WEBSOCKETS

Introdução e aplicações

Pedro H. Affonso

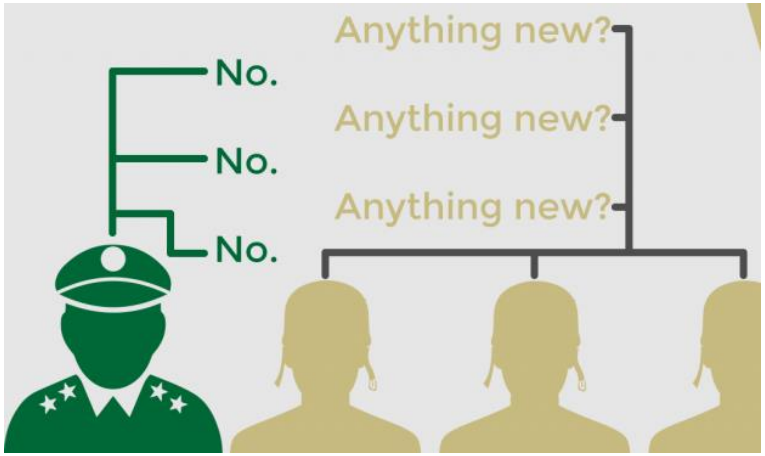
Agenda

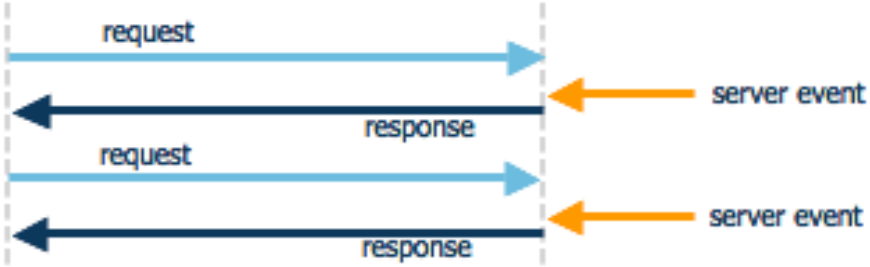
- Motivação
- História e características
- Aplicações
- Filosofia de Projeto
- Handshake
- Data Transfer
- Segurança
- Suporte e implementações
- Exemplo de aplicação

Motivação

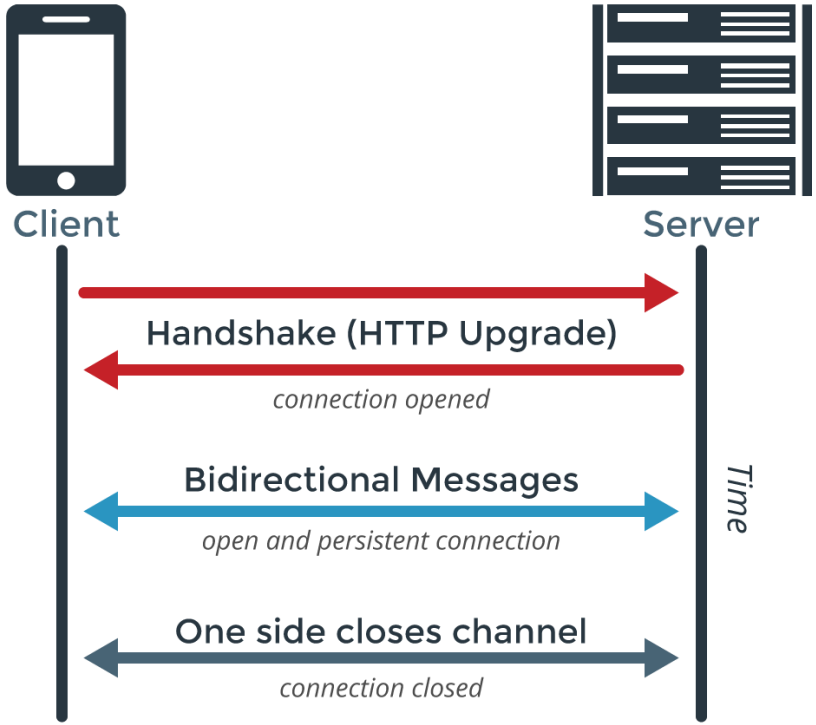
- No início a Web era baseada no modelo requisição/resposta e em páginas estáticas
- ~ 2005: AJAX e uma web mais dinâmica
  - Asynchronous Javascript and XML (XMLHTTPRequest ou XHR)
  - Requisições passam a ser feitas de forma assíncrona, por interação do usuário ou sondagem periódica
- Depois: conexões iniciada pelo servidor
  - Long Polling ou Sondagem Longa
  - Tecnologias Comet ou Push, truques usando Flash, ...
- Problema: overhead do HTTP e latência

