**Compression Dictionary Transport**

Experimental: Esta é uma tecnologia experimental. Verifique a tabela de compatibilidade do navegador com atenção antes de usar em produção.

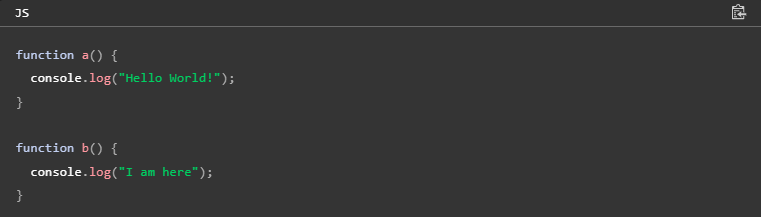
O Transporte de Dicionário de Compressão é uma maneira de usar um dicionário de compressão compartilhado para reduzir drasticamente o tamanho do transporte de respostas HTTP.

**Overview**

Algoritmos de compressão são usados ​​em HTTP para reduzir o tamanho dos recursos baixados pela rede, reduzindo o custo da largura de banda e o tempo de carregamento das páginas. Algoritmos de compressão HTTP sem perdas funcionam encontrando redundância na fonte: por exemplo, locais onde um texto como a string "função" é repetido. Eles então incluem apenas uma cópia da string redundante e substituem as ocorrências dela no recurso por referências a essa cópia. Como as referências são mais curtas que a string, a versão compactada é mais curta.

Observação: uma tentativa anterior dessa tecnologia foi chamada de SDCH (Shared Dictionary Compression for HTTP), mas nunca recebeu amplo suporte e foi removida em 2017. O Compression Dictionary Transport é uma implementação mais bem especificada e mais robusta, com consenso mais amplo do setor.

Por exemplo, veja este JavaScript:



Isso poderia ser compactado substituindo sequências de caracteres repetidas por referências a um local anterior e número de caracteres, assim:



Neste exemplo, [0:9] refere-se à cópia dos 9 caracteres começando no caractere 0. Observe que este é um exemplo simplificado para demonstrar o conceito e os algoritmos reais são mais complexos do que isso.

Os clientes podem então reverter a compactação após o download para recriar o recurso original descompactado.

**Compression dictionaries**

Algoritmos como a compressão Brotli e a compressão Zstandard alcançam eficiência ainda maior ao permitir o uso de dicionários de strings comumente encontradas, dispensando a necessidade de cópias delas no recurso compactado. Esses algoritmos vêm com um dicionário padrão predefinido que é usado na compactação de respostas HTTP.

O Transporte de Dicionário de Compressão se baseia nisso, permitindo que você forneça seu próprio dicionário, especialmente aplicável a um conjunto específico de recursos. O algoritmo de compressão pode então referenciá-lo como uma fonte de bytes ao compactar e descompactar o recurso.

Supondo que as referências do exemplo anterior estejam incluídas naquele dicionário comum, isso poderia ser ainda mais reduzido a isto:



O dicionário pode ser um recurso separado, necessário apenas para o Transporte de Dicionário de Compressão, ou pode ser um recurso de que o site precisa de qualquer maneira.

Por exemplo, suponha que seu site use uma biblioteca JavaScript. Normalmente, você carregaria uma versão específica da biblioteca e poderia incluir o nome da versão no nome da biblioteca, como <script src="my-library.v1.js">. Quando o navegador carregar sua página, ele buscará uma cópia da biblioteca como um sub-recurso.

Se você atualizar a biblioteca para a versão 2, a maior parte do código provavelmente permanecerá inalterada. Assim, os sites podem reduzir significativamente o tamanho do download de my-library.v2.js informando ao navegador para usar my-library.v1.js como um dicionário de compactação para my-library.v2.js. Assim, todas as strings comuns entre as versões 1 e 2 não precisam ser incluídas no download da versão 2, pois o navegador já as possui. A maior parte do tamanho do download de my-library.v2.js é, portanto, apenas o delta entre as duas versões.

O Transporte de Dicionário de Compressão pode atingir uma ordem de magnitude maior de compressão do que a compressão usando um dicionário padrão integrado: veja Exemplos de Transporte de Dicionário de Compressão para alguns resultados da vida real.

Dictionary format

Um dicionário de compressão não segue nenhum formato específico nem possui um tipo MIME específico. São arquivos comuns que podem ser usados ​​na compressão de outros arquivos com conteúdo semelhante.

