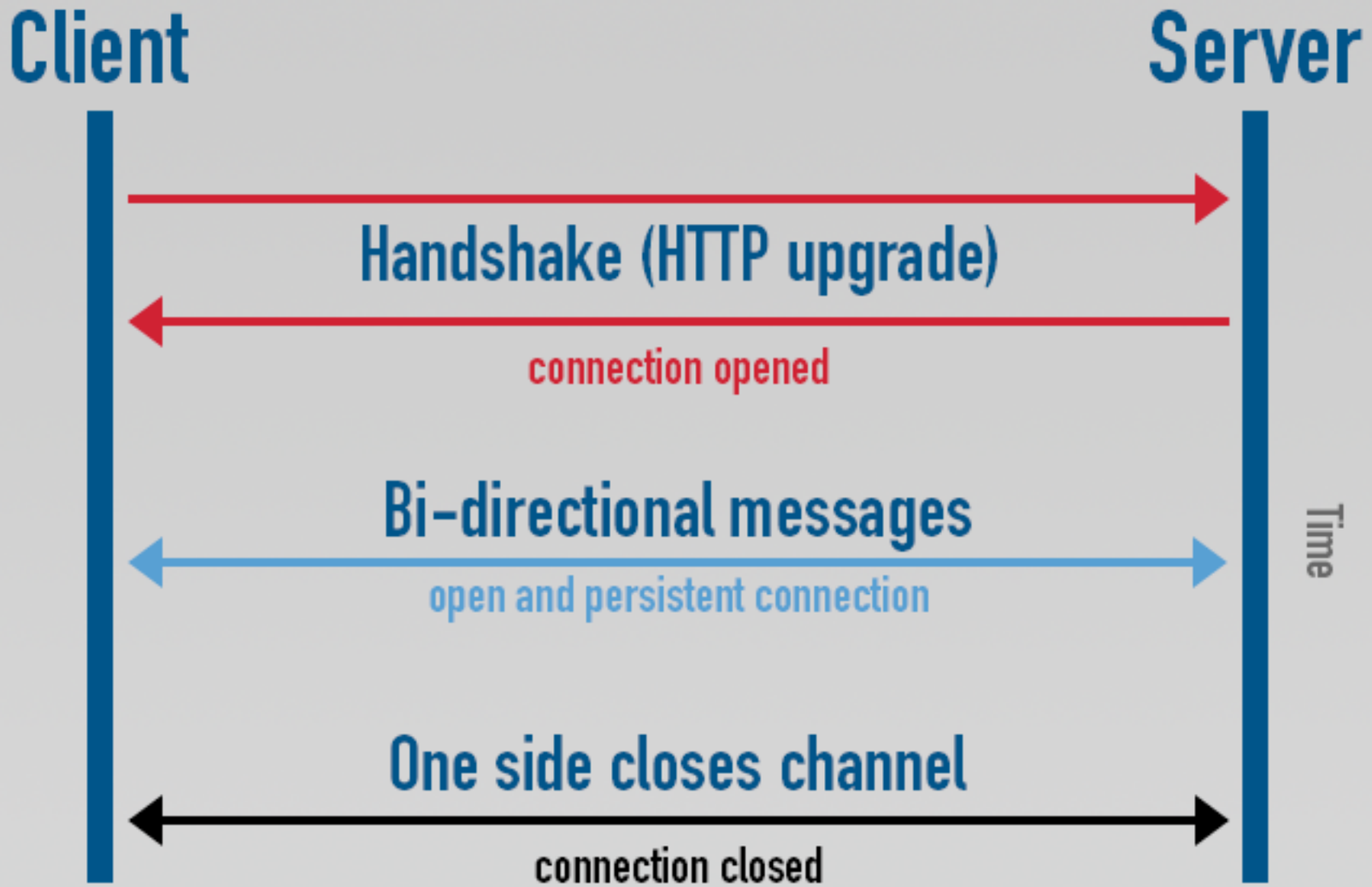
WebSockets

WEBSOCKETS

A VISUAL REPRESENTATION



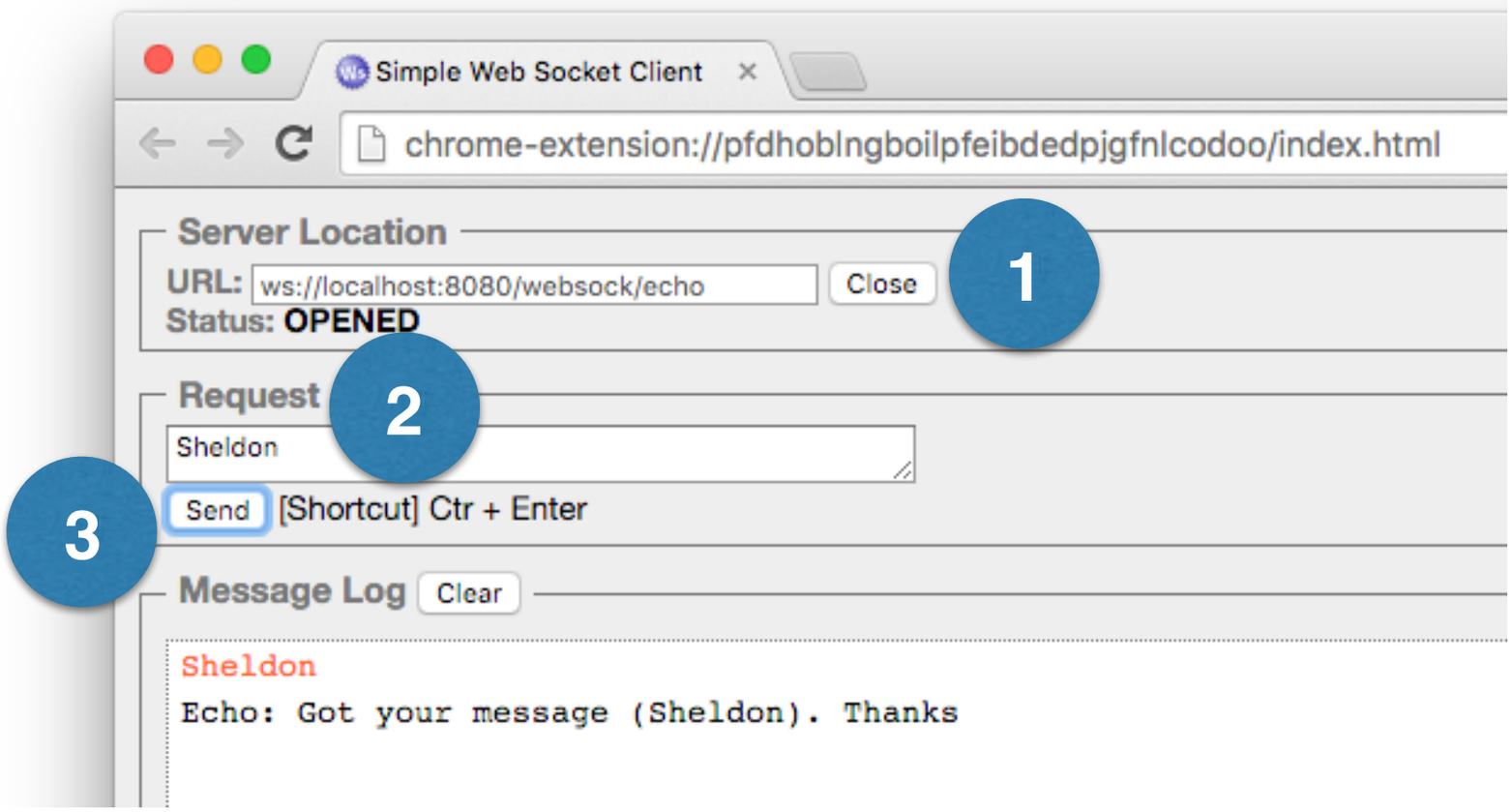
PubNub

pom.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>at.htl.websockets</groupId>
    <artifactId>websockets</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>war</packaging>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>

    <dependencies>
        <dependency>
            <groupId>javax</groupId>
            <artifactId>javaee-web-api</artifactId>
            <version>8.0</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>

    <build>
        <finalName>websocket</finalName>
    </build>

</project>
```

Echo

```java
@ServerEndpoint("/echo")
public class WebSocketServer {

    @OnMessage
    public String sayHello(String message) {
        return String.format("Echo: Got your message (%s). Thanks", message);
    }
}
```



Ein HTML/JS-Client

https://gist.github.com/miebach/3293565

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>WebSocket Echo Client</title>

    <script type="text/javascript">
        var wsUri = "ws://localhost:8080/websock/echo";
        var output;

        function init() {
            output = document.getElementById("output");
            testWebSocket();
        };

        function testWebSocket() {
            websocket = new WebSocket(wsUri);
            websocket.onopen = function (evt) { onOpen(evt) };
            websocket.onclose = function (evt) { onClose(evt) };
            websocket.onmessage = function (evt) { onMessage(evt) };
            websocket.onerror = function (evt) { onError(evt) };
        }

        function onOpen(evt)
        {
            writeToScreen("CONNECTED");
            doSend("WebSocket rocks");
```

...

Ergebnis


WebSocket Echo Client
localhost:8080/websock/echo.html
