Engineer

Carga diferida

Cargar algo solo cuando se necesita por primera vez

Qué es

La carga diferida consiste en aplazar la obtención de algo hasta el momento en que se necesita, en vez de cargarlo todo al principio. En una página web suele aplicarse a las imágenes y los marcos incrustados situados más abajo, que solo se obtienen cuando el lector se desplaza hacia ellos. La misma idea se aplica al código que solo se ejecuta tras un clic y a los registros relacionados que una aplicación lee de una base de datos.

El estándar HTML la incorpora al marcado. Las imágenes y los iframes admiten un atributo `loading` con dos valores. `eager` , el predeterminado, obtiene el recurso de inmediato. `lazy` se utiliza, en palabras del estándar, «para aplazar la obtención de un recurso hasta que se cumplan ciertas condiciones». El estándar define el mismo atributo también para `audio` y `video` . Si falta el valor o está mal escrito, cuenta como `eager` , de modo que una errata deja la página cargándose como siempre.

La condición es que el elemento se acerque al área visible. Cada documento tiene su propio observador de intersección para esta tarea; cuando informa de que un elemento diferido está a punto de entrar en la zona visible, el navegador reanuda su carga. Ese observador mira un poco más allá del área visible para que una imagen pueda llegar mientras el lector aún se desplaza hacia ella, y el estándar deja al navegador la elección del tamaño de ese margen.

Las páginas pueden utilizar directamente el mismo mecanismo. La especificación Intersection Observer del W3C permite a un script solicitar avisos asíncronos cuando un elemento empieza o deja de intersectar el área visible u otro elemento, y menciona la carga diferida personalizada de contenido y datos entre

las prácticas para las que se diseñó. Cubre lo que no cubre el atributo: una imagen de fondo definida en CSS, un mapa que necesita su propio script o una lista que obtiene la página siguiente cuando el lector llega al final.

En el caso del código, la unidad es un módulo. La documentación de Next.js presenta la carga diferida como una forma de reducir el JavaScript necesario para representar una ruta: los componentes de cliente y las bibliotecas quedan fuera del paquete inicial y se obtienen cuando hacen falta. Su ejemplo es un cuadro de diálogo que solo se carga cuando alguien hace clic para abrirlo. `next/dynamic` combina `lazy()` y `Suspense` de React, y una llamada sencilla a `import()` hace lo mismo con una biblioteca.

El término es anterior a todo eso. En el catálogo de patrones de aplicaciones empresariales de Martin Fowler, Lazy Load deja un marcador donde irían los datos relacionados y solo los obtiene si algo los utiliza, de modo que leer un objeto de una base de datos no arrastre una gran red de otros objetos que nadie consulta. Un ORM funciona así cuando obtiene un registro relacionado la primera vez que el código accede a él. Django lo hace de forma predeterminada: un campo que no formaba parte de la consulta original se lee de la base de datos cuando se accede a él por primera vez, algo que su documentación describe sin utilizar ese nombre.

La carga diferida no es almacenamiento en caché, que conserva una copia de algo ya obtenido para reutilizarla. Es lo contrario de la precarga: un `<link rel="preload">` indica al navegador que obtenga un recurso por adelantado porque es muy probable que la página lo necesite.

Utilice el concepto en una decisión

Aplique «¿Qué es la carga diferida?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Interfaces que cargan rápido, funcionan en el teléfono y se comportan igual en los navegadores que utilizan sus clientes.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Dirección del desarrollo integral de mapas GIS — cartografía integral, de la base de datos al navegador

Dirección del desarrollo integral de mapas GIS Póngase en contacto →

Por qué importa

Una página paga por cada byte que solicita, lo vea alguien o no. Un artículo largo con cuarenta fotografías, un pie de página con un mapa incrustado y una página de producto con un vídeo cerca del final gastan ancho de banda, batería y tiempo de conexión en contenido hasta el que algunos lectores nunca se desplazan. Con la carga diferida, esas peticiones solo se hacen para quienes llegan hasta allí, y lo necesario para la primera pantalla no tiene que compartir la conexión con ellas.

El navegador decide cuánta antelación basta. El estándar sugiere que el margen tenga en cuenta la velocidad de desplazamiento del lector y la calidad de la red, y sea mayor en conexiones lentas o con mucha latencia para que las imágenes estén listas al entrar en la zona visible. También exige que el margen no filtre información que permita identificar al lector mediante una huella digital.

