



Entendendo e Documentando RESTful APIs

Cache no Cliente

Caching

“Os objetivos do caching HTTP são **eliminar o envio de requisições o máximo possível**, e caso uma requisição precise ser feita, **reduzir os dados de resposta.**”

Caching

“O primeiro objetivo pode ser alcançado usando-se um mecanismo de expiração conhecido como **Cache-Control**, e o segundo é através do mecanismo de validação **ETag** ou **Last-Modified**.”

Caching

Mínimo de requisições: **Cache-Control**

Mínimo de dados nas respostas: **ETag**

Prevenindo Requisições Inteiras

Caching

“A maneira mais rápida de fazer uma requisição HTTP é não enviá-la inteiramente.”

Caching

“O header **Cache-Control** pode ser usado para definir uma política de cache para um recurso.”

Caching

Exemplos

Cache-Control: max-age=3600

Cache-Control: no-cache

Cache-Control: private, max-age=86400

Caching

max-age: Especifica em segundos quanto tempo o recurso pode ser cacheado. É interessante notar que esse cache também pode ser feito por intermediários, não só o browser em si.

Caching

private/public: Define quem pode fazer o cache. **Public** significa que qualquer um pode fazer o cache. **Private** por sua vez indica que o cache só pode ser feito pelo browser, ou seja, os intermediários como os CDNs não podem fazer cache.

Caching

no-cache/no-store: Essas duas diretivas se confundem mas, a **no-store** informa a resposta não deve ser armazenada seja no browser ou em seus intermediários, já o **no-cache** significa que a resposta pode ser cacheada mas não pode ser reusada sem antes checar o servidor. Ela pode ser combinada com um **ETag** a que veremos a seguir.

Caching

Existem outras diretivas que podem ser usadas, as mesmas podem ser encontradas em:

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control>

Caching

Exemplo

```
curl -I http://g1.globo.com/
```

```
curl -I http://www.uol.com.br/
```

```
curl -I http://www.submarino.com.br/
```