WebSocket Echo Client Test
CONNECTED
SENT: WebSocket rocks
RESPONSE: Echo: Got your message (WebSocket rocks). Thanks
DISCONNECTED

ChatServer

```java
@ServerEndpoint("/chat")
public class ChatEndpoint {
    @OnMessage
    public void message(String message, Session client) throws IOException, EncodeException {
        System.out.println("message: " + message);
        for (Session peer : client.getOpenSessions()) {
            peer.getBasicRemote().sendText(message);
        }
    }
}
```

https://github.com/javaee-samples/javaee7-samples/blob/master/websocket/chat/src/main/java/org/javaee7/websocket/chat/ChatEndpoint.java

Den Client finden Sie unter:
http://dreamand.me/java/jee7-websocket-example/

Aufgabe 1

- Erstellen Sie einen JavaFX-ChatClient mit WebSockets
- http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/BinaryWebSocket/binaryWebSocket.html

Aufgabe 2

- Erstellen Sie einen JavaFX-Client, der Screenshots auf den Server hochladen kann.
- Screenshot mit der Robot-API: http://www.rgagnon.com/javadetails/java-0489.html
- Am Server werden die Screenshots als Blob in einer DerbyDb gespeichert.

Aufgabe 3

- Erstellen Sie einen Primefaces-ChatClient mit WebSockets

WebSockets mit JSON

- http://www.masterthеboss.com/javaee/websockets/websockets-using-encoders-and-decoders
- ev. http://buttso.blogspot.co.at/2014/07/developing-with-websocket-and-json.html

Noch Fragen?

Quellen

- http://jlunaquiroga.blogspot.co.at/2014/05/websockets-in-jee-7-with-wildfly.html
- http://dreamand.me/java/jee7-websocket-example/
- https://www.pubnub.com/blog/2013-09-11-what-are-websockets/
- http://de.slideshare.net/shahzadbadar/javaee7-websockets
- http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/HomeWebsocket/WebsocketHome.html
- Pilgrim: JavaEE 7 Developer Handbook ,2013