Engineer

Chargement différé

Chargement au moment où un élément devient nécessaire

Ce que c’est

Le chargement différé consiste à reporter la récupération d’un élément au moment où il devient nécessaire, plutôt qu’à tout charger d’emblée. Sur une page web, il concerne généralement les images et les cadres intégrés situés plus bas, récupérés seulement lorsque la personne qui lit fait défiler la page vers eux. La même idée s’applique au code exécuté après un clic et aux enregistrements associés qu’une application lit dans une base de données.

Le standard HTML l’intègre au balisage. Les images et les iframes acceptent un attribut `loading` à deux valeurs. `eager`, la valeur retenue sans indication valide, récupère immédiatement la ressource. `lazy` est, selon les mots du standard, « utilisé pour différer la récupération d’une ressource jusqu’à ce que certaines conditions soient remplies ». Le standard définit le même attribut sur `audio` et `video`. Une valeur absente ou mal orthographiée vaut `eager` : une faute de frappe laisse donc la page se charger comme auparavant.

La condition est que l’élément approche de la zone visible. Chaque document possède son propre observateur d’intersection à cette fin ; lorsqu’il signale qu’un élément différé s’apprête à devenir visible, le navigateur reprend son chargement. L’observateur regarde un peu au-delà de la zone visible pour que l’image puisse arriver pendant que la personne fait encore défiler la page vers elle, et le standard laisse au navigateur le choix de cette marge.

Les pages peuvent utiliser directement le même mécanisme. La spécification Intersection Observer du W3C permet à un script de demander une notification asynchrone lorsqu’un élément commence ou cesse d’intersecter la zone visible ou un autre élément. Elle cite le chargement différé personnalisé de contenu et de données parmi les pratiques qu’elle vise à prendre en charge. Elle couvre ce

que l'attribut ne gère pas : une image d'arrière-plan définie en CSS, une carte nécessitant son propre script, une liste qui récupère sa page suivante lorsque la personne atteint la fin.

Pour le code, l'unité est le module. La documentation de Next.js présente le chargement différé comme un moyen de réduire le JavaScript nécessaire au rendu d'une route : les composants client et les bibliothèques sont exclus du premier paquet et récupérés au besoin. Son exemple est une boîte de dialogue qui ne se charge que lorsqu'une personne clique pour l'ouvrir. `next/dynamic` combine `lazy()` et `Suspense` de React, et un simple appel à `import()` fait de même pour une bibliothèque.

Le terme est plus ancien que tout cela. Dans le catalogue de modèles d'architecture d'applications d'entreprise de Martin Fowler, Lazy Load laisse un marqueur à la place des données associées et ne les récupère que si quelque chose les utilise : lire un objet dans une base de données n'entraîne donc pas le chargement d'un vaste réseau d'objets que personne ne consulte. Un ORM fonctionne ainsi lorsqu'il récupère un enregistrement associé au premier accès du code. Django le fait sans configuration supplémentaire : un champ absent de la requête initiale est lu dans la base au premier accès, ce que sa documentation décrit sans employer le nom.

Le chargement différé se distingue de la mise en cache, qui conserve une copie d'un élément déjà récupéré pour pouvoir le réutiliser. Il est l'opposé du préchargement : un `<link rel="preload">` indique au navigateur de récupérer une ressource à l'avance parce que la page aura très probablement besoin d'elle.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce que le chargement différé ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Des interfaces rapides, utilisables sur téléphone et cohérentes dans tous les navigateurs de vos clients.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Développement cartographique GIS full-stack — cartographie full-stack, de la base au navigateur

Développement cartographique GIS full-stack Contactez-nous →

Pourquoi c’est important

Une page paie chaque octet demandé, que quelqu’un le voie ou non. Un long article contenant quarante photographies, un pied de page contenant une carte intégrée et une fiche produit avec une vidéo vers le bas consomment de la bande passante, de la batterie et du temps de connexion pour du contenu que certaines personnes n’atteignent jamais. Avec le chargement différé, seules celles qui vont jusque-là déclenchent ces requêtes, et les ressources du premier écran n’ont pas à partager la connexion avec elles.

