

Aufgabenblatt K1

Webtechnologien Wirtschaftsinformatik SS 2020

Lernziele

Es wird folgendes gelernt:

- Realisierung einer Single-Page-Anwendung mittels JavaScript
- Ajax
- Dropdown Menue

JavaScript, Ajax und <div> - 6 Punkte

Nun wird ein Chat-System realisiert. Der Server kann maximal drei Chat-Räume behandeln; diese haben einen festen Namen. Zu jedem Raum sind maximal drei Teilnehmer zulässig.

Das Ganze verläuft wie folgt: Auf der Startseite wird der Name des Raums ausgewählt und der eigene Name eingegeben. Diese Daten werden an den Server gesandt, der mit der Chat-Seite antwortet. Diese besteht aus drei Textfeldern mit jeweils einer Überschrift. Die Überschrift ist der Name der Person, das darunter stehende Textfeld dessen eingegebener Text. Dies wiederholt sich für die 2. und 3. Person, wobei hier nur die Texte ausgegeben werden. Wenn nun irgendetwas in dieses Feld eingetippt wird, dann erscheint es bei den beiden anderen Kommunikationspartnern jeweils unter dem Namen des Schreibenden und in dem dazugehörigen Textfeld. So, das ist die Idee.

Das Ganze wird durch 2 Seiten realisiert: die erste ist die "Login"-Seite mit der Chat-Raum-Auswahl und dem eigenen Namen und die zweite ist die "Chat"-Seite. Diese 2. Seite wird vollständig mit Ajax implementiert, d.h. es gibt kein Nachladen der Seite vom Server. Bitte beachten Sie, dass Sie auch hier ein minimales Session-Management implementieren müssen; dazu vergeben Sie eine Nummer, die bei jedem Ajax-Request mitgesendet wird. Auf der Server-Seite wird diese ID als Teil eines bestimmten Chats vermerkt.

Und nun die einzelnen Schritte:

1. Sie entwerfen die beiden Seiten mit den Layout-Angaben über Layer und einer externen CSS-Datei.
2. Jetzt konzipieren Sie die Protokolle: wer mit welcher ID was zum Server schickt und was als Antwort erhält.
3. Nun implementieren Sie eine allgemeine Ajax-Routine – als Objekt oder als Sammlung von Routinen. Als Antwort-Format vom Server benutzen Sie nun JSON (bei der Ziegenaufgabe war es ja ein einfacher String). Sie definieren dazu die Felder der JSON-Objekte. Die Ajax-Nachrichten dürfen nicht alle Zeichen enthalten; hier muss also nach URLEncoded umgewandelt werden.
4. Dann realisieren Sie die Routinen zum Beschreiben und Auslesen der Textfelder, wobei hier die Namen der Felder (DOM) als Parameter übergeben werden.
5. Jetzt entscheiden Sie wie ein Refresh der Chat-Seite gemacht wird: Am Einfachsten wird dafür ein Knopf vorgesehen. Wenn jemand etwas getippt hat, klickt er auf diesen Knopf und sein Textfeld wird zum Server geschickt, der mit den aktuellen Textfeldinhalten der anderen beiden Kommunikationspartner antwortet, deren Texte dann natürlich dargestellt werden. Also nach dem Klick werden die Daten ausgetauscht.
6. Diese einfache Lösung wird durch die Folgende ersetzt: Per Timer wird der aktuelle Inhalt alle 1-2s zum Server übertragen. Oder noch komfortabler: Sie können eine JavaScript-Routine an ein Tastatur-Ereignis hängen, so dass bei jedem Tastendruck die Kommunikation zum Server abläuft, was echtes Chat ist, aber auch sehr aufwändig ist. Da ja jemand einmal nichts eingeben kann, also schweigt, kann dies mit der Timeout-Variante kombiniert werden: Wird getippt, gibt es kein Timeout, wird nicht getippt, kommt nach 1-2s ein Timeout und damit ein Aktualisieren.
7. [Abgabe] Es fehlt jetzt nur noch ein "Logout"-Knopf auf der "Chat"-Seite, der an den Server eine "Dann bin ich mal weg"-Nachricht schickt (siehe Aufgabe2: Protokoll). Entsprechend gibt es ja auch die "Login"-Nachricht. Bitte beachten Sie, dass nicht immer alle möglichen Personen chatten und dass manchmal als mehr als drei Personen chatten wollen. Sie müssen dann die Abwesenheit eines Partners behandeln.

Das ist jetzt schon eine richtige, auch etwas komplexe Web-Anwendung. Wer daraus etwas Dolles machen will, muss nun noch so etwas wie Passwörter vorsehen, vergangene Chats in einer Datenbank abspeichern, um sie später wieder an der alten Stelle fortführen zu können und natürlich beliebig viele Chats gleichzeitig erlauben etc.

Abnahme und Bewertung

Das Ergebnis der Aufgabe K1 wird per EMail bei Herrn Sabbagh abgegeben. Dabei muss die gesamte Site als zip-Datei, so wie sie unter htdocs liegt, abgegeben werden, weiterhin gehört definitiv noch das netbeans/eclipse-Projekt als zip-Datei dazu, so dass es unter XAMPP (Windows oder Linux) sofort läuft.

Es sind max. 6 Punkte zu erreichen; fehlt etwas oder ist es nicht in Ordnung, so werden Punkte abgezogen bzw. die Lösung nicht anerkannt.

Es müssen die in den vorherigen Aufgabenblättern vorgestellten Regeln über CSS, Layout und HTML4/5 eingehalten werden.