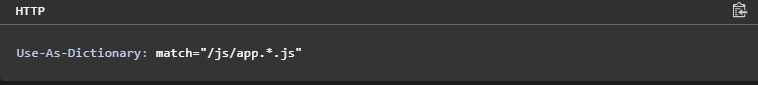
Versões anteriores de arquivos geralmente têm muito conteúdo semelhante, e é por isso que se tornam excelentes dicionários. Usar uma versão anterior de um arquivo como dicionário permite que o algoritmo de compressão referencie com eficiência todo o conteúdo inalterado e capture apenas as diferenças relativamente pequenas na nova versão. Essa abordagem é chamada de compressão delta.

Outra abordagem é listar strings comuns (por exemplo, seus modelos HTML) em um novo arquivo dictionary.txt para que ele possa ser usado para compactar páginas HTML no site. Você pode otimizar isso ainda mais usando ferramentas especializadas, como o gerador de dicionários Brotli, que reduz os dicionários ao tamanho mínimo com sobreposição mínima.

Dicionários também podem ser usados ​​para compactar formatos binários com eficiência. Por exemplo, arquivos binários WASM são recursos grandes que também podem se beneficiar da compactação delta.

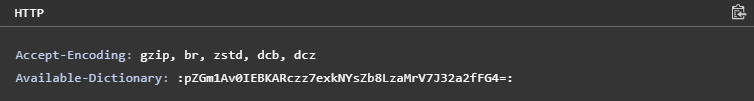
**Recurso existente como um dicionário**

Para usar um recurso como um dicionário, o servidor deve incluir o cabeçalho Use-As-Dictionary na resposta que fornece o recurso:

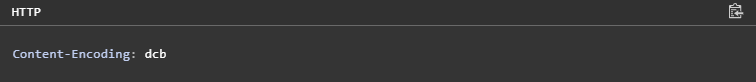


O valor deste cabeçalho indica os recursos que podem usar este recurso como um dicionário: neste caso, isso inclui quaisquer recursos cujos URLs correspondem ao padrão fornecido.

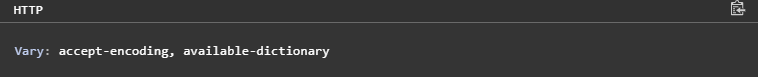
Quando um recurso que corresponde ao padrão fornecido for solicitado posteriormente (por exemplo, app.v2.js), a solicitação incluirá um hash SHA-256 do dicionário disponível no cabeçalho Available-Dictionary, juntamente com valores dcb e/ou dcz no cabeçalho Accept-Encoding (para compactação delta usando Brotli ou ZStandard, conforme apropriado):



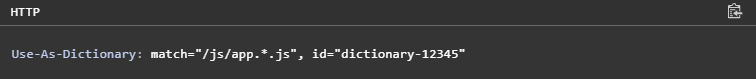
O servidor pode então responder com uma resposta codificada apropriadamente com a codificação de conteúdo escolhida fornecida no cabeçalho Content-Encoding:



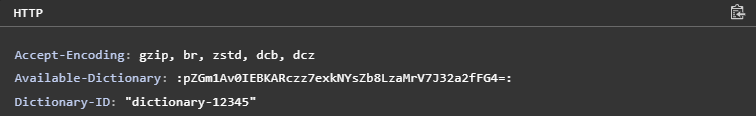
Se a resposta puder ser armazenada em cache, ela deverá incluir um cabeçalho Vary para evitar que os caches sirvam recursos compactados por dicionário para clientes que não os suportam ou sirvam a resposta compactada com o dicionário errado:



Um id opcional também pode ser fornecido no cabeçalho Use-As-Dictionary, para permitir que o servidor encontre o arquivo do dicionário mais facilmente, caso ele não armazene o dicionário pelo hash:



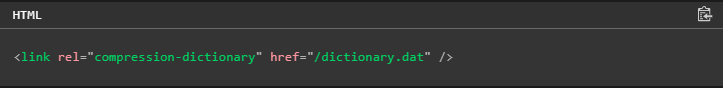
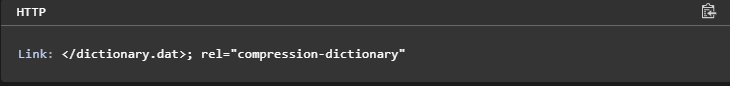
Se isso for fornecido, o valor será enviado em solicitações futuras no cabeçalho Dictionary-ID:



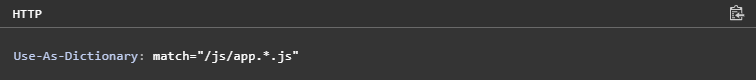
O servidor ainda deve verificar o hash do cabeçalho Available-Dictionary — o Dictionary-ID é uma informação adicional para o servidor identificar o dicionário, mas não substitui a necessidade do cabeçalho Available-Dictionary.

