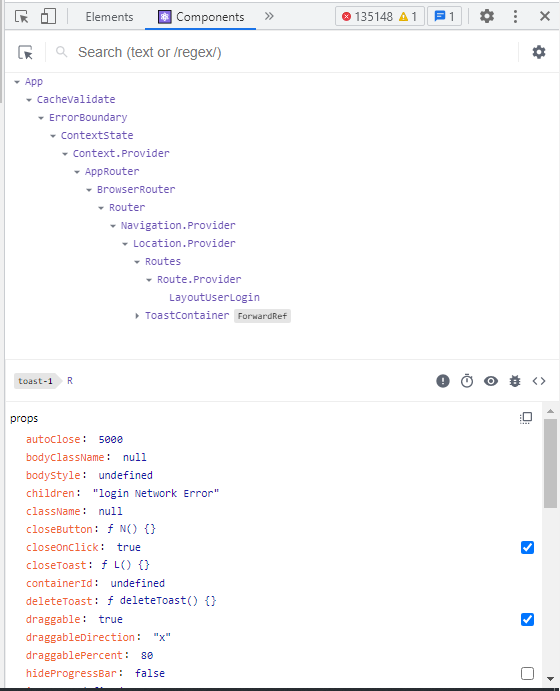
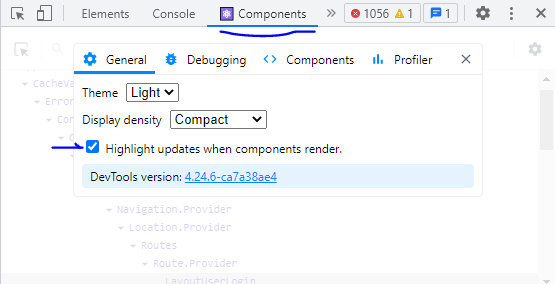
# REACT DEV TOOLS

Apartado “Components” de la extensión donde se puede ver en forma de árbol la estructura que tenemos de componentes:



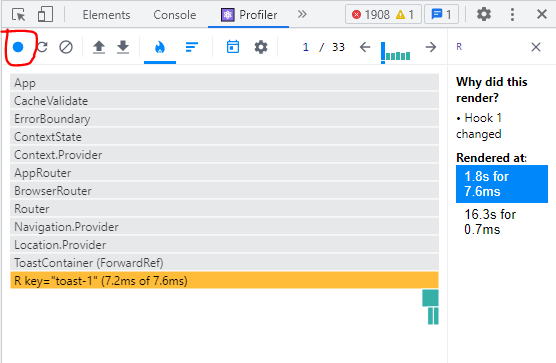
Para sacarle el máximo provecho a la extensión deberemos configurar lo que indico para que nos de más información útil.

Marcar este checkbox que se encuentra dandole click al botón de configuración que encontramos en “Components” (Tanto esta pestaña como la de “Profiler” se añadiran automaticamente al instalar la extensión)



Este parámetro lo que va a hacer es estar alerta para que en el momento que nosotros interactuemos con la web y se genere un render nos lo remarque, de esta manera podemos detectar renders innecesarios y así poder evitarlos.

Por otra parte tenemos la pestaña del “Profiler”:



Desde este apartado en el momento que le demos al botón que indico en la captura, la extensión empezará a auditar los renders que se realizan y el motivo de ellos, en mi caso como podéis ver en “R key=”toast-1” se ha realizado un render a causa de que se ha modificado un Hook.

# EVITAR WASTED RENDERS

**React memo:**

Una de las opciones que tenemos es React Memo, este realiza una comparación de las props que recibe el componente y además de ello nos da la opción de que por medio de un segundo argumento podamos indicar nosotros mismos la comparación que queremos realizar. Para que en el momento que la comparación no de resultado true se vuelva a renderizar el componente.



**React pure components:**

La clase React.PureComponent cuenta con una implementación por defecto del método “shouldComponentUpdate” con una [“shallow comparison”](https://reactjs.org/docs/shallow-compare.html) de las props y el estado. Por tanto, sólo vuelve a renderizarse si las props o el state cambian.



# AÑADIR EFECTO EN RENDERIZADO

Para realizar esta función se utiliza “useEffect”, es un hook que recibe como parámetro una función que se ejecutará cada vez que nuestro componente se renderice, ya sea por un cambio de estado, por recibir props nuevas o, y esto es importante, porque es la primera vez que se monta (En algún caso podría ser conveniente sustituir el ComponentDidMount por este).

import React, { useEffect } from 'react'