

Hello World WebSocket tutorial

Balázs Simon (sbalazs@iit.bme.hu), BME IIT, 2015

1 Introduction

This document describes how to create a Hello World WebSocket service using Eclipse and WildFly.

2 Application

Create a new **Dynamic Web Project** in Eclipse called **HelloWebSocket** accepting the default settings of the wizard.

Add a new class called **HelloEndpoint** under the package **helloworlds** with the following content:

```
1 package helloworlds;
2
3 import javax.websocket.OnClose;
4 import javax.websocket.OnError;
5 import javax.websocket.OnMessage;
6 import javax.websocket.OnOpen;
7 import javax.websocket.Session;
8 import javax.websocket.server.ServerEndpoint;
9
10 @ServerEndpoint("/hello")
11 public class HelloEndpoint {
12     @OnOpen
13     public void open(Session session) {
14         System.out.println("WebSocket opened: " + session.getId());
15     }
16
17     @OnClose
18     public void close(Session session) {
19         System.out.println("WebSocket closed: " + session.getId());
20     }
21
22     @OnError
23     public void error(Throwable t) {
24         System.out.println("WebSocket error: " + t.getMessage());
25     }
26
27     @OnMessage
28     public String open(String message) {
29         System.out.println("WebSocket message: " + message);
30         return "Hello: " + message;
31     }
32 }
33
```

Add a new HTML file called **hello.html** under the **WebContent** folder with the following content:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Hello World WebSocket client</title>
  <script language="javascript" type="text/javascript">
    var wsUrl = getRootUri() + "/HelloWebSocket/hello";
    var websocket = null;

    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 initWebSocket() {
      websocket = new WebSocket(wsUrl);
      websocket.onopen = function(evt) {
        onOpen(evt);
        doSend();
      };
      websocket.onmessage = function(evt) {
        onMessage(evt);
      };
      websocket.onerror = function(evt) {
        onError(evt);
        websocket = null;
      };
      websocket.onclose = function(evt) {
        onClose(evt);
        websocket = null;
      };
    }

    function send_message() {
      if (websocket == null) {
        initWebSocket();
      } else {
        doSend();
      }
    }

    function onOpen(evt) {
      writeToScreen("Connected to endpoint.");
    }
  </script>
</head>
<body>
  <div id="output"></div>
</body>
</html>
```

```

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

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

function onClose(evt) {
    writeToScreen("Connection closed.");
}

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

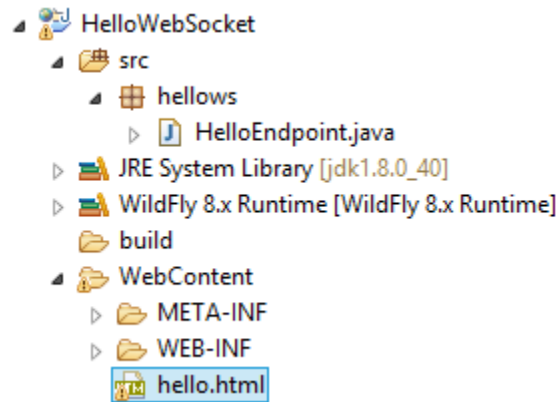
</script>
</head>
<body>
    <h1 style="text-align: center;">Hello World WebSocket client</h2>

