Web Browser

Silvia Rodriguez y Julia Figueredo

23 de agosto de 2021

Resumen

1. Introducción

En este trabajo haremos un resumen sobre los navegadores web, los cuales son el software más utilizados actualmente. Tocaremos los 5 puntos técnicos más relevantes de cómo funcionan.

2. Cinco puntos técnicos más resaltantes y transversales

2.1. Los componentes principales del navegador son:

*La interfaz de usuario: Esto incluye la barra de direcciones, el botón de retroceso / avance, el menú de marcadores, etc. Cada parte de la pantalla del navegador, excepto la ventana donde se ve la página solicitada.

*El motor del navegador: calcula las acciones entre la interfaz de usuario y el motor de renderizado.

*El motor de renderizado: responsable de mostrar el contenido solicitado. Por ejemplo, si el contenido solicitado es HTML, el motor de renderización analiza HTML y CSS y muestra el contenido analizado en la pantalla.

*Redes : para llamadas de red como solicitudes HTTP, utilizando diferentes implementaciones para diferentes plataformas detrás de una interfaz independiente de la plataforma.

*Backend de interfaz de usuario : se utiliza para dibujar widgets básicos como cuadros combinados y ventanas. Este backend expone una interfaz genérica que no es específica de la plataforma. Debajo, utiliza métodos de interfaz de usuario del sistema operativo.

*Intérprete de JavaScript: Se utiliza para analizar y ejecutar código JavaScript.

*Almacenamiento de datos: Esta es una capa de persistencia. Es posible que el navegador necesite guardar todo tipo de datos localmente, como cookies. Los navegadores también admiten mecanismos de almacenamiento como localStorage, IndexedDB, WebSQL y FileSystem.

Entre éstos el más destacado es el sgte.:

2.2. El motor de renderizado

La responsabilidad del motor de renderizado es mostrar los contenidos solicitados en la pantalla del navegador.

De forma predeterminada, el motor de renderizado puede mostrar documentos e imágenes HTML y XML. Puede mostrar otros tipos de datos a través de complementos o extensiones; por ejemplo, mostrar documentos PDF mediante un complemento de visor de PDF.

El motor de renderizado comenzará a analizar el documento HTML y convertirá elementos en nodos DOM en un árbol llamado "árbol de contenido". El motor analizará los datos de estilo, tanto en archivos CSS externos como en elementos de estilo. La información de estilo junto con las instrucciones visuales en el HTML se utilizará para crear otro árbol: el árbol de renderizado . El árbol de renderizado contiene rectángulos con atributos visuales como color y dimensiones. Los rectángulos están en el orden correcto para mostrarse en la pantalla.

Después de la construcción del árbol de renderizado, pasa por un proceso de "diseño ". Esto significa darle a cada nodo las coordenadas exactas donde debería aparecer en la pantalla. La siguiente etapa

es la pintura: se recorrerá el árbol de renderizado y cada nodo se pintará utilizando la capa de backend de la interfaz de usuario.

2.3. Árbol DOM y árbol CSSOM

El árbol DOM es un árbol de análisis HTML que se reformatea para su consumo en el navegador.

El árbol de análisis sintáctico de CSS no es exactamente el árbol CSSOM. El árbol CSSOM es un árbol de análisis CSS que se formatea para su consumo en el navegador. Cada nodo CSSStyleDeclaration es un diccionario que tiene entradas para cada propiedad CSS posible . Para las propiedades configuradas, el valor es el valor dado. Para las propiedades no configuradas, el valor es simplemente vacío.

2.4. Árbol renderizado

Mientras se construyen el árbol DOM y el árbol CSSOM, se combinan para formar un tercer árbol: el árbol de renderizado. Este árbol contiene nodos visuales (es decir, cosas que realmente aparecerán en la página).

La información visual se calcula a partir de las propiedades de estilo de cada elemento. Los contextos de estilo se dividen en estructuras. Una estructura es esencialmente una propiedad CSS (Por ejemplo border, color, font-size, etc.) Recuerde que renderizar + pintar es un proceso gradual. El navegador no esperará a analizar cada documento HTML y hoja de estilo antes de renderizar + pintar.

Si los estilos de un nodo DOM dado no se han cargado (debido a la latencia de la red) en el momento en que se llama a attach(), se utilizan los estilos de marcador de posición predeterminados. El nodo DOM está marcado y sus estilos se actualizarán cuando se carguen.

2.5. Evolución de los navegadores

Antes de que existieran los navegadores modernos, usábamos el navegador de proceso único. IE6 fue el más popular. En un navegador de un solo proceso, el proceso debe encargarse de todo, incluida la representación de la página, la ejecución de JavaScript y más. Esta arquitectura trae tres problemas: inestabilidad, bajo rendimiento e inseguridad. En un navegador multiproceso, bueno, tenemos múltiples procesos. Entre procesos, se comunican con la comunicación entre procesos (IPC). Cada proceso ejecuta varios subprocesos. Desde Chrome 67, todos los iframes entre sitios obtienen un proceso de renderizado independiente. En 2019, Firefox inicia Project Fission, un proyecto relacionado con la integración de la función de aislamiento del sitio. Chrome ahora tiene más procesos que se ejecutan por separado. La arquitectura orientada a servicios fue inicializada por el equipo de Chrome, introduciendo Chrome Foundation Service en la arquitectura. Es una arquitectura modular. SOA es un movimiento audaz y el equipo tardará años en completar el cambio. Actualmente, el equipo está mejorando progresivamente el navegador hacia SOA.

Referencias

https://www.html5rocks.com/en/tutorials/internals/howbrowserswork/
https://codeburst.io/how-browsers-work-6350a4234634
https://cabulous.medium.com/how-browser-works-part-i-process-and-thread-f63a9111bae9