

WebSocket

Introduction

- ❖ HTML 5 WebSockets defines a communication channel that operates over the web and allows both direction communication over a single socket.
- ❖ Using HTML 5 WebSockets we can dramatically reduce unnecessary network traffic and latency

Introduction

- ❖ Using HTML 5 WebSockets, when data changes on the web server the web server can send a request to the client. We no longer need to implement a client that polls the server.

Web Browser Support

- ❖ We can easily check whether the web browser supports WebSocket or not.

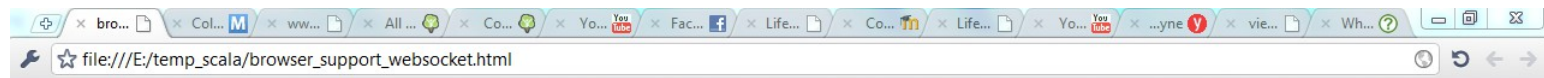
```
if (window.WebSocket)
{
    ...
}
```

Web Browser Support

```
<div id="msg"></div>
<script>
if (window.WebSocket)
{
    document.getElementById("msg").innerHTML = "browser supports";
}
else
{
    document.getElementById("msg").innerHTML = "browser doesn't support";
}
</script>
```



Web Browser Support



browser supports



Creating Web Socket

```
var ws = new WebSocket("ws://services.abelski.com/samples");
```

We should pass over to the WebSocket constructor the URL address of the web socket server we intend to use. That address should start with 'ws' which stands for Web Sockets.

Call Back Functions

```
ws.onopen = function(event)
{
    ...
}
```

This function will be called when the connection is established.

Call Back Functions

```
ws.onmessage = function(event)
{
    alert(event.data);
    ...
}
```

This function will be called when a message arrives from the server.

Call Back Functions

```
ws.onclose = function(event)
{
    ...
}
```

This function will be called when the connection is closed.

Sending Data

```
ws.postMessage("this is the message sent to the server");
```

We call the `postMessage` function in order to send a message

Close Connection

```
ws.disconnect();
```

Calling this function will disconnect the connection with the server.

Sample

- ❖ The following code sample uses Java EE support for Web Sockets API as was implemented in Apache Tomcat.

Sample

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple WebSockets Chat</title>
  <style type="text/css">
    input#user-message {
      width: 360px
    }
    #chat-messages {
      width: 355px;
      border: 1px solid #BBBBBB;
      height: 200px;
      overflow-y: scroll;
      padding: 4px;
    }
  </style>
</head>
```

The HTML File



Sample

```
<body>
<div>
  <input type="text" placeholder="enter your message here"
    id="user-message">
  <div id="chat-messages"></div>
</div>
<script type="text/javascript">
  var chat = {};
  chat.socket = null;
  chat.connect = (function(host) {
    if ('WebSocket' in window) {
      chat.socket = new WebSocket(host);
    } else if ('MozWebSocket' in window) {
      chat.socket = new MozWebSocket(host);
    } else {
      return;
    }
    chat.socket.onopen = function () {
      document.getElementById('user-message').onkeydown =
        function(event) {
          if (event.keyCode == 13) {
            chat.sendMessage();
          }
        };
    };
  });
};
```

Sample

```
chat.socket.onclose = function () {
    document.getElementById('user-message').onkeydown = null;
};
chat.socket.onmessage = function (message) {
    messages.add(message.data);
};
});
chat.initialize = function() {
    if (window.location.protocol == 'http:') {
        chat.connect(
            'ws://' + window.location.host +
            '/websocketproj/WebSocket');
    } else {
        chat.connect
            ('wss://' + window.location.host +
            '/websocketproj/WebSocket');
    }
};
chat.sendMessage = (function() {
    var message = document.getElementById('user-message').value;
    if (message != '') {
        chat.socket.send(message);
        document.getElementById('user-message').value = '';
    }
});
```


Sample

```
var messages = {};  
messages.add = (function(message) {  
    var ob = document.getElementById('chat-messages');  
    var div = document.createElement('div');  
    div.style.wordWrap = 'break-word';  
    div.innerHTML = message;  
    ob.appendChild(div);  
    while (ob.childNodes.length > 10) {  
        ob.removeChild(ob.firstChild);  
    }  
});  
chat.initialize();  
</script>  
</body>  
</html>
```

Sample

The Servlet File

```
@WebServlet("/WebSocket")
public class WebSocket extends WebSocketServlet {

    private static final long serialVersionUID = 1L;

    private final AtomicInteger idGenerator = new AtomicInteger(0);
    private final Set<ChatUser> connections = new HashSet<ChatUser>();

