Topic: Websocket (Java EE 7 - WebSocket API & Tyrus).

1. Declare Websocket Server Endpoint with @ServerEndpoint annotation:

Step.1 Create a Java EE 7 project.

Step.2 Create a java class with @ServerEndpint annotation.

ChatServer.java

|  |
| --- |
| @ApplicationScoped  **@ServerEndpoint("/chat")**  public class ChatServer {  @Inject  private ChatSessionHandler sessionHandler;  **@OnOpen**  public void onOpen(Session session) {  sessionHandler.addSession(session);  }  **@OnClose**  public void onClose(Session session) {  sessionHandler.removeSession(session);  }  **@OnError**  public void onError(Throwable error) {  Logger.getLogger(ChatServer.class.getName()).log(Level.SEVERE, null, error);  }  **@OnMessage**  public void handleMessage(String message, Session session) {  sessionHandler.sendToAllConnectedSessions(session.getId()+": "+message);  } |

ChatSessionHandler.java

|  |
| --- |
| @ApplicationScoped  public class ChatSessionHandler {  private int deviceId = 0;  private final Set<Session> sessions = new HashSet<>();  public void addSession(Session session) {  sessions.add(session);  }  public void removeSession(Session session) {  sessions.remove(session);  }  public void sendToAllConnectedSessions(String message) {  for (Session session : sessions) {  sendToSession(session, message);  }  }  private void sendToSession(Session session, String message) {  try {  session.getBasicRemote().sendText(message);  } catch (IOException ex) {  sessions.remove(session);  }  }  } |

1. Web client with HTML5 - WebSocket

|  |
| --- |
| <html>  <head>  <title>TODO supply a title</title>  <meta charset="UTF-8">  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  <script language="javascript">  **var socket = new WebSocket("ws://192.168.2.54/ChatServer/chat");**  **socket.onmessage = onMessage;**  function onMessage(event) {  var chatForm = document.getElementById("chatForm");  var messages = chatForm.elements["messages"];  if (messages.value.length > 0) {  messages.value = messages.value + "\n" + event.data;  } else {  messages.value = event.data;  }  }  function sendMessage() {  var form = document.getElementById("chatForm");  var message = form.elements["message"].value;  **socket.send(message);**  form.elements["message"].value = '';  }  </script>  </head>  <body>  <form id="chatForm">  <input id="message" />  <input type="button" onclick="sendMessage()" value="Send"/><br />  <textarea id="messages" cols="80" rows="30"></textarea><br />  </form>  </body>  </html> |

1. Android client with Tyrus

MainActivity.java

|  |
| --- |
| **private Session session;**  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  connect();  final Button sendButton = (Button) findViewById(R.id.sendButton);  sendButton.setOnClickListener(new View.OnClickListener() {  public void onClick(View v) {  try {  TextView textView = (TextView) findViewById(R.id.message);  final String message = textView.getText().toString();  textView.setText("");  final AsyncTask<Void, Void, Void> asyncTask =  new AsyncTask<Void, Void, Void>() {  @Override  protected Void doInBackground(Void... voids) {  try {  **MainActivity.this.session.getBasicRemote().sendText(message);**  }catch (Exception e) {  e.printStackTrace();  }  return null;  }  };  asyncTask.execute();  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  private void connect() {  final AsyncTask<Void, Void, Void> asyncTask =  new AsyncTask<Void, Void, Void>() {  @Override  protected Void doInBackground(Void... voids) {  try {  **ClientManager clientManager = ClientManager.createClient();**  **clientManager.connectToServer(new Endpoint() {**  **@Override**  **public void onOpen(Session session, EndpointConfig EndpointConfig) {**  **MainActivity.this.session = session;**  **session.addMessageHandler(new MessageHandler.Whole<String>() {**  **@Override**  **public void onMessage(String message) {**  **changeMessages(message);**  **}**  **});**  **}**  **}, ClientEndpointConfig.Builder.create().build(),**  **URI.create("ws:// 192.168.2.54/ChatServer/chat"));**  } catch (DeploymentException e) {  e.printStackTrace();  } catch (IOException e) {  e.printStackTrace();  }  return null;  }  };  asyncTask.execute();  }  private void changeMessages(String message) {  TextView messages = (TextView) findViewById(R.id.messages);  messages.setText(messages.getText().toString().concat("\n").concat(message));  } |

Reference:

[**Java EE 7: Building Web Applications with WebSocket, JavaScript and HTML5**](http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/HomeWebsocket/WebsocketHome.html)

[**Whick Websocket library to use in android APP**](http://stackoverflow.com/questions/30547517/which-websocket-library-to-use-in-android-app)

[**Oracle - Tyrus**](https://tyrus.java.net/)