También incorpora una regla de privacidad. Los navegadores solo aplican la carga diferida cuando la ejecución de scripts está habilitada. De lo contrario, explica el estándar, un sitio podría repartir imágenes por su marcado y deducir aproximadamente hasta dónde se desplaza cada lector a partir de qué imágenes se solicitan al servidor y cuándo. Con los scripts deshabilitados, `loading="lazy"` se convierte sin avisar en una carga normal.

Para el código, Next.js expresa la ventaja en términos de la carga inicial: hace falta menos JavaScript antes de poder representar una ruta. Una biblioteca de gráficos utilizada en una sola pestaña o un editor de texto enriquecido detrás de un botón Editar no tienen por qué ralentizar todas las demás páginas.

En un ORM, la ventaja está en lo que nunca se lee. Fowler señala que cargar un objeto junto con todo lo relacionado resulta cómodo hasta que implica cargar una enorme cantidad de objetos cuando solo se quieren unos pocos. Lazy Load rompe esa cadena.

Cómo se aplica en la práctica

Para imágenes y marcos, el atributo hace el trabajo. La imagen y el mapa siguientes están muy abajo en una página, así que ambos pueden esperar:

```
<img
  src="/photos/harbour-800.jpg"
  srcset="/photos/harbour-800.jpg 800w, /photos/harbour-1600.jpg 1600w"
  sizes="auto, 100vw"
  width="1600"
  height="900"
  alt="Fishing boats in the harbour at dawn"
  loading="lazy"
/>

<iframe
  src="/embeds/route-map.html"
  title="Map of the walking route"
  width="640"
  height="360"
  loading="lazy"
></iframe>
```

`width` y `height` dan a la imagen sus proporciones antes de que haya llegado parte alguna de ella: las reglas de representación de HTML convierten los dos números en una relación de aspecto, de modo que el navegador puede reservar un recuadro con la forma correcta. `sizes="auto"` solo se permite en una imagen de carga diferida. Indica al navegador que elija entre los recursos de `srcset` según la anchura con la que se ha dispuesto realmente la imagen, que una imagen diferida ya tiene cuando se obtiene. El `100vw` posterior es la alternativa cuando no se puede determinar esa anchura.

Algunas cargas diferidas necesitan un script. Aquí se obtiene una biblioteca de mapas solo cuando su contenedor está a punto de entrar en la zona visible durante el desplazamiento:

```
const box = document.querySelector("#route-map");

const observer = new IntersectionObserver(
  async (entries) => {
    if (!entries.some((entry) => entry.isIntersecting)) return;
    observer.disconnect();
    const { drawMap } = await import("./route-map.js");
    drawMap(box);
  },
  { rootMargin: "200px 0px" },
);

observer.observe(box);
```

`rootMargin` amplía el área visible en 200 píxeles por arriba y por abajo para esta comprobación, de modo que el módulo empieza a cargarse poco antes de que el recuadro sea visible. La función de devolución de llamada se ejecuta de forma asíncrona y puede comunicar varias entradas a la vez; comprueba `isIntersecting` porque el observador también informa de elementos que están fuera del área. Desconectarlo antes hace que la función realice su trabajo una sola vez, por muchas veces que el lector pase por allí. Hasta entonces, `route-map.js` no forma parte de la página en absoluto.

Las fuentes se pueden dividir para que un navegador obtenga solo las partes que utiliza una página, sin scripts. Una nota de este sitio sobre un fallo de subrayado en Safari documenta cómo se servía entonces la tipografía del sitio, Ubuntu, en seis archivos: tres grosores para texto latino y tres para cirílico, cada uno con el intervalo de caracteres que cubre. El navegador utiliza las variantes latinas para el texto latino y las cirílicas para el texto cirílico. Por eso, una etiqueta de idioma escrita en cirílico puede necesitar una variante cirílica incluso en una página cuyo resto esté en inglés o danés. La nota no lo llama carga diferida, pero esa misma selección según la necesidad evita que la página solicite una variante no utilizada, salvo que se precargue explícitamente.

