

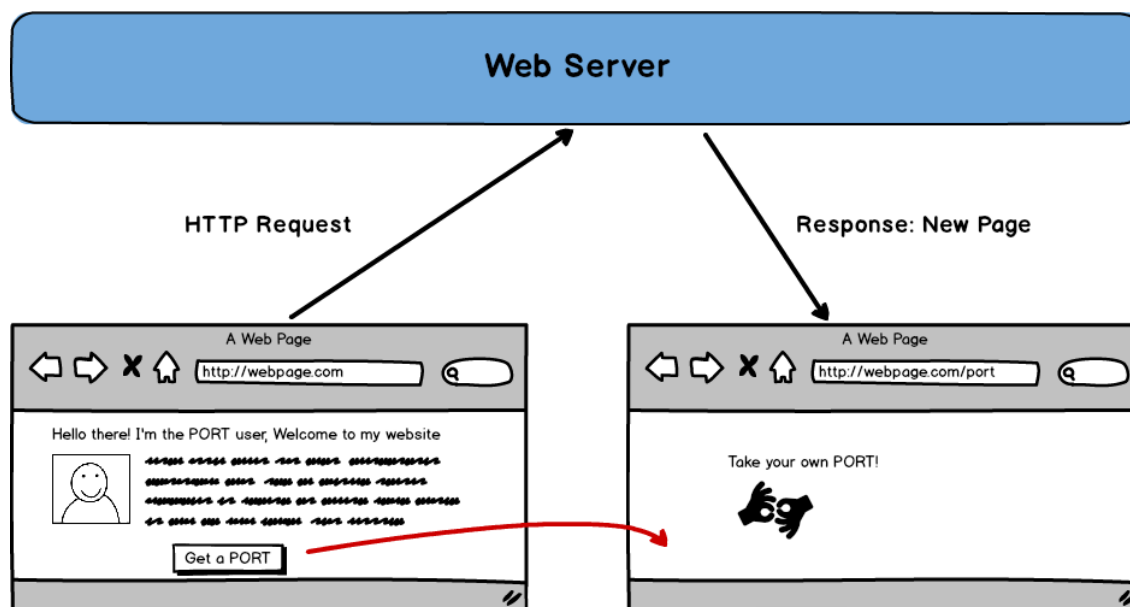


Single Page Application SPA

Ing. Luis Guillermo Molero Suárez

Antecedentes

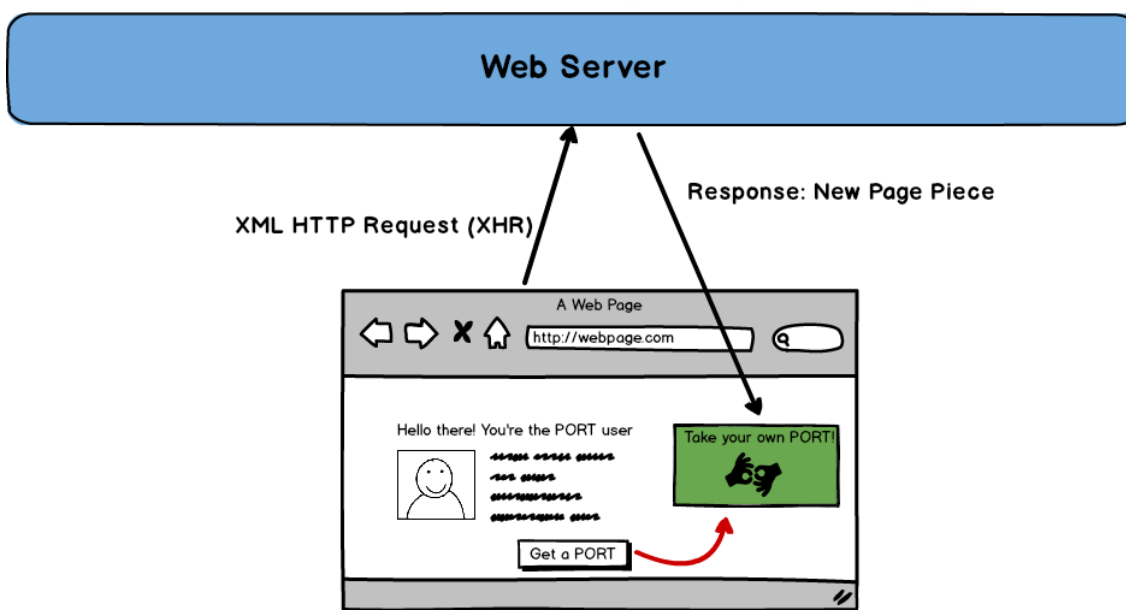
Antes todo esto era campo. Relación Cliente — Servidor, tal como se muestra continuación, lo que podría llamarse un escenario SSR (Server Side Rendering): Renderizado del lado del servidor, es decir, la capacidad para construir el etiquetado de páginas y vistas con datos y resultados que se lanzarán al navegador y este se limitará a pintar.



