TINVENTION

WebSockets with Spring

Internal WorkShop - BootCamp

Contents

- TBC

(almost) Real-Time Web

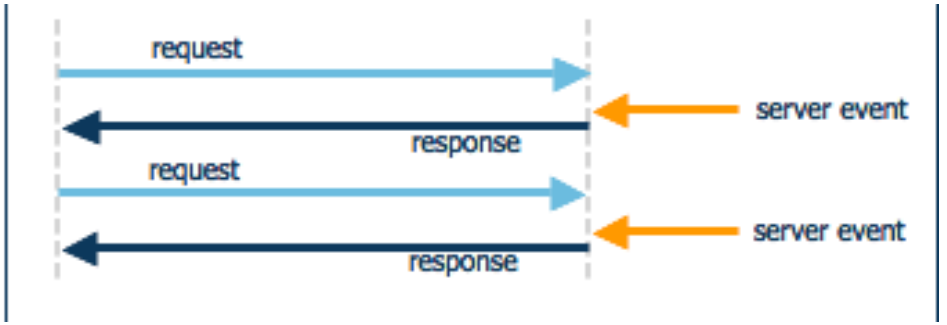
- Users Web GUIs need to be notified (updated) in case of a server side event (change server side state)
  - es. a new Email is received, FB new comment, auction new offer, …
- Update frequency is a key aspect
- Event/Message Driven paradigms fit well on describe this context.
  - Message Driven, Event Driven ,...
  - Reactive Programming, SOA, Time-driven programming (RTC), ...

Comet

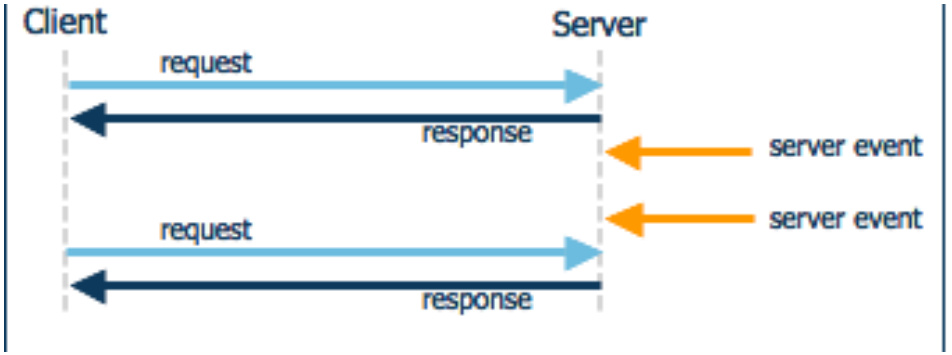
Comet is a web application model in which a long-held HTTP request allows a web server to push data to a browser, without the browser explicitly requesting it.

Comet is known by several other names, including: *Ajax Push, Reverse Ajax, Two-way-web,HTTP Streaming, and HTTP server push, ...*

Polling Techniques



- Waste of Requests also without server side changes
- Each request is new, HTTP handshake,…
- Not scalable
- Need to manage periodc requests ,..



- It reduces the use of resources, less http new requests, but same has same disadvantages of polling…
- Need to manage reconnection,..
- Not possible for firewall / web servers distinguish between long polling connection and slow connections ...

Others: HTTP Streaming, Server-sent events

Web socket :: What is ?

- WebSocket is a protocol, providing full-duplex communication channels over a single TCP connection.
- The WebSocket protocol was standardized by the IETF as RFC 6455 in 2011, and the WebSocket API in Web IDL is being standardized by the W3C.
- WebSocket is designed to be implemented in web browsers and web servers, but it can be used by any client or server application.
- The WebSocket Protocol is an independent TCP-based protocol. Its only relationship to HTTP is that its handshake is interpreted by HTTP servers as an Upgrade request.
- The url scheme are “ws:// or wss://” , default ports are the same of HTTP