    @Override
    protected StreamInbound createWebSocketInbound(String subProtocol,
        HttpServletRequest request) {
        return new ChatUser(idGenerator.incrementAndGet());
    }

    private final class ChatUser extends MessageInbound {

        private final String username;

        private ChatUser(int id) {
            this.username = "guest #" + id;
        }
    }
}
```

Sample

```
@Override
protected void onOpen(WsOutbound outbound) {
    connections.add(this);
    String message = username + " has joined the chat";
    broadcast(message);
}
```

```
@Override
protected void onClose(int status) {
    connections.remove(this);
    String message = username + " has left the chat";
    broadcast(message);
}
```

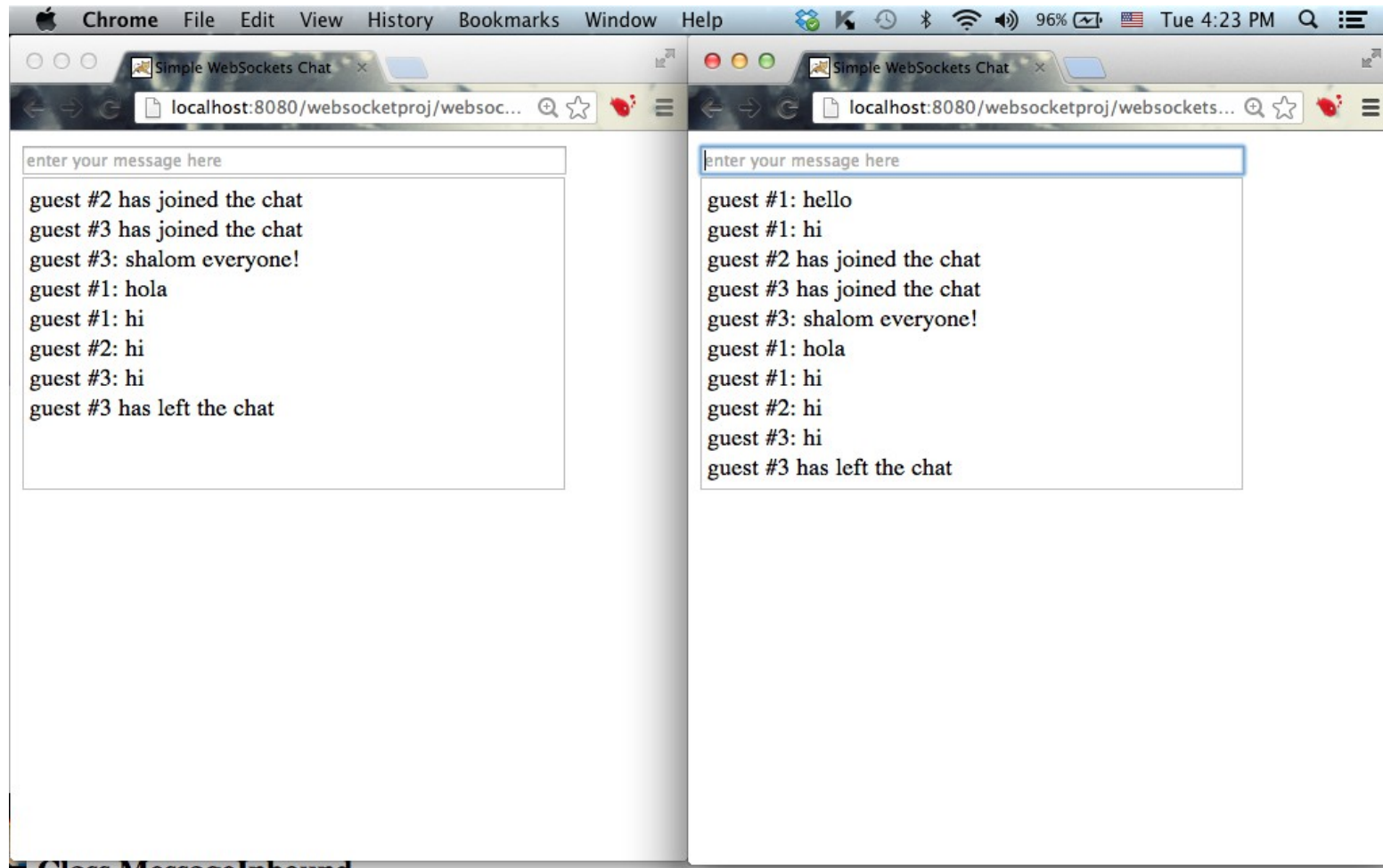
```
@Override
protected void onBinaryMessage(ByteBuffer message) throws IOException {
    throw new UnsupportedOperationException(
        "binary messages are not supported");
}
```

Sample

