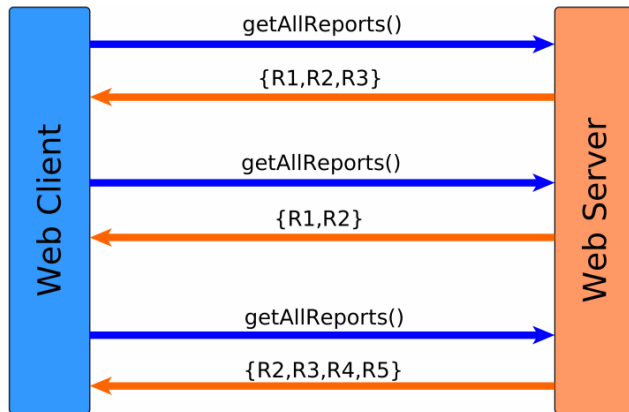


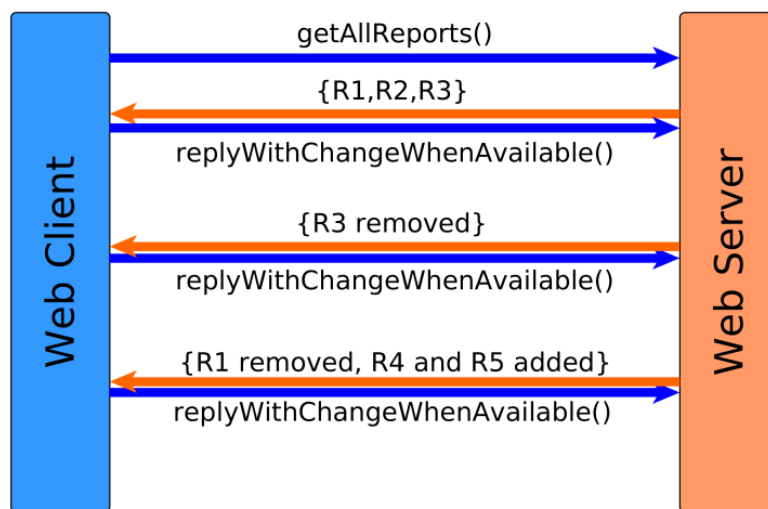
Aplikační servery, Web sockets

Co dělat, když se na serveru objeví nová data a klient o tom neví?

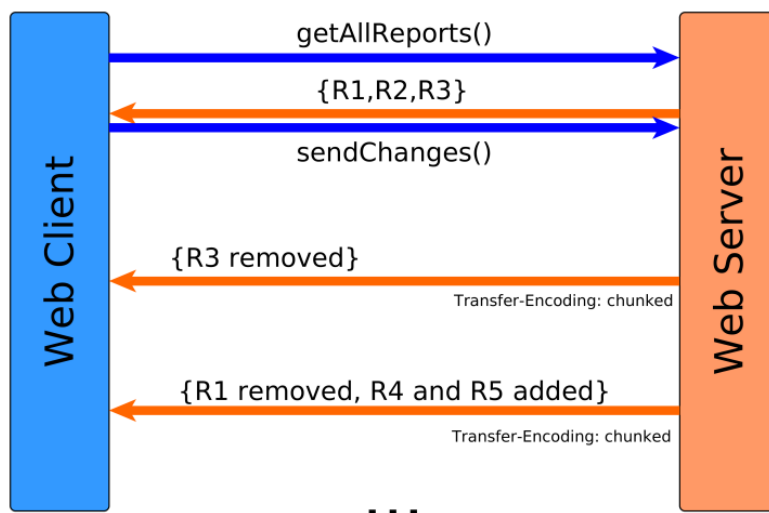
Simple Solution Using HTTP



Better Solution Using HTTP – AJAX push (Long Poll)



Better Solution Using HTTP – AJAX push (Streaming)



AJAX push

- klient se nemusí neustále dotazovat, jestli nejsou na serveru nová data
- řekne serveru, ať ho informuje, pokud se nová data objeví
- ALE:
 - o klient musí vytvořit nové HTTP spojení pro každý typ (getReports, getUsers,...)
 - o HTTP headery se posílají tam a zpět
 - o klient musí rozumět a parsovat low-level HTTP chunks

Long poll

- klient pošle request na server
- server nechá request otevřený, dokud nejsou dostupná nová data
- klient obdrží data a pošle další request

Streaming

- podobné jako LP, ale server odpoví headerem s Transfer-Encoding: chunked
- není třeba pokaždé iniciovat nový request

WebSockets

- komunikační protokol pro klient-server
- součást Java EE 7
- obousměrné, full-duplex, real-time
- nízká odezva; přes TCP/IP

Web Socket Handshake

<pre>GET ws://server.org/wsendpoint HTTP/1.1 Host: server.org Connection: Upgrade Upgrade: websocket Origin: http://server.org Sec-WebSocket-Version: 13 Sec-WebSocket-Key: GhkZiCk+0/91FXIbUuRlVQ== Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits</pre>	<pre>HTTP/1.1 101 Switching Protocols Upgrade: websocket Connection: upgrade Sec-WebSocket-Accept: jpwu9a/SXDrsoRR260a3JUEFchY= Sec-WebSocket-Extensions: permessage-deflate;client_max_window ...</pre>
---	--

Java API for WebSocket

- annotations on POJOs to interact with WebSocket lifecycle events
- interfaces to implement to interact with WebSocket lifecycle events
- integration with other Java EE technologies – EJB, CDI

```
@ServerEndpoint("/actions")
public class WebSocketServer {

    @OnOpen
    public void open(Session session) { ... }

    @OnClose
    public void close(Session session) { ... }

    @OnError
    public void onError(Throwable error) { ... }

    @OnMessage
    public void handleMessage(String message, Session session) {
        // actual message processing
    }
}
```