

Desenvolvimento de Aplicações Java
Plataforma Corporativa

Tutorial
WebSockets

Outubro 2016

Sumário

1.Introdução.....	3
2. Conceituação.....	4
2.1. O WebSocket.....	4
Protocolo WebSocket	4
3. Exemplo.....	6
3.1. WSEndpoint.java	7
3.2. exemplo.html	9
3.3. exemplo2.html	10

1.Introdução

Introdução e uso do WebSocket.

2. Conceituação

2.1. O WebSocket

- Um protocolo de comunicação
- Uma API.
- Especificação da API definida pelo W3C.
- Protocolo de comunicação definido pelo IETF (Internet Engineering Task Force) RFC6455.
- “ WebSocket é uma tecnologia que permite a comunicação bidirecional por canais full-duplex sobre um único soquete Transmission Control Protocol (TCP). Ele é projetado para ser executado em browsers e servidores web que suportem o HTML5[1] , mas pode ser usado por qualquer cliente ou servidor de aplicativos.”
- wikipedia

Protocolo WebSocket

Mecanismo de mapeamento entre um nome (descrição simples) e um recurso computacional.

Utilizado para:

- Associar um nome a um recurso computacional (processo conhecido como *binding*).
- Localizar um recurso a partir de seu nome. Protocolo criado para funcionar na infraestrutura Web existente.
- A conexão WebSocket tem seu ciclo de vida iniciado como uma conexão HTTP, garantindo total compatibilidade com o mundo pré-WebSocket.
- Depois da conexão HTTP estabelecida, o protocolo é modificado de HTTP para WebSocket em um processo conhecido como WebSocket Handshake.
- O cliente envia uma requisição para o servidor indicando que deseja alterar o protocolo de HTTP para o WebSocket.
-

WebSocket Handshake

- O cliente envia sua intenção através o header Upgrade:

```
GET ws://echo.mackenzie.br/?encoding=text HTTP/1.1
Origin: http://www.mackenzie.br
Cookie: __utma=99as
Connection: Upgrade
Host: echo.mackenzie.br
Sec-WebSocket-Key: uRovscZjNol/umbTt5uKmw==
Upgrade: websocket
Sec-WebSocket-Version: 13
```

- Se o servidor aceitar a proposta do cliente:

- **HTTP/1.1 101 WebSocket Protocol Handshake**
- **Date: Fri, 10 Feb 2012 17:38:18 GMT**
- **Connection: Upgrade**
- **Server: Wildfly Gateway**
- **Upgrade: WebSocket**
- **Access-Control-Allow-Origin: http://www.mackenzie.br**
- **Access-Control-Allow-Credentials: true**
- **Sec-WebSocket-Accept: rLHCkw/SKsO9GAH/ZSFhBATDKrU=**
- **Access-Control-Allow-Headers: content-type**

Conexão WebSocket

- Depois do WebSocket Handshake bem sucedido:
 - Conexão HTTP é derrubada e substituída pela conexão WebSocket, na mesma conexão TCP/IP.
 - Portanto, a conexão WebSocket continua nas portas HTTP (80) WS e HTTPS(443) WSS.

3. Exemplo

Utilizar o projeto corporativo. Não se esqueça de atualizar as URLs nas classes e páginas html de acordo com o nome de sua aplicação WEB.

- Implementar a classe `mack.websocket.WSEndpoint`.
- Implementar a página `exemplo.html`.
- Implementar a página `exemplo2.html`.
- Limpar e construir.
- Implantar no Wildfly.

3.1. WSEndpoint.java

```
package mack.websocket;

import java.io.IOException;
import javax.websocket.server.ServerEndpoint;

import javax.websocket.CloseReason;
import javax.websocket.OnClose;
import javax.websocket.OnMessage;
import javax.websocket.OnOpen;
import javax.websocket.Session;

@ServerEndpoint("/example")
public class WSEndpoint {

    @OnMessage
    public String receiveMessage(String message, Session session) {
        System.out.println("Received : "+ message + ", session:" + session.getId());
        return "Response from the server";
    }

    @OnOpen
    public void open(Session session) {
        System.out.println("Open session:" + session.getId());
        final Session s = session;
        try{
            s.getBasicRemote().sendText("Message from server");
        } catch ( IOException e) {
            e.printStackTrace();
        }
    }

    @OnClose
    public void close(Session session, CloseReason c) {
```

```
System.out.println("Closing:" + session.getId());
```

```
}
```

```
}
```


3.2. exemplo.html

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>WebSocket Client</title>

<script type="text/javascript">

    var ws = new WebSocket("ws://localhost:8080/AppFrontController/example");

    ws.onopen = function()

    {

        alert("Web Socket is connected!!");

    };

    ws.onmessage = function (evt)

    {

        var msg = evt.data;

        alert("Message received:" + msg);

    };

    ws.onclose = function()

    {

        alert("Connection is closed...");

    };

</script>

</head>

<body>


</body>

</html>
```

3.3. exemplo2.html

```
<html>

  <head>

    <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">

  </head>

  <body>

    <meta charset="utf-8">

    <title>Web sockets</title>

    <script language="javascript" type="text/javascript">

      var wsUri = getRootUri() + "/AppFrontController/example";

      function getRootUri() {

        return "ws://" + (document.location.hostname == "" ? "localhost" :
document.location.hostname) + ":" +

          (document.location.port == "" ? "8080" : document.location.port);

      }

      function init() {

        output = document.getElementById("output");

      }

      function send_message() {

        websocket = new WebSocket(wsUri);

        websocket.onopen = function(evt) {

          onOpen(evt)

        };

        websocket.onmessage = function(evt) {

          onMessage(evt)

        };

        websocket.onerror = function(evt) {

          onError(evt)

        };

      }

    </script>

  </body>

</html>
```

```
}

function onOpen(evt) {
    writeToScreen("Connected to Endpoint!");
    doSend(textID.value);
}

function onMessage(evt) {
    writeToScreen("Message Received: " + evt.data);
}

function onError(evt) {
    writeToScreen('<span style="color: red;">ERROR:</span> ' + evt.data);
}

function doSend(message) {
    writeToScreen("Message Sent: " + message);
    websocket.send(message);
}

function writeToScreen(message) {
    var pre = document.createElement("p");
    pre.style.wordWrap = "break-word";
    pre.innerHTML = message;

    output.appendChild(pre);
}

window.addEventListener("load", init, false);

</script>

<h1 style="text-align: center;">Hello World WebSocket Client</h2>
```

```
<br>

<div style="text-align: center;">
  <form action="">
    <input onclick="send_message()" value="Send" type="button">
    <input id="textID" name="message" value="Hello WebSocket!" type="text"><br>
  </form>
</div>

<div id="output"></div>
</body>
</html>
```