Le navigateur choisit ce qui est assez tôt. Le standard suggère que la marge tienne compte de la vitesse de défilement et de la qualité du réseau, avec une marge plus large sur les connexions lentes ou à forte latence pour que les images soient en place lorsqu’elles deviennent visibles. Il exige aussi que cette marge ne divulgue pas d’informations exploitables pour identifier la personne par empreinte numérique.

Une règle de confidentialité est également intégrée. Les navigateurs ne diffèrent le chargement que si les scripts sont activés. Sinon, explique le standard, un site pourrait disperser des images dans son balisage et déduire approximativement jusqu’où chaque personne a défilé à partir des images demandées au serveur et du moment des demandes. Sans scripts, `loading="lazy"` devient discrètement un chargement ordinaire.

Pour le code, Next.js exprime le gain en termes de chargement initial : moins de JavaScript nécessaire avant le rendu d’une route. Une bibliothèque de graphiques utilisée dans un seul onglet ou un éditeur de texte enrichi derrière un bouton Modifier n’a pas à ralentir toutes les autres pages.

Dans un ORM, le gain tient à ce qui n’est jamais lu. Fowler souligne que charger un objet avec tout ce qui lui est associé est commode jusqu’à ce que cela entraîne le chargement d’un très grand nombre d’objets pour quelques-uns seulement recherchés ; Lazy Load brise cette chaîne.

À quoi cela ressemble en pratique

Pour les images et les cadres, l’attribut fait le travail. L’image et la carte ci-dessous se trouvent loin dans la page : toutes deux peuvent donc attendre :

```
<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` et `height` donnent à l'image ses proportions avant la réception du moindre fragment : les règles de rendu HTML associent les deux nombres à un rapport largeur-hauteur, pour que le navigateur réserve un rectangle de la bonne forme. `sizes="auto"` n'est autorisé que sur une image à chargement différé. Il indique au navigateur de choisir dans `srcset` selon la largeur réelle de mise en page de l'image, déjà connue au moment de sa récupération différée ; le `100vw` qui suit sert de repli lorsque cette largeur ne peut pas être déterminée.

Certains chargements différés nécessitent un script. Ici, une bibliothèque cartographique n'est récupérée que lorsque son conteneur s'apprête à entrer dans la zone visible pendant le défilement :

