

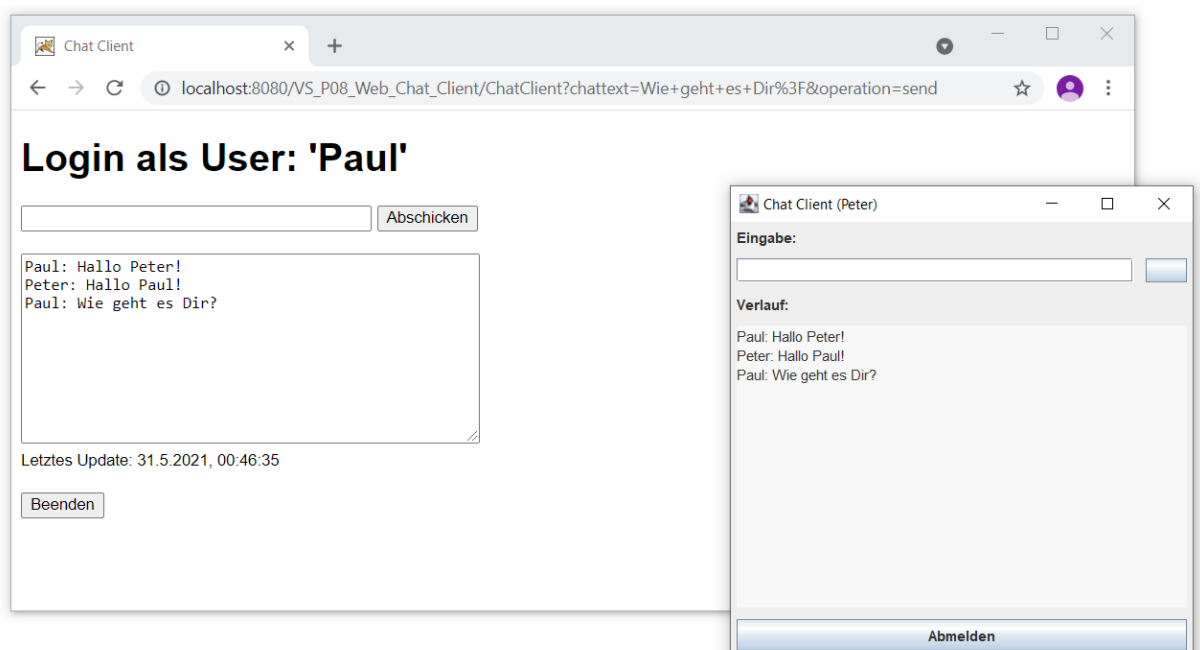
## Aufgabenblatt 7

### Aufgabe – Web-Client für das Chat-System

Implementieren Sie einen Web-Client für das RMI-basierte Chat-System des vorhergehenden Aufgabenblattes.

Folgende Anforderungen sind zu erfüllen:

- Die Realisierung soll per Servlets und statische Seiten (**jsp**- oder **html**) erfolgen.
- Der native RMI-Client aus dem vorhergehenden Aufgabenblatt soll weiterhin parallel genutzt werden können.
- Ein Teil der Web-Anwendung (nämlich der Java RMI-Client und die **rmiregistry**) laufen außerhalb des Web-Servers.
- Bei der Initiierung des Chats gibt der Benutzer seinen Namen über ein Formular ein, mit dem er beim Chat registriert wird. Unter diesem Namen wird eine **HttpSession** erzeugt, die beim Verlassen des Chats invalidiert wird. Parallel zum Erzeugen und Beenden der **HttpSession** wird der Client in der RMI-Registry ein- bzw. ausgetragen.
- Über ein Formular können Chat-Texte verschickt und die Nachrichten des Chats angezeigt werden. Über dasselbe Formular kann der Benutzer auch aus dem Chat ausgetragen werden.



- Im Fehlerfall (Server / Registry kann nicht erreicht werden, Kommunikationsfehler, Session abgelaufen, ... ) sollen entsprechende Seiten Aufschluss über die aufgetretenen Fehler geben.
- Erklären Sie im Praktikumsprotokoll Ihre Lösung anhand eines Paket-Diagramms.

## Hinweise:

Im Prinzip sind Sie frei, eigene Wege zur Lösung der Aufgabenstellung einzuschlagen. Damit die Aufgabe auch durch Studierende mit geringeren Vorkenntnissen im Bereich der Web-Programmierung behandelt werden kann, werden hier einige Hinweise gegeben.

- Da Sie an verschiedenen Stellen in der Web-Applikation auf die RMI-Komponenten zugreifen müssen, bietet es sich an, an einer Stelle die Verweise auf die entsprechenden Objekte der RMI-Kommunikation zu verwalten. Da die Funktionalität der Anwendung überschaubar ist, kann diese durch **ein einziges Servlet** realisiert werden.
- Unter dieser Prämisse, muss das Servlet auch **parametrisierbar** hinsichtlich der durchzuführenden Operationen (initialisieren, de-initialisieren, Text senden, Nachrichten anzeigen,...) sein. Diese Parametrisierung kann über die Parameter-Listen entsprechender **GET**-Requests vorgenommen werden. Hier ist die Nutzung sog. **hidden-Felder** in den Formularen sinnvoll. Bsp.:  
`<input type="hidden" name="operation" value="init">`
- Am komplexesten ist die Aktualisierung der Anzeige der empfangenen Chat-Nachrichten. Sie können hierzu die bei der Initiierung erzeugte **HttpSession** verwenden. Z.B. kann man in der **Session eine Liste der letzten n Nachrichten** speichern. Wird Client-seitig über **receiveMessage()** eine Nachricht über das Callback-Objekt empfangen, so wird diese in die Liste im Session-Objekt eingetragen. Beim Anzeigen des Formulars wird die Liste ausgelesen und in ein entsprechendes **html**-Element ausgegeben. Für die Ausgabe kann z.B. verwendet werden:  
`<textarea name="chatoutput" cols="50" rows="10" readonly>`
- Den jeweiligen Sessions zuzuordnen sind dabei auch die Verweise auf die Chat-Objekte.
- Achten Sie beim Test auf die unterschiedlichen Möglichkeiten der Realisierung der Session. Wollen Sie von *einem* Browser aus mit mehreren Client-Sitzungen die Applikation testen, dann müssen Sie **URL-Rewriting** unterstützen. Bei der Verwendung von **Cookies** werden die UI-Instanzen nicht in Form von unterschiedlichen Sessions unterschieden.
- Die Anzeige der empfangenen Nachrichten muss ständig aktualisiert werden. Es gibt zahlreiche Wege dies zu tun. Die einfachste Möglichkeit ist, in **zyklischen Intervallen (Polling)** durch Absetzen eines entsprechenden **GET**-Requests, die Seite einfach neu zu laden :  
`<meta http-equiv="refresh" content="10";  
URL=ChatClient?name=heiyo&operation=show">`
- Eine umfangreiche Sammlung von einsetzbaren **html**-Elementen finden Sie unter:  
`http://de.selfhtml.org/`

Für das Testat ist ein Protokoll bekannter Formatierung (siehe Blatt 1) inkl. der durchgeführten Tests vorzulegen. Das Protokoll zu dieser Aufgabe soll neben der Beschreibung der Umsetzung und der durchgeführten Tests insbesondere ein Paketdiagramm der Lösung beinhalten.

**Testierung:** 1./2./3.6.2022