Classical SSR Scenario (Server Side Rendering): Full Page Response

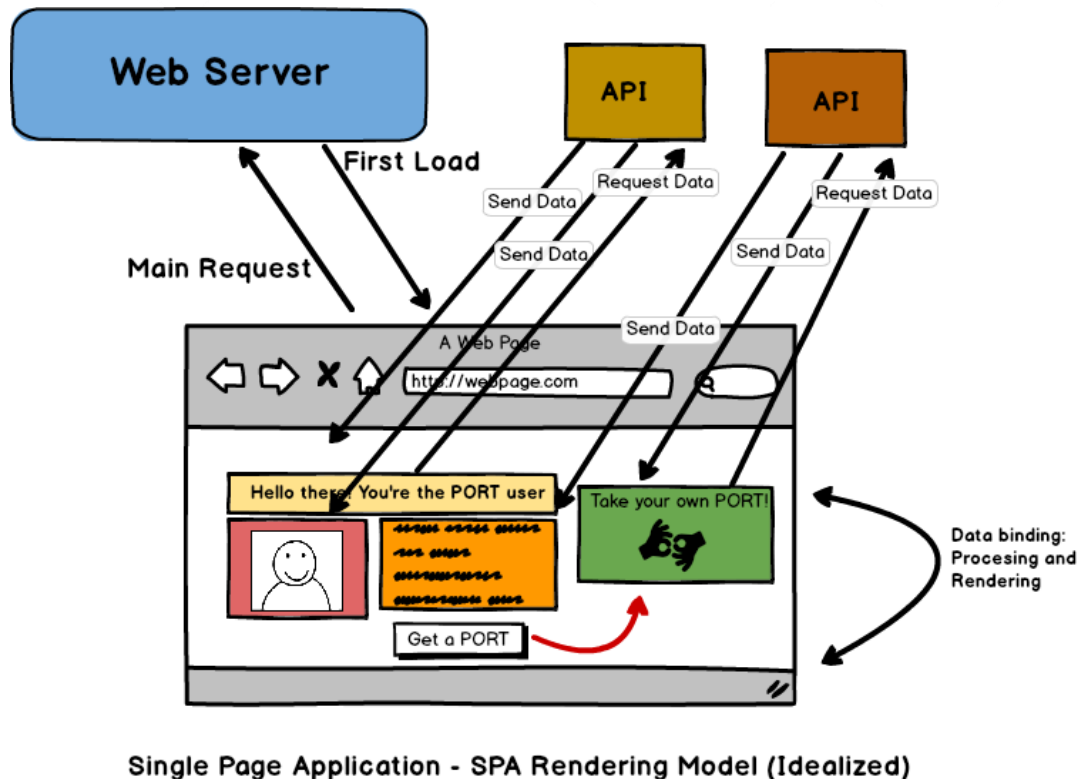


AJAX, realiza peticiones parciales de AJAX a través de Javascript que servían para actualizar secciones específicas, repintar bloques con datos actualizados, modificar vistas, etc. Esto hacía un SSR modificado, algo más liviano y llevadero:



Typical Partial Rendering (AJAX): XMLHttpRequest - Response

SPA: Desacoplamiento MVC donde las vistas son bloques parciales que obtienen o ingresan datos al modelo y se actualizan de manera individualizada:



Single Page Application - SPA Rendering Model (Idealized)

Caracterizando una SPA

Inicialmente, SPA son aplicaciones web creadas sobre una sola página que muta mediante interacciones Javascript.

Intuitivamente, es un tipo de aplicación que consiste en mostrar todas sus pantallas en la misma página, sin realizar sucesivas cargas de otras direcciones / secciones de la plataforma provenientes del servidor al cambiar de dirección interna URL.

Conceptualmente, es un sitio web que recarga y muestra su contenido en respuesta a acciones propias de la navegación (enviar un formulario, clickar en un enlace, acceder a una sección interna...) sin tener que realizar peticiones al servidor para volver a cargar más código HTML.



Filosóficamente, es ofrecer una experiencia de usuario similar a la de una aplicación de escritorio, pero desde un navegador, manejando datos, flujos y estructuras que dependen de una conectividad a Internet en menor medida que el resto de aplicaciones web tradicionales, de una manera ágil y grata, ofreciendo una experiencia de usuario muy cómoda, rápida y agradable.

Funcionalmente, podríamos describir varias características de una SPA:

- **Punto de entrada central:** Un punto de entrada único, uno que se va transformando y adaptando mediante acciones.
- **Página fija, Vistas cambiantes:** Como en el caso de una aplicación de escritorio, nos mantenemos en un “marco único” y fijo, mientras que “vistas dinámicas” van ofreciéndonos las distintas posibilidades del uso y navegación.
- **Página fija, no URL fija:** Es posible que la dirección URL sufra cambios en base a las actividades de uso de la plataforma y vaya modificándose, aunque ese “marco único” se mantenga fijado.
- **Viajar ligero de equipaje:** Las peticiones cliente — servidor tienden a ser más laxas y livianas que en el caso de una plataforma web al uso. Datos, solo datos. Y además muchos procesos quedan del lado del navegador web del cliente a partir de diversas herramientas (como LocalStorage, por ejemplo). Datos y preferentemente en formato JSON.
- **Ejemplos de aplicaciones SPA:** Gmail, Netflix, Trello, Github...