Protocol handshake

```
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

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

WS :: Subprotocol

- WS It does not define any application protocol
- WS It is at too low level, applications need to understand the meaning of the messages.
- A sub-protocol can be negotiated during hand-shake.
  - STOMP, WAMP, XMPP, ..
- Spring supports STOMP

STOMP

Simple (or Streaming) Text Oriented Message Protocol (STOMP), formerly known as TTMP, is a simple text-based protocol, designed for working with message-oriented middleware (MOM).

The protocol is similar to HTTP, and works over TCP using commands like: CONNECT, SEND, SUBSCRIBE, UNSUBSCRIBE, ...

Communication is through a **frame** consisting of a number of lines. The first line contains the command, followed by headers in the form <key>: <value> (one per line), followed by a blank line and then the body content, ending in a null character.

Spring WebSocket & STOMP

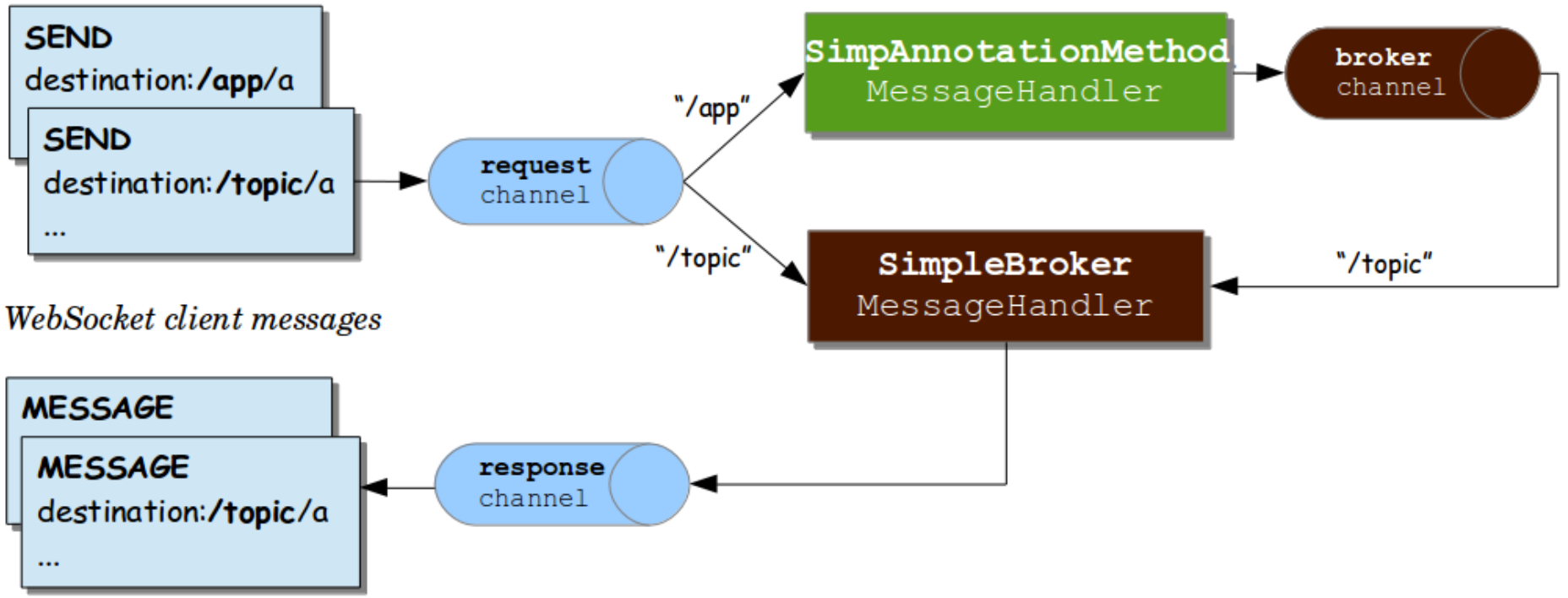
- It is a event driven message architecture
- Compatible with JSR 356: JavaTM API for WebSocket
- Destinations can be:
  - Message Controllers (application)
  - Message Brokers
    - Simple
      - In memory, built-in
    - Replay
      - Better for scaling, RabbitMQ, ActiveMQ, ...
  - A specific user/client

SockJS

- SockJS is a browser JavaScript library that provides a WebSocket-like object. SockJS gives you a coherent, cross-browser, Javascript API which creates a low latency, full duplex, cross-domain communication channel between the browser and the web server.
- Under the hood SockJS tries to use native WebSockets first. If that fails it can use a variety of browser-specific transport protocols and presents them through WebSocket-like abstractions.
- SockJS is intended to work for all modern browsers and in environments which don't support the WebSocket protocol -- for example, behind restrictive corporate proxies.
- SockJS-client does require a server counterpart
- Spring has out-of-box server side support .

HWCE
Functional Demo

STOMP Over WebSocket Messaging Architecture



HWCE :: WebSocketConfig :: SS

Server side

`/topic = Simple Broker`

`/app` => MessageController “mapped” methods

- `/message` , *onSend* save and send to broker

`/users` => MessageController “mapped” methods

- `/init/messages` , *onSubcribe* send saved msgs directly only to the client

HWCE :: Client Side Config

Connect

```
var socketRT = new SockJS('/hello-world-websocket');
stompClientRT = Stomp.over(socketRT);
stompClientRT.connect({}, function (frame) { ..} ...
```

Subscriptions and sends

```
stompClientRT.subscribe('/topic/messages' ..
stompClientRT.send("/app/message", {},
                              JSON.stringify( { … } ...
stompClientRT.subscribe('/user/init/messages' ...
```

HWCE
Demo Code !!

Questions or Beer ?