

📌 در Lazy Loading راهنمای کش مرورگر و React و Vue

✓ مقدمه

:دو مفهوم مهم در بهینه‌سازی عملکرد وب هستند. در این داکيومنت توضیح می‌دهیم Lazy Loading کش مرورگر و

- کش مرورگر چیست و چگونه کار می‌کند؟
- باعث کش شدن می‌شود؟ Lazy Loading آیا
- چیست؟ React.js و Vue.js بهترین روش برای استفاده از کامپوننت‌های مشترک در
- تفاوتی دارند؟ Non-Lazy Loading و Lazy Loading آیا

♦ کش مرورگر چگونه کار می‌کند؟

و تصاویر را در حافظه‌ی خود نگه می‌دارد. وقتی یک فایل جاوااسکریپت مانند CSS، مرورگر فایل‌های جاوااسکریپت یک بار دانلود شود، اگر تنظیمات کش درست باشد، در درخواست‌های بعدی از کش `button.js` یا `main.js` خوانده می‌شود.

❗ ندارد! اگر یک کامپوننت یک بار دانلود شود، در دفعات بعدی نیازی به Lazy Loading **نکته:** کش شدن ربطی به `button.js` یا `main.js` ندارد. نشده باشد Lazy Loading دانلود مجدد آن نیست، حتی اگر

✓ و کش کامپوننت‌های مشترک Vue.js

هم در صفحه اصلی و هم در صفحات دیگر استفاده شود، بهتر `Button.vue` اگر یک کامپوننت مانند `Vue.js` در **نشود** زیرا **Lazy Load** است

1. قرار می‌گیرد و مرورگر پس از یک بار دانلود، آن را از کش (`app.js`) کد در فایل اصلی، Lazy Loading بدون می‌خواند.
2. جداگانه دانلود شود که باعث `button.js` فعال شود، هر بار که صفحه جدید باز شود، باید Lazy Loading اگر تأخیر می‌شود.

✓ نکنیم تا از کش مرورگر Lazy Load را Button.vue بهتر است کامپوننت‌های پرتکرار مانند Vue.js پس در استفاده شود.

♦ کش کامپوننت‌های مشترک React.js

باندل می‌کند. حالا سوال main.js تمام کدهای مربوط به برنامه را در یک فایل مثل Vite یا Webpack در React.js اینجاست:

دوباره main.js باشد و در صفحه‌های مختلف استفاده شود، آیا باید کل main.js در (Button.js) اگر دکمه ؟ پردازش شود؟

✓ یک بار دانلود و پردازش شد main.js خیر! وقتی

1. کل فایل در کش ذخیره می‌شود.
2. اجرا می‌کند (RAM) هنگام تغییر صفحه، مرورگر فقط بخش موردنیاز را از حافظه.
3. نیست، بلکه از کش خوانده می‌شود main.js نیازی به دانلود مجدد کل.

✓ نکنیم، مگر اینکه خیلی بزرگ باشند Lazy Load هم بهتر است کامپوننت‌های پرتکرار را React پس در

🔥 Lazy Loading و Non-Lazy Loading مقایسه

✓ وجود ندارد Lazy Loading و Non-Lazy Loading از نظر کش شدن، هیچ تفاوتی بین

✓ تفاوت اصلی این دو روش در "زمان دانلود" و "زمان اجرا" است

📌 Button.js تفاوت‌ها در استفاده از

کش شدن	تأخیر در نمایش	زمان دانلود	حالت
✓ کش می‌شود	ندارد، سریع اجرا می‌شود ✗	دانلود main.js همراه می‌شود	Lazy Loading بدون
✓ کش می‌شود	ممکن است تأخیر ایجاد شود ⌚	وقتی لازم باشد دانلود می‌شود	Lazy Loading با

📌 کنیم یا نه، در نهایت یک بار دانلود و کش می‌شود! تفاوت در این است که Lazy Load را Button.js پس اگر اگر شود، ممکن است کمی تأخیر در نمایش داشته باشد Lazy Load اگر

⚡ مفید است؟ Lazy Loading چه زمانی

- ✓ (Quill.js مثل یک ویرایشگر متن مانند) اگر یک کامپوننت خیلی بزرگ باشد
- ✓ (در صفحه گزارشات Chart.js مثل) اگر فقط در برخی صفحات استفاده شود
- ✓ اگر نیاز به بهینه‌سازی بارگذاری اولیه باشد

✂ چطور بفهمیم یک فایل از کش خوانده شده است؟

Chrome در مرورگر:

1. DevTools را باز کنید (F12 → Network → JS)
2. صفحه را رفرش کنید و ببینید:
 - یعنی فایل از کش خوانده شده است (from disk cache) اگر نوشته باشد
 - باشد، یعنی سرور بررسی کرده ولی فایل تغییر نکرده و از کش استفاده شده (Not Modified) اگر 304 است.

✓ نتیجه‌گیری

- ✓ نکنید، چون مرورگر از کش استفاده می‌کند Lazy Load را Button.js کامپوننت‌های پرتکرار مثل
- ✓ کنید تا حجم فایل‌های اولیه کاهش یابد Lazy Load فقط کامپوننت‌های بزرگ و کم‌استفاده را
- ✓ ارتباطی ندارد؛ اگر یک فایل دانلود شود، در صورت عدم تغییر، از کش خوانده خواهد Lazy Loading کش شدن به شد.
- ✓ فقط در "زمان دانلود" است، نه در کش شدن Non-Lazy Loading و Lazy Loading در نهایت، تفاوت

🚀! امیدوارم این راهنما کمک کرده باشد 🍀