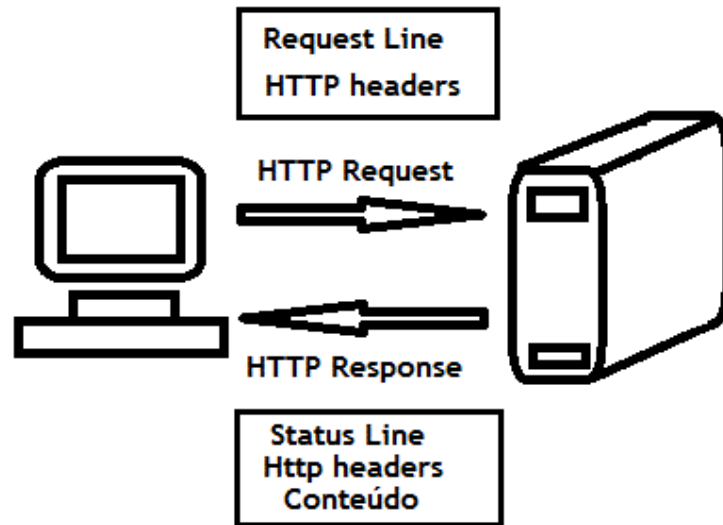


Métodos HTTP

O HTTP define um conjunto de métodos de solicitação para indicar a ação desejada a ser executada para um determinado recurso. Embora também possam ser substantivos, esses métodos de solicitação as vezes são chamados de verbos HTTP. Cada um deles implementa uma semântica diferente, mas alguns recursos comuns são compartilhados por um grupo: por exemplo um método de solicitação pode ser seguro, idempotente ou armazenável em cache.



Métodos HTTP

- Métodos mais comuns: GET e POST
 - GET:
 - POST:
- Outros métodos HTTP – não comumente utilizados:
 - HEAD
 - CONNECT
 - TRACE
 - PUT
 - DELETE
 - OPTIONS

Respostas HTTP

Os códigos de *status* das respostas HTTP indicam se uma requisição HTTP foi corretamente concluída. As respostas são agrupadas em cinco classes:

1. Respostas de informação (100-199),
2. Respostas de sucesso (200-299),
3. Redirecionamentos (300-399)
4. Erros do cliente (400-499)
5. Erros do servidor (500-599).