AJAX, Long Polling e Websocket





HTTP Long Polling



Websocket

História e Características

- Primeira versão surge em 2008
- Implementado no Chrome em dezembro de 2009
- Padronizado pelo IETF em 2011 – RFC 6455
- API dos browsers está em fase de padronização pelo W3C
- Comunicação Full-duplex
- Streams de mensagens sobre o TCP
- Não compromete a segurança da Web
- Foi feito pensando em Browsers e Servidores Web, mas pode ser usado por qualquer tipo de cliente ou servidor

Aplicações

- Dashboards em tempo real
- Instant messengers e notificações
- Edição multiusuário simultânea
- Jogos online
- MQTT sobre Websockets
- Aplicações que precisem passar por firewalls
- Aplicações em tempo real em geral
- Alguns exemplos:
  - Trello
  - StackOverflow (notificações)
  - socket.io e engine.io

Filosofia de Projeto

- Camada sobre o TCP
- Serviço oferecido é próximo do TCP
- Diferenças
  - Segurança baseada em origem (XSS, CSRF)
  - Endereçamento e subprotocolos para suportar múltiplos serviços em uma mesma porta e múltiplos nomes de host no mesmo ip
  - Mecanismo de quadros e tipo de dados
  - Handshake de despedida extra para contornar problemas com proxies e intermediários
- Pode compartilhar a porta com servidor HTTP
- Semântica de mensagem adicionada para simplificar o uso

Handshake

Cliente:

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
```

Servidor:

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: chat
```

É um protocolo sobre o TCP indepentente do HTTP
Apenas o handshake é interpretado como uma requisição de upgrade HTTP

Data Transfer

- Troca de dados em forma de mensagens
  - Podem ser compostas por vários quadros
  - Quadros possuem tipos: binário, texto, ou controle

Segurança

- Modelo de origem
  - Muito útil para browsers
  - Inútil para clientes dedicados

```
Origin: http://example.com
```

- Falha ao se tentar conectar com um servidor que não seja de Websocket
- Cabeçalhos Sec-* no cabeçalho do cliente
  - Não podem ser acrescentados por atacantes usando apenas HTML e Javascript
  - Impedem conexão usando HTTP com forms ou XHR

```
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
```

Suporte e implementações

- A maioria dos browsers modernos – Chrome, Firefox, Edge, IE, Opera, Safari
- Servidor
  - Node.js – ws, websocket, nodejs-websocket, socket.io, ...
  - Java – Incluído na especificação JavaEE 7- JSR356
  - Ruby – Faye, websocket-ruby, web-socket-ruby
  - PHP – PHPWebSockets, Ratchet
  - Python
  - C++ - LibWebsockets
  - MQTT - Mosquitto

Exemplo – Aplicação de chat

- Cliente – browser
- Servidor – Node.js e biblioteca socket.io

Referências

- https://www.html5rocks.com/pt/tutorials/websockets/basics/
- https://tools.ietf.org/html/rfc6455
- https://en.wikipedia.org/wiki/WebSocket