```
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);
```

Pour ce contrôle, `rootMargin` agrandit la zone visible de 200 pixels en haut et en bas : le module commence donc à charger peu avant que le rectangle soit visible. La fonction de rappel s'exécute de façon asynchrone et peut signaler plusieurs entrées à la fois. Elle vérifie `isIntersecting` parce que l'observateur signale aussi les éléments situés hors de la zone. Se déconnecter d'abord permet de n'effectuer ce travail qu'une fois, quel que soit le nombre de passages pendant le défilement. Jusque-là, `route-map.js` ne fait pas du tout partie de la page.

Les polices peuvent être divisées pour que le navigateur ne récupère que les parties utilisées par une page, sans aucun script. Une note de ce site sur un défaut de soulignement dans Safari documente la distribution, à cette époque, de la police du site, Ubuntu, en six fichiers : trois graisses pour le texte latin et trois pour le cyrillique, chacun déclarant les caractères couverts. Le navigateur choisit les fichiers latins pour le texte latin et les fichiers cyrilliques pour le texte cyrillique. Un libellé de langue en cyrillique peut donc nécessiter un fichier cyrillique sur une page par ailleurs en anglais ou en danois. La note ne parle pas de chargement différé, mais ce même choix en fonction des besoins évite de demander un fichier de police inutilisé, sauf s'il est explicitement préchargé.

Les fichiers nécessaires immédiatement sur chaque page appellent la décision inverse. Une réalisation sur la réduction du paquet de feuilles de style explique le retrait d'une police intégrée à la feuille de style en base64. Elle y avait été placée pour accélérer le premier affichage, mais un fichier séparé préchargé procure le même avantage, tandis que la copie intégrée alourdissait chaque page, que celle-ci ait besoin de la police ou non. À cette étape, trois graisses de la police étaient préchargées à la place ; la réalisation précise que les données de terrain, plutôt qu'un souci d'exhaustivité, ont ajouté la troisième à cette liste. Chargement différé et préchargement sont les deux faces d'une même décision : ce dont le premier écran a besoin arrive tôt, le reste attend son tour.

Où les problèmes surviennent

La première image arrive en dernier. Une image à chargement différé n'attend pas seulement le réseau. Sa récupération ne reprend que lorsque l'observateur du navigateur la signale près de la zone visible ; les observateurs sont mis à jour lors du rendu d'une nouvelle image à l'écran, après le calcul des styles et de la disposition. Pour une photographie loin dans la page, cette attente ne coûte rien. Pour la grande image du haut, elle retarde l'image que tous les visiteurs voient. Omettez `loading` ou choisissez `eager` pour tout élément susceptible d'apparaître dans le premier écran sur téléphone ou ordinateur, et réservez `lazy` à ce qui suit.

La page saute à l'arrivée des images. Une image sans `width` ni `height` n'a pas de forme connue avant l'arrivée de son fichier, puis pousse tout ce qui suit vers le bas. Avec le chargement différé, cela arrive pendant le défilement, souvent au milieu d'une phrase. Donnez ses dimensions à chaque image dans le balisage ou en CSS. Pour `sizes="auto"`, le standard HTML est explicite : sans dimensions déclarées, une telle image sera probablement dessinée à 300 par 150 pixels ; il encourage donc vivement leur déclaration.

Des images qui ont besoin d'un script pour exister. Une version artisanale met l'adresse réelle dans un attribut `data-src` et laisse un script la copier dans `src` à l'approche de l'image. Si le script ne se charge pas, lève une erreur ou est bloqué, l'image n'apparaît jamais, et tout outil lisant le balisage sans exécuter les scripts voit une image sans source. L'attribut intégré prévoit le repli inverse : sans scripts, le standard fait charger l'image immédiatement par le navigateur. Utilisez `loading="lazy"` pour les images et les cadres, et réservez les scripts à ce que l'attribut ne couvre pas.

Différer ce dont chaque visite a besoin. Le chargement différé est rentable lorsqu'une grande partie de ce qui est reporté n'est jamais utilisée. Si tout sert, il divise seulement une récupération en plusieurs. Une boucle lisant un enregistrement associé pour chaque élément d'une liste envoie une requête distincte par élément : la documentation de Django compte une requête pour la liste, plus une par élément dans le mode standard, et appelle cela le problème des requêtes N+1. Une page dont les composants demandent chacun leurs données à mesure qu'ils apparaissent fait la même chose sur le réseau. Chargez avec la liste ce qui sera certainement nécessaire, en une requête de base de données ou une requête réseau, et réservez le chargement différé aux éléments qui ne seront probablement pas consultés.

Pour aller plus loin

- Standard HTML : attributs de chargement différé — les mots-clés `lazy` et `eager`, l'observateur d'intersection du navigateur et la règle qui désactive le chargement différé sans scripts.
- Intersection Observer — le projet de spécification du W3C qui définit l'observateur : racines, marges, seuils et mise en file des fonctions de rappel.
- Charger à la demande les composants client et les bibliothèques — `next/dynamic`, `React.lazy()` avec `Suspense`, omission du rendu serveur et chargement d'une bibliothèque avec `import()`.
- Lazy Load — le modèle de Martin Fowler dans Patterns of Enterprise Application Architecture, avec ses quatre variantes.

- Modes de récupération — comment Django récupère les champs absents de la requête initiale et les modes qui regroupent ou bloquent ces récupérations.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce que le chargement différé ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

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

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/lazy-loading/