**Dicionário separado**

Um documento HTML também pode fornecer ao navegador um dicionário de compressão, que não é um recurso que o navegador esteja baixando de qualquer forma por meio de um elemento como uma tag <script>. Há dois métodos para fazer isso:

* Inclua um elemento <link> cujo atributo rel esteja definido como compression-dictionary:
* Faça referência ao dicionário usando o cabeçalho Link: 

Este dicionário é então baixado pelo navegador durante o tempo ocioso, e essa resposta deve incluir o cabeçalho Use-As-Dictionary:



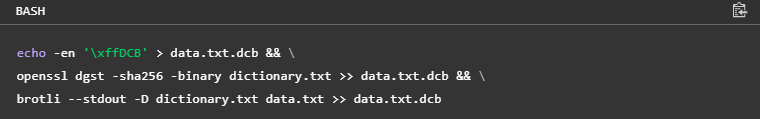
A partir daqui, o processo é semelhante ao exemplo anterior, quando um recurso correspondente é solicitado.

**Creating dictionary-compressed responses**

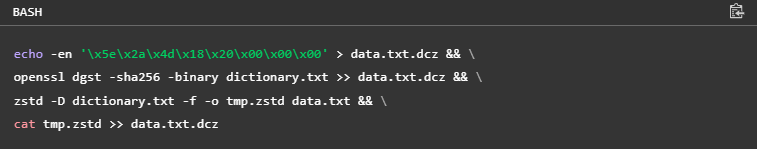
As respostas compactadas por dicionário podem usar os algoritmos Brotli ou ZStandard, com dois requisitos extras: elas também devem incluir um cabeçalho mágico e um hash de dicionário incorporado.

Recursos compactados por dicionário podem ser criados dinamicamente, mas para recursos estáticos, pode ser melhor criá-los antecipadamente, no momento da compilação. Ao usar versões anteriores como dicionários, será necessário decidir quantas versões compactadas por delta serão criadas — apenas para a última versão ou para as últimas X versões para algum valor de X.

Dado um arquivo de dicionário chamado dictionary.text e um arquivo para compactar chamado data.text, o seguinte comando Bash compactará o arquivo usando Brotli, produzindo um arquivo compactado chamado data.txt.dcb:



Dados os mesmos arquivos de entrada, o seguinte comando Bash compactará o arquivo usando o ZStandard, produzindo um arquivo compactado chamado data.txt.dcz:



Observe que você precisará do OpenSSL instalado localmente, bem como do Brotli ou do ZStandard.

**Restrições**

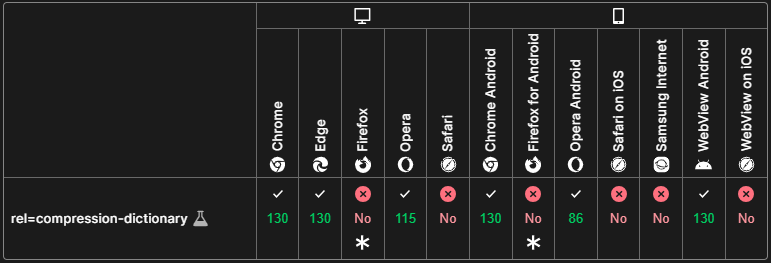
Algoritmos de compressão correm risco de ataques de segurança, portanto, há uma série de restrições para o Transporte de Dicionário de Compressão, incluindo:

* Os dicionários devem ter a mesma origem que o recurso que está sendo utilizado.
* Os recursos compactados por dicionário devem ter a mesma origem que a origem do documento ou seguir as regras do CORS e, portanto, ser solicitados com o atributo crossorigin e servidos com um cabeçalho Access-Control-Allow-Origin apropriado.
* Os dicionários são vinculados ao particionamento de cache HTTP usual e, portanto, não podem ser compartilhados entre origens, mesmo que baixem os mesmos recursos. O dicionário precisará ser baixado novamente para cada origem.

