Conceitos

Conceitos

Stateless = Não guarda informações, não existe relações entre conexões e sessões: Cookies e storages que serve para guardar informações para próxima conexões.

Extensível

Proxies

São representantes e ficam entre o cliente e o servidor e ajudam a fazer o transporte de dados

Algumas funções são: Cache, filtro (tipo um antivírus), load balancing (distribuição da carga), autenticação e autorização.

URL

Resource

Conceito = Indentificador de recursos, nome ou localização

Resource = Alvo do pedido que pode ser qualquer coisa indentificavel, chamado também de entidade que pode ser: um digital como um email, abstrata como uma sessão ou autenticação, física como produtos e usuários. Se podemos identificar, manipular, nomear etc. identifica um recurso.

URL

Locator ou URL (uniform resource locator) = ela tem componentes obrigatórios que são o protocolo e o domínio, e existem os componentes opcionais que são: Subdomínio, Path, Parâmetros, porta e âncora

URN

URN (uniform resource name) = procura pelo nome

Messages

Mensagens

Serve para request e responses, existe o http/1.1 que é uma forma antiga, porém é legível e é feita de texto e exite a http/2 que é feita em binário com otimizações, mas ainda possui a mesma estrutura do 1.1

Request

O request é um pedido com método, versão de protocolo e URI, com o envio de header e as vezes de body.

Response

Vamos receber a versão do protocolo, status code, header e status message.

Methods

Methods

Métodos são ações que o cliente deseja operar. Podem ser chamados de verbos HTTP e cada um possui a sua semântica. Algumas de suas características são seguros: Não altera o estado do servidor que são o GET, HEAD e OPTIONS, os idempotentes: Ao executar o método a resposta será sempre a mesma, mas o estado poderá ser diferente que são todos os seguros, PUT e DELETE. Os status code poderá ser diferente.

Options

Informação sobre disponibilidade da requisição. O request e o response não tem, e não tem uso em formulários e não guarda valores.

GET

Serve para pegar um recurso e somente recebe dados. Não há request, mas à um response, armazena valores cacheable e tem uso em formulários HTML.

HEAD

Se assemelha ao GET, porém recebemos somente o cabeçalho HEAD/posts. Não tem request e nem response, não possui uso em formulários, mas tem cacheable

POST

Serve para publicar/cadastrar um recurso. Não é seguro e nem idempotente, request e reponse heverá corpo, tem uso em formulários HTML e poderá ser cacheable.

PUT

Cria ou atualiza um recurso. A diferença entre PUT e POST é que o PUT é idempotente seu maior uso é para atualização. O status da criação é 201 e o da atualização é 204 sem resposta ou 200 com resposta. Tem request, mas não tem response, não pode ser usado em formulário HTML e não faz cacheable.

PATCH

Modificação parcial de um recurso. Não é seguro e nem idempotente, tem request e reponse, não tem uso em formulário HTML e não faz cacheable.

DELETE

Serve para remover um recurso. Seu status code será 202 para aceito e não processado, 204 quando não tem resposta de corpo e 200 quando foi apagado e teve um corpo de resposta. Não é seguro, mas é idempotente, à possibilidade de request e response, não tem utilidade em formulários HTML e não tem cacheable.

Headers

General

Os headers do general existem no request e no response como request URL, method, status code, remote Address e referrer policy,

Request

Accept são para os valores aceitosm accept-encoding serve para compactar e o modo como será compactado, cookies são para deixar rastros sobre suas informações no próximo request, user-aggent para o tipo de vrowser que vc está utilizando.

Response

Cache-control mostra qual é o tipo de conteúdo e o a vida máxima, content-lenght é uma reposta para onde foi direcionado, contente-type para o tipo de arquivo, set-cookies são os cookies que serão deixados.

Obtendo Ajuda

O DevDocs é um site, mas também à um programa baixado, que serve para entender detalhes dos headers.

Status

Status Code e Mais

100 {

100: Continue

}

200 {

200: Ok [GET, POST]

201: Created [PUT]

204: No Content [DELETE, PUT]

}

300 {

301: Moved Permanently

308: Permanent Redirect

302: Found [para GET]

307: Temporary Redirect

}

400 {

400: Bad Request

401: Unauthorized

403: Forbidden

404: Not Found

405: Method Not Allowed

429: Too Many Requests

}

500 {

500: Internal Server Error

503: Service Unvaiable

}

Para mais informações

Entre no DevDocs e procute por HTTP status para aprender sobre os status.