

Push and Client-Server

Misc Slides #1

Content

- Realtime Web
- Server push
- Server sent events
- Websocket

The Chat Problem



How to build a Web Application Chat?

- Messages broadcasted to all clients ...
- ... but HTTP "can't do" that (it's a request-response protocol)

Real Time Web



"The real-time web is a set of technologies and practices that enable users to receive information **as soon** as it is published by its authors, rather than requiring that they or their software check a source periodically for updates [polling]"
//Wikipedia

This is **soft** real time (**as soon** ~ in a few seconds)

Real Time Technologies

Many Terms Used Across the Years to Refer to All This

A word cloud of various real-time technologies and protocols. The words are arranged in a roughly circular pattern, with some appearing more frequently than others. The terms include: Real-Time Messaging, Push Technology, Web Streaming, Comet, WebSockets, Long Polling, Real-Time Notifications, Data Push, Web Push, Internet Messaging, Data Streaming, Ajax Push, Last Mile Messaging, Reverse Ajax, and others...

Real-Time Messaging Push Technology Web Streaming
Real-Time Web Comet WebSockets
Real-Time Notifications Long Polling
Internet Messaging Data Push Web Push
Last Mile Messaging Data Streaming Ajax Push
Reverse Ajax and others...

Some examples

- [Java Applets](#) (deprecated)
- [Polling](#) (let JavaScript continuously call server)
- [Comet](#)
 - Long-Polling (tweaking HTTP to it's limits, a "hack")
 - HTTP Streaming
- HTML5 APIs
 - [Server Sent Events](#)
 - [WebSockets](#)

Nice [technical overview](#)

Server Sent Events

```
if (typeof (EventSource) !== "undefined") {  
    var source = new  
        EventSource("http://localhost:8080/sse/SseServer");  
    source.onmessage = function (event) {  
        document.getElementById("ServerTime").innerHTML += event.data  
        + "<br><br>";  
    };  
} else {  
    document.getElementById("ServerTime").innerHTML =  
        "Sorry, your browser does not support server-sent events...";  
}
```

"To enable servers to push data to Web pages over HTTP or using dedicated server-push protocols"

// W3C

Basically

- Plain old HTTP (others possible)
- MIME: text/event-stream
- [API](#)
- Not supported by IE, other browsers: late versions
- Java Server Side support by Servlet or [Jersey](#) (JAX-RS)

PrimeFaces Push

```
<p:commandButton value="Send"
  actionListener="#{notifyView.send}" />

<p:socket onMessage="handleMessage" channel="/notify" />
<script type="text/javascript">
  function handleMessage(facesmessage) {
    ...
  }
</script>
```

```
@Named
@RequestScoped
public class NotifyView {
    public void send() {
        EventBus eventBus = EventBusFactory.
            getDefault().eventBus();
        eventBus.publish(CHANNEL,
            new FacesMessage(StringEscapeUtils
                .escapeHtml(summary),
                StringEscapeUtils
                .escapeHtml(detail)));
    }
}

@PushEndpoint("/notify")
public class NotifyResource{
    @OnMessage(encoders={JSONEncoder.class})
    public FacesMessage onMessage(
        FacesMessage message) {
        Object[] {message.getSummary(),
            message.getDetail()});
        return message;
    }
}
```

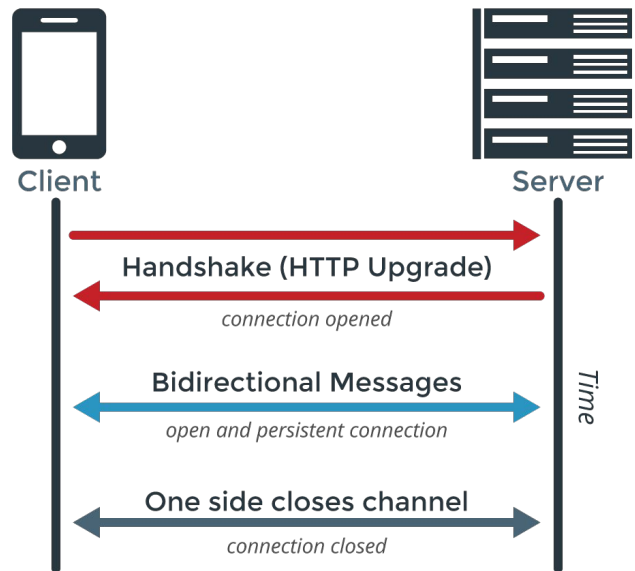
POST

Server

Push ([Notify](#)) included in PrimeFaces component suite (JSF)

- Using version 5.x of PrimeFaces
- Built on top of [Atmosphere](#) (supporting many of the mentioned technologies)

WebSocket



To enable Web applications to maintain [full duplex] bidirectional communications with server-side processes
// W3C

Basically

- Not like TCP Sockets
- Message based (like UDP) reliable (like TCP)
- Initial HTTP-handshake then switch to websocket [protocol](#) (ws://.. and wss://)
- W3C [WebSocket interface](#) (client side)
- Server side depends on platform
 - [JEE Server side websocket](#)
- [Some comments](#)