

TP5 - Frontend Client Application

Dimase, Matias - Isea, Maximiliano

1. ¿Qué es una Dynamic Web Page?+

Son aquellos que permiten crear aplicaciones dentro de la propia web, como encuestas, votaciones, foros, reservas, compras online, etc. Están basadas en generar contenido a través de la información introducida por el usuario. Proveen una mayor interactividad con el usuario, dado que puede alterar el contenido de la página mientras la utiliza. Se caracterizan por brindar funcionalidad (mediante lenguajes de programación), procesando y almacenando los datos ingresados por el usuario. La actualización del contenido se realiza desde la página y no necesariamente desde el servidor.

2. ¿Qué es una Single Page Application?

Es un tipo de aplicación web donde todas las pantallas las muestra en la misma página, sin recargar el navegador.

Técnicamente, una SPA es un sitio donde existe un único punto de entrada, generalmente el archivo index.html. En la aplicación no hay ningún otro archivo HTML al que se pueda acceder de manera separada y que nos muestre un contenido o parte de la aplicación, toda la acción se produce dentro del mismo index.html.

Aunque solo tengamos una página, lo que sí tenemos en la aplicación son varias vistas, entendiendo por vista algo como lo que sería una pantalla en una aplicación de escritorio. En la misma página, por tanto, se irán intercambiando vistas distintas, produciendo el efecto de que tienes varias páginas, cuando realmente todo es la misma, intercambiando vistas.

Se realizan utilizando el lenguaje JavaScript.

Ejemplos: YouTube, Netflix, Gmail

3. ¿Qué ventajas tiene el desarrollo de una SPA + REST sobre una DWP?

Aquí el cliente consiste en un conjunto de archivos estáticos HTML + CSS + Javascript (con un archivo HTML "maestro"), que se comunica con un servidor web usando la arquitectura REST (generalmente sobre el protocolo HTTP). Si bien un servidor web puede implementarse en casi cualquier lenguaje de programación, los desarrolladores generalmente usan frameworks especiales para implementar el SPA en el lado del cliente y una API REST en el lado del servidor. Ejemplos incluyen:

Frameworks de API REST:

- Express Js
- API web ASP.Net
- Nancy Fx
- Jersey

Frameworks de SPA:

- JS angular
- Ember JS

- Ext JS

Algunas ventajas notables de SPA-REST:

API REST libres: Una API REST presenta una interfaz uniforme, flexible, segura (a través de HTTPS) y ampliamente reconocida para sistemas de terceros o sus otros clientes (como una aplicación móvil integrada) que desean interactuar con su aplicación.

Diseño más sencillo: Dado que la API REST es el punto crucial de un sistema y, al mismo tiempo, un lenguaje bastante fácil de entender, es posible centrar el diseño inicial en torno a él. Esto permite implementar el servidor, el cliente y las pruebas automáticas casi de manera simultánea.

Pruebas más sencillas e integración continua: Con SPA + REST puede construir y ejecutar toda su solución de servidor y (llamando a la API REST) desarrollar pruebas automáticas de todas las capas que trabajan juntas, tal como lo hacen en producción (un proceso generalmente conocido como prueba de integración).

Mejor interpretación: Las SPA residen en un solo documento HTML estático, cargado inicialmente, y muestran todo su contenido cambiando ese documento en tiempo real. Esto da como resultado una experiencia de usuario similar a una aplicación incorporada, mucho más fluida, más receptiva y también aumenta el rendimiento, ya que hay menos archivos para que el navegador descargue y no hay gastos generales relacionados con su generación.

Mejor escalado: Dado que una API REST no tiene estado, no hay diferencia si el cliente está hablando con uno o varios servidores web con cada solicitud. Esto le permite escalar su aplicación de forma rápida y sencilla, mediante la implementación de varias instancias de la solución de servidor, una al lado de la otra, detrás de una puerta de enlace común.

Bajo acoplamiento entre cliente y servidor: Con SPA + REST, el cliente y el servidor son entidades separadas, vinculadas con nada excepto la documentación de la API REST. Esto le permite desarrollarlos, probarlos e implementarlos de forma independiente.

Mejor calidad de código: SPA + REST traza una división clara entre los mundos del servidor (es decir, transacciones, entidades, dominios comerciales, tareas en segundo plano, consistencia) y el cliente (es decir, formularios, gráficos, estilos, UX, capacidad de respuesta, claridad). Esto, a su vez, hace que la solución sea limpia.

4. Mencione brevemente algunos frameworks o librerías para SPA.

AngularJS, ReactJS, VueJS, EmberJS, MeteorJS, React