```
@Override
protected void onTextMessage(CharBuffer message) throws IOException {
    // it would be best filter the message
    // ...
    String str = username + ": " + message;
    broadcast(str);
}

private void broadcast(String message) {
    for (ChatUser connection : connections) {
        try {
            CharBuffer buffer = CharBuffer.wrap(message);
            connection.getWsOutbound().writeTextMessage(buffer);
        } catch (IOException ignore) {
            // ...
        }
    }
}
}
```

Sample



WebSocket

Introduction

- ❖ HTML 5 WebSockets defines a communication channel that operates over the web and allows both direction communication over a single socket.
- ❖ Using HTML 5 WebSockets we can dramatically reduce unnecessary network traffic and latency

© 2010 Haim Michael. All Rights Reserved.

Introduction

- ❖ Using HTML 5 WebSockets, when data changes on the web server the web server can send a request to the client. We no longer need to implement a client that polls the server.

© 2010 Haim Michael. All Rights Reserved.

Web Browser Support

- ❖ We can easily check whether the web browser supports WebSocket or not.

```
if (window.WebSocket)
{
    ...
}
```

© 2010 Haim Michael. All Rights Reserved.

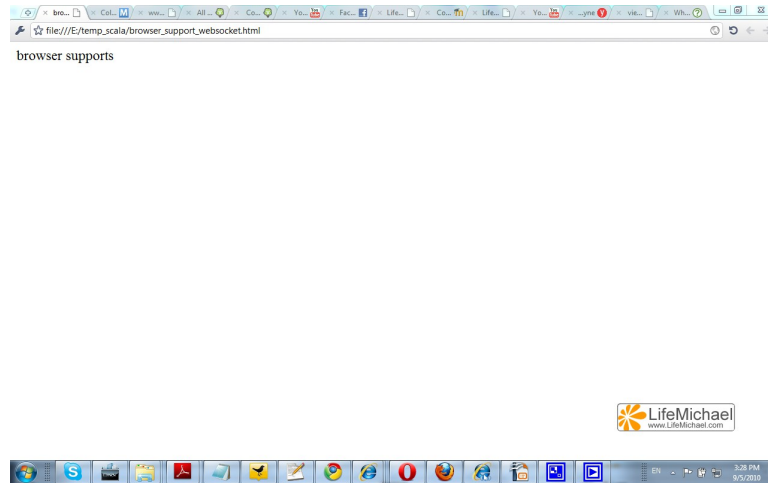
Web Browser Support

```
<div id="msg"></div>
<script>
if (window.WebSocket)
{
    document.getElementById("msg").innerHTML = "browser supports";
}
else
{
    document.getElementById("msg").innerHTML = "browser doesn't support";
}
</script>
```



© 2010 Haim Michael. All Rights Reserved.

Web Browser Support



© 2010 Haim Michael. All Rights Reserved.

Creating Web Socket

```
var ws = new WebSocket("ws://services.abelski.com/samples");
```

We should pass over to the WebSocket constructor the URL address of the web socket server we intend to use. That address should start with 'ws' which stands for Web Sockets.

© 2010 Haim Michael. All Rights Reserved.

Call Back Functions

```
ws.onopen = function(event)
{
    ...
}
```

This function will be called when the connection is established.

© 2010 Haim Michael. All Rights Reserved.

Call Back Functions

```
ws.onmessage = function(event)
{
    alert(event.data);
    ...
}
```

This function will be called when a message arrives from the server.

© 2010 Haim Michael. All Rights Reserved.

Call Back Functions

```
ws.onclose = function(event)
{
    ...
}
```

This function will be called when the connection is closed.

© 2010 Haim Michael. All Rights Reserved.

Sending Data

```
ws.postMessage("this is the message sent to the server");
```

We call the `postMessage` function in order to send a message

© 2010 Haim Michael. All Rights Reserved.

Close Connection

```
ws.disconnect();
```

Calling this function will disconnect the connection with the server.

© 2010 Haim Michael. All Rights Reserved.

Sample

- ❖ The following code sample uses Java EE support for Web Sockets API as was implemented in Apache Tomcat.

© 2010 Haim Michael. All Rights Reserved.

Sample

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple WebSockets Chat</title>
  <style type="text/css">
    input#user-message {
      width: 360px
    }
    #chat-messages {
      width: 355px;
      border: 1px solid #BBBBBB;
      height: 200px;
      overflow-y: scroll;
      padding: 4px;
    }
  </style>
</head>
```

The HTML File



© 2010 Haim Michael. All Rights Reserved.

