

Performance Web I: otimizando o front-end: Aula 1 - Atividade 11 Análise com DevTools | Alura

2-3 minutes

-
- [Sugerir alteração](#)

As ferramentas de desenvolvimento embutidas nos navegadores modernos são muito úteis para um especialista em performance front-end. Vamos usar agora a inspeção de Rede do seu navegador para medir o impacto da nossa otimização de minificação que fizemos antes.

Vou usar o Chrome nos exemplos, mas outros navegadores possuem funcionalidade parecida.

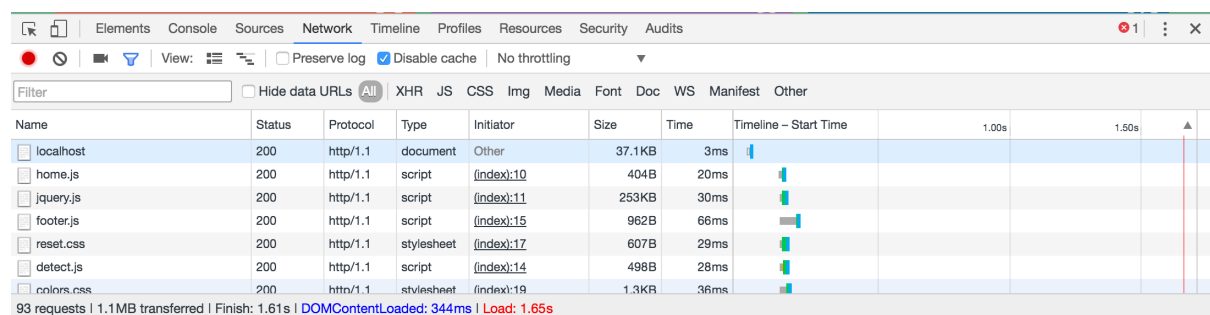
1) Abra duas abas, uma com cada versão do site: <http://localhost:2020/> e <http://localhost:3030/>

2) Em ambas, abra as ferramentas de desenvolvedor com **F12** e selecione a aba **Network**.

3) Marque a opção **Disable cache** para evitar pegar arquivos antigos do cache. **Atualize** as páginas para ele capturar as requisições da página.

4) Observe o painel Network. Se você nunca abriu essa ferramenta antes, explore suas opções. Veja a lista de requests, analise seus tamanhos individuais. Observe a ordem de carregamento, os tempos. Veja no fim o tamanho total da página e o número total de requests.

Meu DevTools do site original não otimizado:



Name	Status	Protocol	Type	Initiator	Size	Time	Timeline - Start Time	1.00s	1.50s
localhost	200	http/1.1	document	Other	37.1KB	3ms			
home.js	200	http/1.1	script	(index):10	404B	20ms			
jquery.js	200	http/1.1	script	(index):11	253KB	30ms			
footer.js	200	http/1.1	script	(index):15	962B	66ms			
reset.css	200	http/1.1	stylesheet	(index):17	607B	29ms			
detect.js	200	http/1.1	script	(index):14	498B	28ms			
colors.css	200	http/1.1	stylesheet	(index):19	1.3KB	36ms			

93 requests | 1.1MB transferred | Finish: 1.61s | DOMContentLoaded: 344ms | Load: 1.65s

E o DevTools da versão já minificada:

The screenshot shows the Chrome DevTools Network tab with the following data:

Name	Status	Protocol	Type	Initiator	Size	Time	Timeline - Start Time	1.00s	1.50s
localhost	200	http/1.1	document	Other	32.9KB	3ms			
home.js	200	http/1.1	script	(index):1	389B	27ms			
jquery.js	200	http/1.1	script	(index):1	84.4KB	30ms			
menu.js	200	http/1.1	script	(index):1	427B	27ms			
busca.js	200	http/1.1	script	(index):1	1.0KB	28ms			
detect.js	200	http/1.1	script	(index):1	455B	28ms			
fontawesome.js	200	http/1.1	script	(index):1	763B	28ms			

93 requests | 949KB transferred | Finish: 1.59s | DOMContentLoaded: 294ms | Load: 1.77s

Que conclusões podemos tirar? O que a minificação melhorou? Quanto?

Opinião do instrutor

-
- [Sugerir alteração](#)

O tamanho final da página caiu consideravelmente. Antes tínhamos 1,1MB e agora 950KB. Uma queda de 14% por apenas rodar um comando simples de minificação!

Repare que outras métricas não são afetadas: o número de requests é o mesmo, por exemplo.

Podemos observar quais requests foram mais impactados pela minificação. Veja o HTML (1º request) que caiu de 37KB para 32KB. O mais gritante é o jQuery que caiu de 253KB para 84KB!