Frase de Status

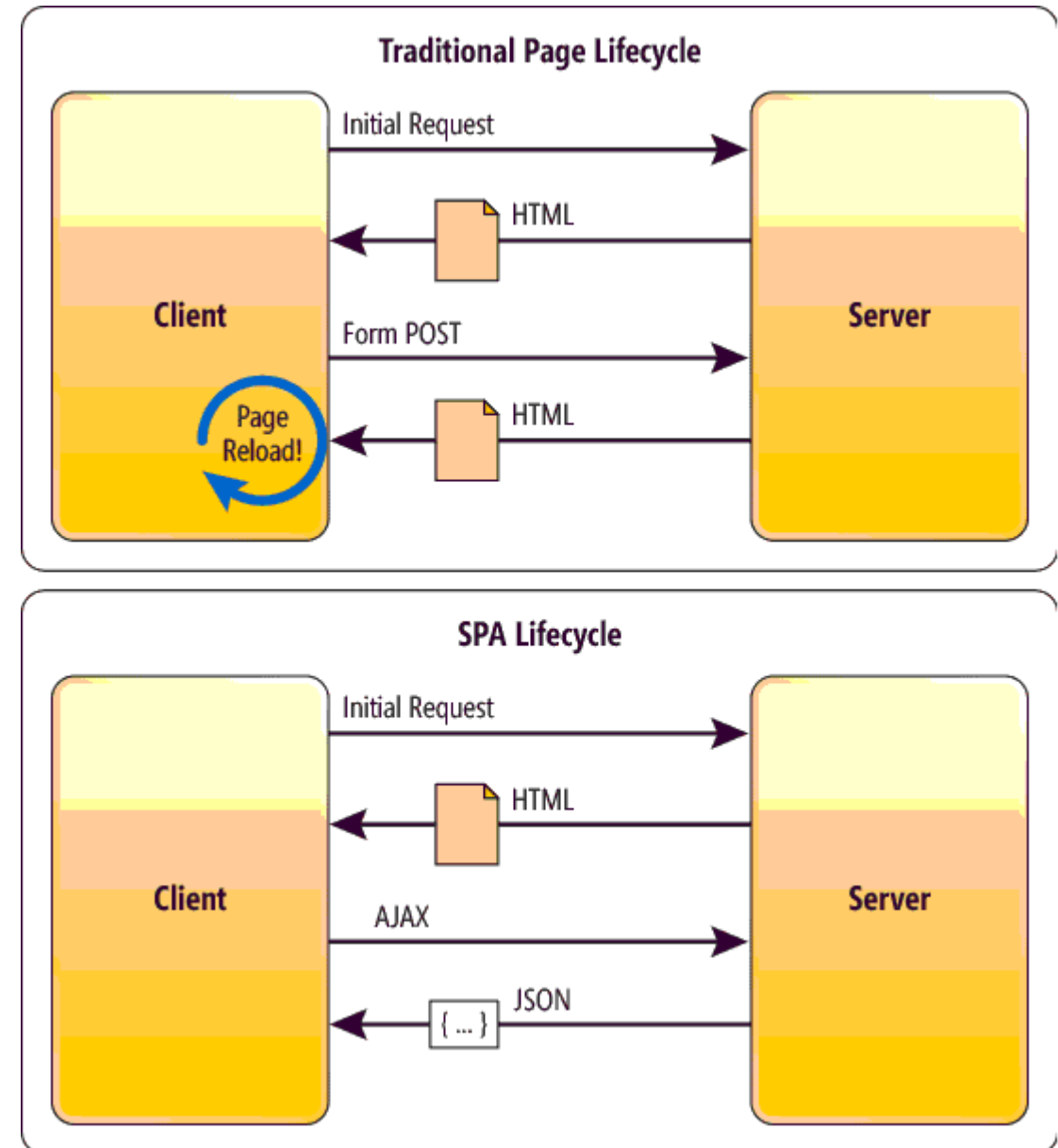


- Esse campo é usado em mensagens de resposta. Ele expande o código de status apresentando informações mais detalhadas na forma de texto.

Código	Frase	Descrição
200	OK	A solicitação foi bem sucedida
400	Bad Request	Erro de sintaxe na solicitação
401	Unauthorized	A solicitação não tem autorização suficiente para ser executada.
403	Forbidden	Serviço negado
404	Not Found	O Documento não foi encontrado
500	Not implemented	Há um erro, como um crash, por exemplo, no servidor
503	Service unavaible	O serviço está temporariamente indisponível mas poderá ser solicitado no futuro

