```
@ServerEndpoint("/echo")
public class EchoServer {
   /**
    * @OnOpen allows us to intercept the creation of a new session.
    * The session class allows us to send data to the user.
    * In the method onOpen, we'll let the user know that the handshake was
    * successful.
    */    @OnOpen
   public void onOpen(Session session){
      System.out.println(session.getId() + " has opened a connection");
      try {
         session.getBasicRemote().sendText("Connection Established");
      } catch (IOException ex) {
         ex.printStackTrace();
      }
   }

   /**
    * When a user sends a message to the server, this method will intercept the message
    * and allow us to react to it. For now the message is read as a String.
    */    @OnMessage
   public void onMessage(String message, Session session){
      System.out.println("Message from " + session.getId() + ": " + message);
      try {
         session.getBasicRemote().sendText(message);
      } catch (IOException ex) {
         ex.printStackTrace();
      }
   }
```

```
<!DOCTYPE html>

<html>
   <head>
      <title>Echo Chamber</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width">
   </head>
   <body>

      <div>
         <input type="text" id="messageinput"/>
      </div>
      <div>
         <button type="button" onclick="openSocket();" >Open</button>
         <button type="button" onclick="send();" >Send</button>
         <button type="button" onclick="closeSocket();" >Close</button>
      </div>
      <!-- Server responses get written here -->
      <div id="messages"></div>

      <!-- Script to utilise the WebSocket -->
      <script type="text/javascript">

         var webSocket;
         var messages = document.getElementById("messages");



         function openSocket(){
            // Ensures only one connection is open at a time
```

```
    if(webSocket !== undefined && webSocket.readyState !== WebSocket.CLOSED){
       writeResponse("WebSocket is already opened.");
        return;
    }
    // Create a new instance of the websocket
    writeResponse("Trying de create a connection.");
    webSocket = new WebSocket("ws://localhost:8080/EchoChamber/echo");

    /**
     * Binds functions to the listeners for the websocket.
     */
    webSocket.onopen = function(event){
        // For reasons I can't determine, onopen gets called twice
        // and the first time event.data is undefined.
        // Leave a comment if you know the answer.
        if(event.data === undefined)
            return;

        writeResponse(event.data);
    };

    webSocket.onmessage = function(event){
        writeResponse(event.data);
    };

    webSocket.onclose = function(event){
        writeResponse("Connection closed");
    };
}

/**
```

```
         * Sends the value of the text input to the server
         */
        function send(){
            var text = document.getElementById("messageinput").value;
            webSocket.send(text);
        }

        function closeSocket(){
            webSocket.close();
        }

        function writeResponse(text){
            messages.innerHTML += "<br/>" + text;
        }

    </script>

  </body>
</html>
  /**
   * The user closes the connection.
   *
   * Note: you can't send messages to the client from this method
   */   @OnClose
  public void onClose(Session session){
    System.out.println("Session " +session.getId()+" has ended");
  }
}
```