Diez claves fundamentales de una SPA

Integración: Básicamente, una SPA debe fusionar lo mejor de las dos maneras clásicas de entender el contexto de uso de las aplicaciones. Por un lado, la rapidez y la fluidez de una aplicación de escritorio y por otro, la portabilidad de una aplicación web.

Sin viajes ida-vuelta al servidor: Una SPA que se digne de serlo es capaz de repintar cualquier parte de la UI del cliente sin realizar una petición completa para renderizar una página HTML, separando datos de presentación.

Reduce los tiempos de respuesta: Si estamos ante una SPA, entonces estamos tratando (o deberíamos) con una reducción de los tiempos de respuesta habituales de la aplicación, debido a la delegación de procesos del servidor al navegador web. Que solo queden en el servidor remoto procesos relacionados con autenticación, validación de datos, o almacenamiento persistente. Ese es el objetivo.

Reduce el consumo de ancho de banda: Las conexiones de una SPA con un servidor deben ser en pequeños tramos muy puntuales (en términos generales, serán constantes si hablamos de streaming en VOD, por ejemplo). Pero en general, los datos serán transferidos en objetos manejables tipo JSON.

Actualizaciones automáticas y distribuidas: Como un website, los usuarios no tienen que realizar ninguna acción para actualizar el software, solo recargar la dirección de la aplicación, sin más.



Informa de su estado al usuario: Como en una aplicación de escritorio, reporta al usuario cuando no está saliendo como se espera, o bien si queda “*waiting for*”, esperando una respuesta, muestra elementos de UX suficientes como para conocer qué ocurre: barras de progreso, mensajes de estado, avisos de errores/esperas, etc.

Routing eficaz: Igualmente, debe ser capaz de construir bien el seguimiento del estado y localización (state and location) del usuario, a lo largo de su navegación.

Cross-Platform: disponible para cualquier dispositivo; Tablets, smartphones, portátiles, sobremesa, televisiones inteligentes, consolas de videojuegos, entre otros.

Flexibilidad y rendimiento: Debe transferir todo lo relacionado con Interfaz de Usuario al cliente, vía Javascript, para reducir el consumo de ancho de banda y las respuestas offline de la plataforma funcionando “*a red caída*”.

Facilidad para desconexión — conexión: Debe resultar grata al perder conectividad y también debe ser responsable de realizar todas las sincronizaciones necesarias entre cliente — servidor en cuanto se restablezca la conexión a la red.

Cómo implementan las SPAs estas claves

“A través del ensamblado de una SPA, quien tiene tres recursos básicos proporcionados por la relación entre HTML5 y Javascript para realizar este tipo de tareas. “

Todas se sirven del BOM (Browser Object Model) para poder relacionarse con el navegador gracias a los métodos de sus APIs. Las tres APIs elementales para hacer SPA:

Location API: Se encarga de procesar información acerca de la ubicación de objetos dentro del sistema recogiendo parámetros de la dirección URL.

History API: Es una colección de métodos con la que los navegadores te permiten interactuar tanto con la barra de direcciones como con el historial de navegación.

Web Storage API: Proporciona mecanismos de almacenamiento de datos en formato clave/valor dentro del navegador web.

“Una web SPA o single page application se refiere a una **forma de desarrollo web en la que la página web está contenida en un único archivo.**”

Es decir, **todo el contenido html, css y javascript de la página web al completo se carga una primera vez y nada más.** De esta forma se mejoran los tiempos de respuesta y, por consiguiente, la experiencia de usuario.



En una SPA tenemos una única página, pero contaremos con diferentes vistas, es decir,
distintos apartados de nuestra web

En qué lenguaje de programación se hacen las SPA

Una SPA se creará siempre en **Javascript**. Ya que, al ser una aplicación web ejecutada del lado del cliente, no hay otro lenguaje que pueda hacer eso. A esto habrá que sumarle, lógicamente, el **HTML y CSS**.

Dentro de Javascript, contaremos con multitud de **librerías y frameworks** que nos facilitarán el desarrollo de las SPA. Algunas de las más usadas son:

- AngularJS
- Angular
- VUE
- React
- EmberJS
- Polymer

Ventajas de las páginas web SPA

- Fáciles de desarrollar, desplegar y depurar.
- Muestran el contenido de forma sencilla y elegante al cargar todo el contenido en una única página.
- Ejecutando la lógica del lado del navegador, hace que las interacciones del usuario se ejecuten más rápido.
- La analítica de la web es mucho mas sencilla puesto que solo analiza una única página.
- También facilitará el SEM puesto que todas las llamadas y posibilidades de interacción se realizarán en una única página.