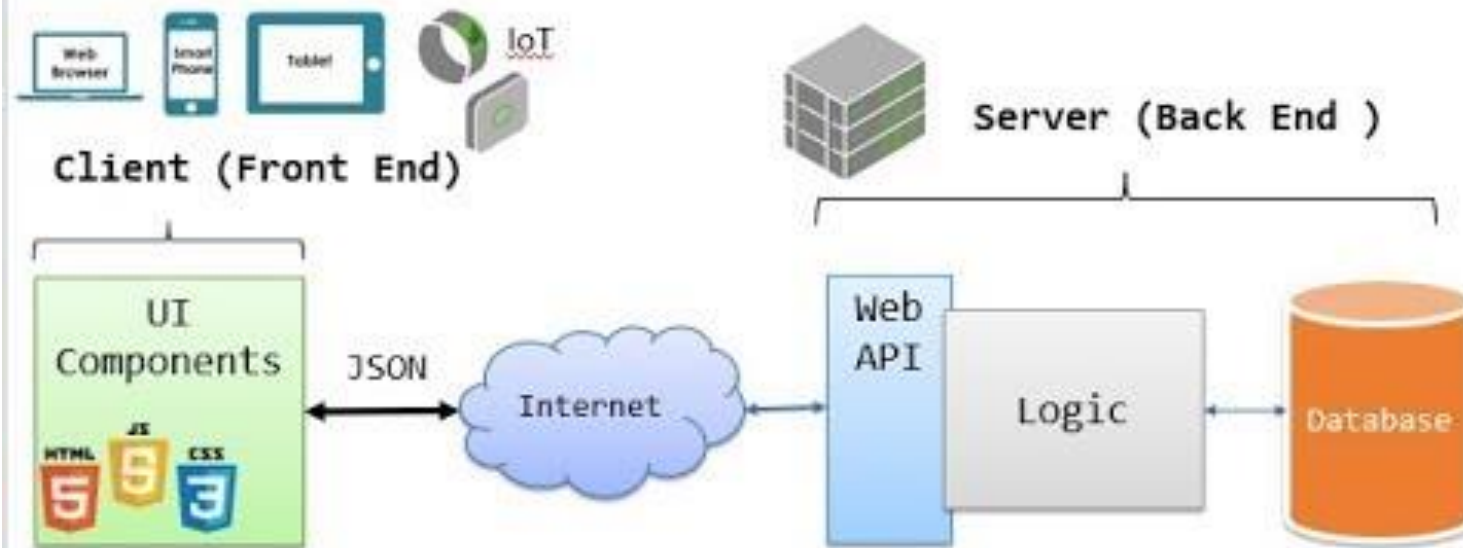
Estruturas de Respostas

Ajax é um **conjunto de técnicas** de desenvolvimento da Web que utiliza muitas tecnologias da Web no lado do cliente para criar aplicativos da Web assíncronos. Com o Ajax, os aplicativos da web podem enviar e **recuperar dados de um servidor de forma assíncrona** sem interferir na exibição e no comportamento da página existente. Ao desacoplar a camada de intercâmbio de dados da camada de apresentação, o Ajax permite que páginas da Web e, por extensão, aplicativos da Web **alterem dinamicamente o conteúdo sem a necessidade de recarregar a página inteira**. Na prática, implementações modernas geralmente utilizam JSON em vez de XML.

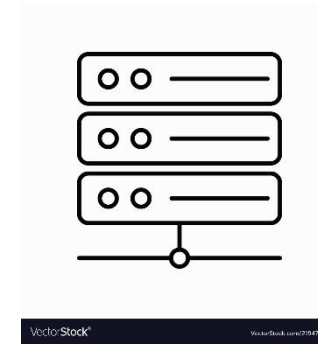


Lógica SPA

Components of Single Page Application (SPA)



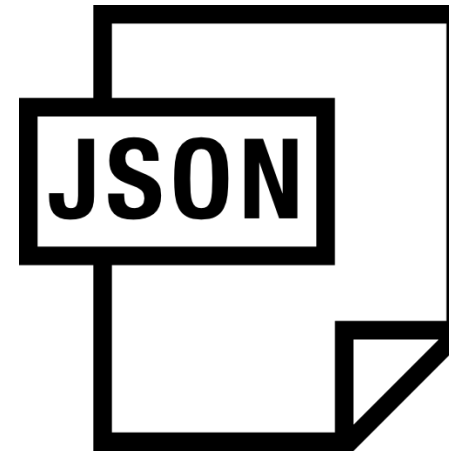
SPA Prática



Resposta HTML: <https://github.com/>



Resposta JSON: <https://api.github.com>



Promise JavaScript

Uma *Promise* é um proxy para um valor não necessariamente conhecido quando a promessa é criada. Ele permite associar manipuladores ao eventual valor de sucesso ou motivo de falha de uma ação assíncrona. Isso permite que métodos assíncronos retornem valores como métodos síncronos: em vez de retornar imediatamente o valor final, o método assíncrono retorna uma promessa de fornecer o valor em algum momento no futuro.