    <br>

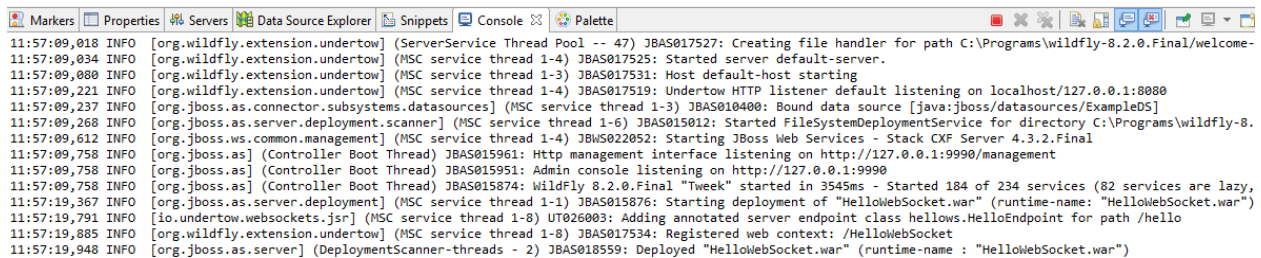
    <div style="text-align: center;">
        <form action="">
            <input onclick="send_message()" value="Send" type="button">
            <input id="textID" name="message" value="me" type="text"><br>
        </form>
    </div>
    <div id="output"></div>
</body>
</html>

```

The project should look like this:



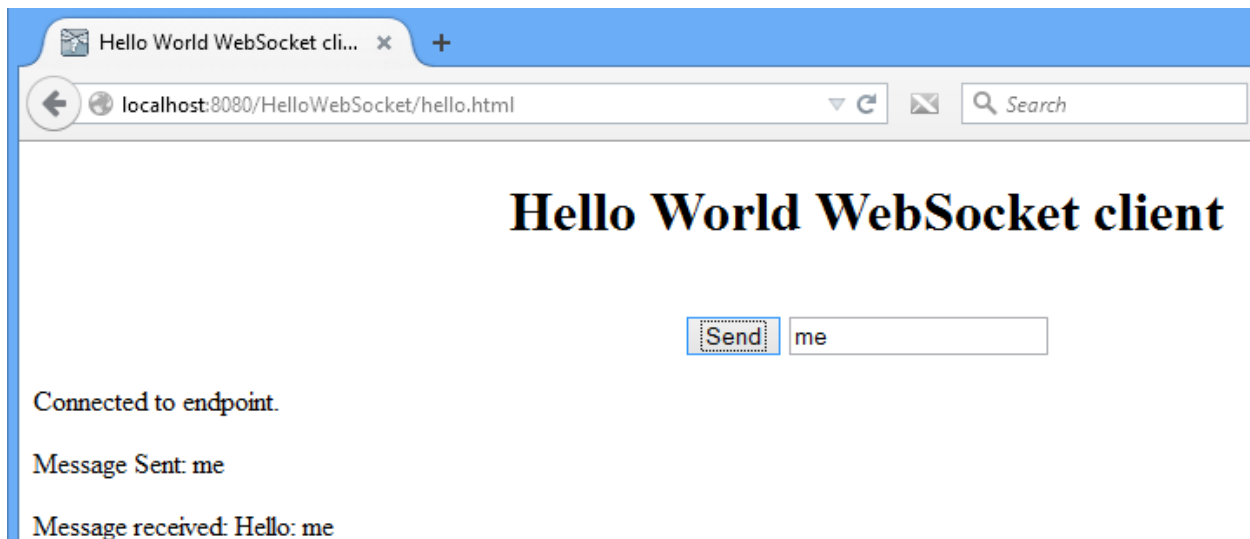
Right click on the project and deploy it to the WildFly server. It should deploy without errors:



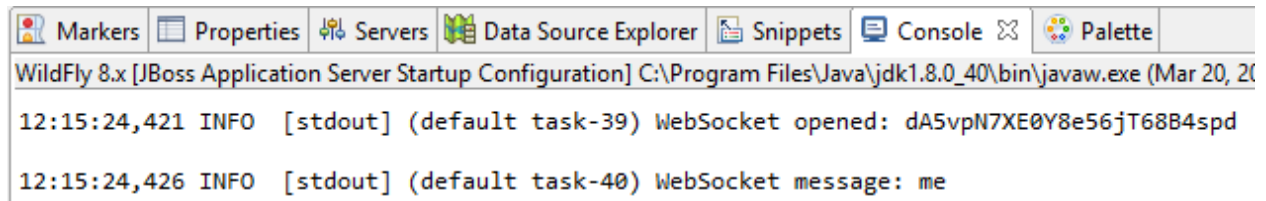
Open the following URL in a browser to test the application:

<http://localhost:8080/HelloWebSocket/hello.html>

Type something in the input field and click on the **Send** button:



Also, in the server log there should be:



The screenshot shows the Eclipse IDE's console window. The top toolbar includes icons for Markers, Properties, Servers, Data Source Explorer, Snippets, Console, and Palette. The console title bar reads "WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_40\bin\javaw.exe (Mar 20, 2016)". The log output contains two lines: "12:15:24,421 INFO [stdout] (default task-39) WebSocket opened: dA5vpN7XE0Y8e56jT68B4spd" and "12:15:24,426 INFO [stdout] (default task-40) WebSocket message: me".

```
WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_40\bin\javaw.exe (Mar 20, 2016)
12:15:24,421 INFO [stdout] (default task-39) WebSocket opened: dA5vpN7XE0Y8e56jT68B4spd
12:15:24,426 INFO [stdout] (default task-40) WebSocket message: me
```