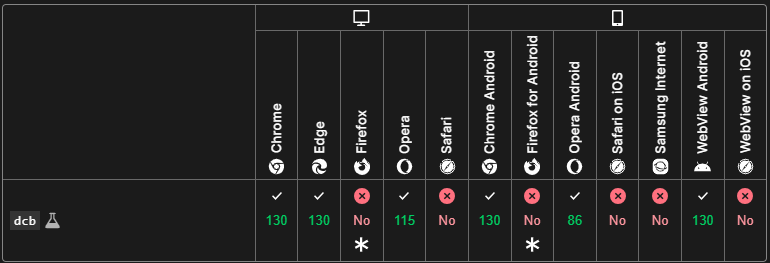
Além disso, os próprios dicionários podem se tornar vetores de rastreamento, de modo que os navegadores podem restringir esse recurso quando os cookies estão desabilitados ou quando outras proteções extras de privacidade estão habilitadas.

**Browser compatibility**

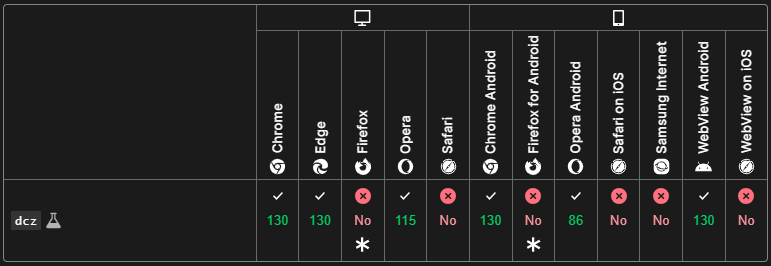
html.elements.link.rel.compression-dicionário



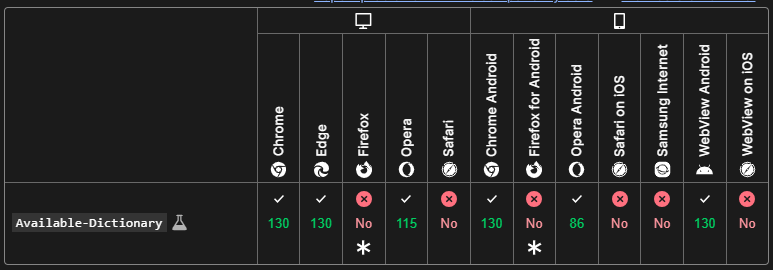
http.headers.Accept-Encoding.dcb



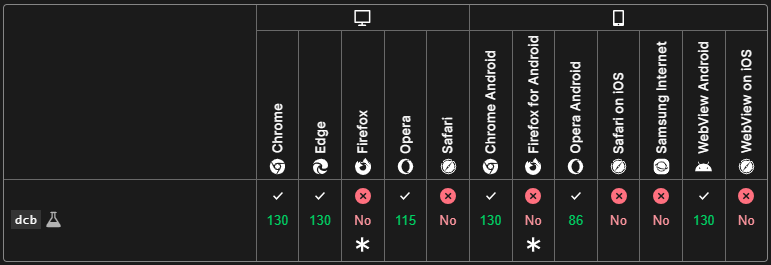
http.headers.Accept-Encoding.dcz



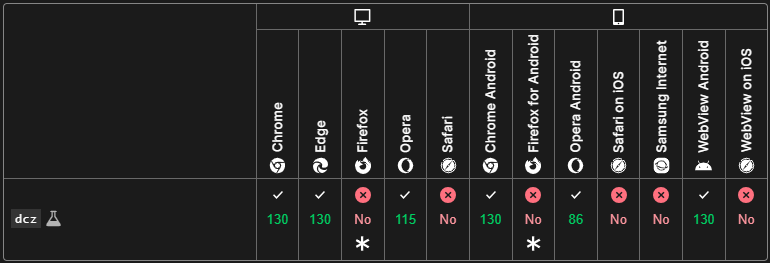
http.headers.Available-Dictionary



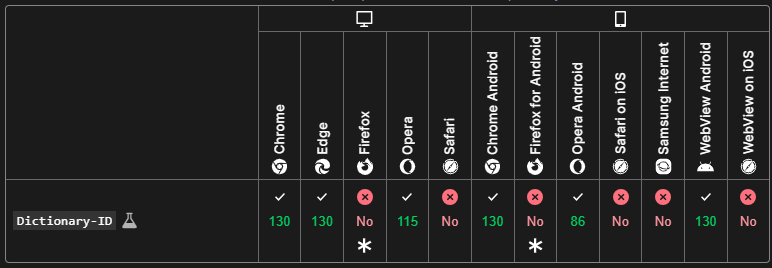
http.headers.Content-Encoding.dcb



http.headers.Content-Encoding.dcz



http.headers.Dictionary-ID



http.headers.Use-As-Dictionary