Sample

```
<body>
<div>
  <input type="text" placeholder="enter your message here"
    id="user-message">
  <div id="chat-messages"></div>
</div>
<script type="text/javascript">
  var chat = {};
  chat.socket = null;
  chat.connect = (function(host) {
    if ('WebSocket' in window) {
      chat.socket = new WebSocket(host);
    } else if ('MozWebSocket' in window) {
      chat.socket = new MozWebSocket(host);
    } else {
      return;
    }
  })();
  chat.socket.onopen = function () {
    document.getElementById('user-message').onkeydown =
      function(event) {
        if (event.keyCode == 13) {
          chat.sendMessage();
        }
      };
  };
};
```

© 2010 Haim Michael. All Rights Reserved.

Sample

```
chat.socket.onclose = function () {
    document.getElementById('user-message').onkeydown = null;
};
chat.socket.onmessage = function (message) {
    messages.add(message.data);
};
});
chat.initialize = function() {
    if (window.location.protocol == 'http:') {
        chat.connect(
            'ws://' + window.location.host +
            '/websocketproj/WebSocket');
    } else {
        chat.connect(
            'wss://' + window.location.host +
            '/websocketproj/WebSocket');
    }
};
chat.sendMessage = (function() {
    var message = document.getElementById('user-message').value;
    if (message != '') {
        chat.socket.send(message);
        document.getElementById('user-message').value = '';
    }
})();
```

© 2010 Haim Michael. All Rights Reserved.

Sample

```
var messages = {};  
messages.add = (function(message) {  
    var ob = document.getElementById('chat-messages');  
    var div = document.createElement('div');  
    div.style.wordWrap = 'break-word';  
    div.innerHTML = message;  
    ob.appendChild(div);  
    while (ob.childNodes.length > 10) {  
        ob.removeChild(ob.firstChild);  
    }  
});  
chat.initialize();  
</script>  
</body>  
</html>
```

© 2010 Haim Michael. All Rights Reserved.

Sample

```
@WebServlet("/WebSocket")
public class WebSocket extends WebSocketServlet {

    private static final long serialVersionUID = 1L;

    private final AtomicInteger idGenerator = new AtomicInteger(0);
    private final Set<ChatUser> connections = new HashSet<ChatUser>();

    @Override
    protected StreamInbound createWebSocketInbound(String subProtocol,
        HttpServletRequest request) {
        return new ChatUser(idGenerator.incrementAndGet());
    }

    private final class ChatUser extends MessageInbound {

        private final String username;

        private ChatUser(int id) {
            this.username = "guest #" + id;
        }
    }
}
```

The Servlet File

© 2010 Haim Michael. All Rights Reserved.

Sample

```
@Override
protected void onOpen(WsOutbound outbound) {
    connections.add(this);
    String message = username + " has joined the chat";
    broadcast(message);
}

@Override
protected void onClose(int status) {
    connections.remove(this);
    String message = username + " has left the chat";
    broadcast(message);
}

@Override
protected void onBinaryMessage(ByteBuffer message) throws IOException {
    throw new UnsupportedOperationException(
        "binary messages are not supported");
}
```

© 2010 Haim Michael. All Rights Reserved.

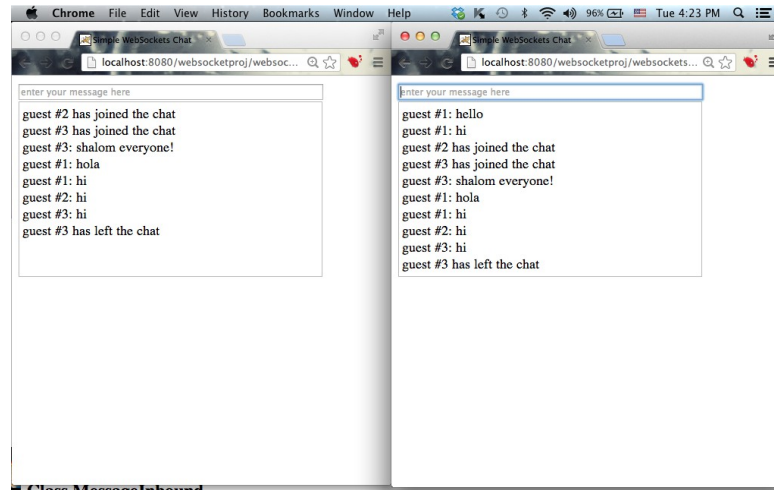
Sample

```
@Override
protected void onTextMessage(CharBuffer message) throws IOException {
    // it would be best filter the message
    // ...
    String str = username + ": " + message;
    broadcast(str);
}

private void broadcast(String message) {
    for (ChatUser connection : connections) {
        try {
            CharBuffer buffer = CharBuffer.wrap(message);
            connection.getWsOutbound().writeTextMessage(buffer);
        } catch (IOException ignore) {
            // ...
        }
    }
}
}
```

© 2010 Haim Michael. All Rights Reserved.

Sample



© 2010 Haim Michael. All Rights Reserved.