Los archivos que toda página necesita de inmediato requieren la decisión contraria. Un logro sobre la reducción del paquete de estilos explica por qué se retiró una fuente incrustada como base64 en la hoja de estilos. Se había incluido así para acelerar el primer pintado, pero un archivo separado con precarga aporta la misma ventaja, mientras que la copia incrustada añadía su peso a cada página, hiciera falta o no. En aquella etapa se precargaban tres grosores de la fuente, y el logro señala que fueron los datos de campo, no el afán de incluirlos todos, los que llevaron a añadir el tercero a esa lista. La carga diferida y la precarga son dos caras de una misma decisión: lo que necesita la primera pantalla llega pronto y lo demás espera su turno.

Dónde surgen los problemas

La primera imagen llega la última. Una imagen diferida espera a algo más que a la red. Su obtención solo se reanuda cuando el observador del navegador informa de que está cerca del área visible, y los observadores se actualizan al representar un fotograma, después de calcular los estilos y la disposición. Para una fotografía muy abajo en la página, esa espera no cuesta nada. Para la imagen grande de la parte superior, retrasa esa fotografía que ve todo visitante. Omita `loading` o asígnele `eager` en todo lo que pueda aparecer en la primera pantalla de un teléfono o un ordenador, y reserve `lazy` para lo que viene después.

La página salta al llegar las imágenes. Una imagen sin `width` ni `height` no tiene una forma conocida hasta que llega su archivo y entonces empuja hacia abajo todo lo que tiene debajo. Con carga diferida ocurre mientras el lector se desplaza, a menudo en mitad de una frase. Indique las dimensiones de cada imagen en el marcado o en CSS. Para `sizes="auto"`, el estándar HTML es explícito: sin dimensiones declaradas, es probable que una imagen así se represente a 300 por 150 píxeles, por lo que recomienda encarecidamente indicarlas.

Imágenes que necesitan un script para existir. Una versión hecha a mano coloca la dirección real en un atributo `data-src` y deja que un script la copie a `src` cuando la imagen se acerca. Si ese script no se carga, lanza un error o se impide su ejecución, la imagen nunca aparece, y cualquier herramienta que lea el marcado sin ejecutar scripts ve una imagen sin origen. Con el atributo incorporado ocurre lo contrario: con los scripts deshabilitados, el estándar hace que el navegador cargue la imagen de inmediato. Utilice `loading="lazy"` para imágenes y marcos y reserve los scripts para lo que el atributo no cubre.

Aplazar lo que necesita cada visita. La carga diferida compensa cuando gran parte de lo aplazado nunca se utiliza. Cuando se va a utilizar todo, solo divide una obtención en muchas. Un bucle que lee un registro relacionado por cada elemento de una lista envía una consulta independiente para cada uno: la documentación de Django cuenta una consulta para la lista más una por elemento en su modo predeterminado y lo llama el problema de las consultas N+1. Una página cuyos componentes solicitan cada uno sus propios datos al aparecer hace lo mismo por la red. Cargue lo que con seguridad se necesitará junto con la lista, en una consulta o una petición, y reserve la carga diferida para lo que probablemente no se llegue a utilizar.

Lecturas adicionales

- Estándar HTML: atributos de carga diferida — las palabras clave `lazy` y `eager`, el observador de intersección del propio navegador y la regla que desactiva la carga diferida sin scripts.
- Intersection Observer — el borrador de trabajo del W3C en que se basa el observador: raíces, márgenes, umbrales y cómo se ponen en cola las funciones de devolución de llamada.
- Cómo cargar de forma diferida componentes de cliente y bibliotecas — `next/dynamic`, `React.lazy()` con `Suspense`, la omisión de la representación en el servidor y la carga de una biblioteca mediante `import()`.

- Carga diferida — el patrón de Martin Fowler en Patterns of Enterprise Application Architecture, con sus cuatro variantes.
- Modos de obtención — cómo obtiene Django los campos que no se cargaron con la consulta original y los modos que agrupan o bloquean esas lecturas.

Glosario

Elija según sus restricciones reales

Describa el flujo de trabajo, las interfaces existentes y los usuarios que necesitan el cambio. Podemos definir una entrega comprobable y un traspaso que permita a su equipo mantener lo entregado.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es la carga diferida?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/lazy-loading/