

Aufgabenblatt 2 – Recherchebericht

Projekt: Keynode 11
Gruppe: swp11-5
Projektleiter: Florian Golemo
Rechercheleiter: Manuel Konrad

Inhaltsverzeichnis

1	Begriffe.....	3
1.1	HTML/XHTML.....	3
1.2	Slide.....	3
1.3	Foliensatz.....	3
1.4	Websocket.....	3
1.5	Push.....	3
1.6	Template.....	4
1.7	Framework.....	4
1.8	Javascript.....	4
1.9	Secure Shell.....	4
1.10	Browser.....	4
2	Konzepte.....	5
2.1	Slidy.....	5
2.2	node.js.....	5
2.3	socket.IO.....	5
3	Zu studierende Aspekte.....	6
3.1	Installation socket-chat.....	6
3.2	Browser-Tests des socket-chats.....	7
3.3	Architektur des Chat-Systems.....	7
3.4	Funktionsweise Slidy.....	8

1 Begriffe

1.1 HTML/XHTML

HTML steht für Hypertext Markup Language (zu deutsch Hypertext-Auszeichnungssprache). Sie ist eine vom W3C entwickelte Sprache und dient der strukturierten Darstellung von Texten, Bildern, u.ä. in den gleichnamigen HTML-Dokumenten, die das Grundgerüst einer jeden Website bilden. Dabei werden die HTML-Befehle vom Webbrowser interpretiert, der daraus die das Erscheinungsbild der Internetseite erstellt.

XHTML (Extensible HyperText Markup Language) ist eine Anwendung von XML auf HTML 4.01. Dies hat zur Folge, dass im Unterschied zum normalen HTML die syntaktischen Grundregeln der Sprache strenger eingehalten werden müssen, so wird z.B. die Wohlgeformtheit eines Dokuments verlangt, d.h. dass jedes geöffnete HTML-Element durch ein gleichnamiges, schließendes beendet werden muss, was in HTML keine Probleme darstellte.

Diese beiden Konzepte sollen durch das sich zur Zeit in der Entwicklung befindliche HTML5 vollständig abgelöst werden, welches durch viele neue Funktionen einen gewaltigen Sprung zu den vorherigen Versionen darstellt. Zu diesen Funktionen zählen u.a. die Möglichkeit Websockets zu nutzen, die für dieses Projekt essentiell sind, um Aktionen, wie das Weiterblättern des Präsentierenden auf den Computern der Zuschauer auszuführen.

1.2 Slide

Slide ist das englische Wort für Folie und bezeichnet eine einzelne Seite einer Präsentation. Er kann mit verschiedenen Inhalten, wie Text, Grafiken und Audiodateien gefüllt und in manchen Programmen (PowerPoint, Impress) durch spezielle Effekte, wie z.B. das Einfliegen eines Textes, erweitert werden.

1.3 Foliensatz

Ein Foliensatz ist eine Ansammlung einzelner Slides. Die Auswahl der einzelnen Folien ist dabei nicht auf einen bestimmten Typen beschränkt, so können verschiedene Folienarten, wie z.B. Titel- oder Gliederungsfolie darin enthalten sein.

Im Zusammenhang mit diesem Projekt befindet sich ein Foliensatz in einer HTML-Datei.

1.4 Websocket

Der Begriff Websocket beschreibt eine bidirektionale Verbindung einer Webanwendung zu einem Server. Der Unterschied zu einer HTTP-Verbindung (wird z.B. beim Aufruf einer Internetseite, deren Adresse mit „http“ beginnt, benutzt) besteht darin, dass das Websocket direkt vom Server verwaltet wird und dementsprechend Daten direkt, ohne Anfrage des Clients an diesen senden kann (Push).

1.5 Push

Im Internet ist in vielen Teilen ein Pull-Medium, weil der Nutzer direkt nach bestimmten Informationen fragt (Pull) und diese von einem Server geliefert wird (z.B. Ansteuern einer Internetseite).

Unter Push, oder auch Server-Push, versteht man das Senden von Daten vom Server an den Empfänger, ohne eine direkte Anfrage von diesem erhalten zu haben. Als Beispiel kann man ein

auf Websockets basierendes Chatsystem betrachten. Nutzer1 und Nutzer2 sind angemeldet. Nutzer1 sendet eine Nachricht. Diese wird an den Server übermittelt, der die Nachricht direkt an Nutzer2 weiterleitet (pushed). Sofort wird diese Nachricht bei Nutzer2 angezeigt.

1.6 Template

Ein Template (zu deutsch Schablone) ist eine Vorlage, die mit relevanten Informationen gefüllt werden kann.

Im Zusammenhang mit diesem Projekt handelt es sich bei Templates um Vorlagen für einzelne Präsentationsfolientypen wie z.B. Titel-, Gliederungs- oder Informationsfolie.

1.7 Framework

Ein Framework ist eine auf einer bereits existierenden Programmiersprache basierende Sammlung von Methoden, Entwurfsmustern, Komponenten, u.ä., das zur schnelleren Entwicklung einer Anwendung beitragen soll.

Neben diesen Eigenschaften weist ein Framework oftmals leichte Änderungen oder Ergänzungen der Syntax der zu Grunde liegenden Programmiersprache auf.

1.8 Javascript

Bei Javascript handelt es sich um eine Scriptsprache, die meist im Browser und, anders als z.B. PHP, auf Anwenderseite ausgeführt wird. Das hat auf der einen Seite den großen Vorteil, dass der Server durch diese Skripte nicht zusätzlich belastet wird, auf der anderen Seite ist die Leistung abhängig von der Rechenleistung des Clientrechners, was bei ungünstigen Bedingungen eine schlechte Performance der Internetseite aufweisen kann.

Dafür bietet Javascript viele Funktionen mit denen dynamische Änderungen an Internetseiten, wie z.B. das Einfärben von Elementen, ohne wiederholtes Laden der gesamten Seite möglich ist.

1.9 Secure Shell

Unter Secure Shell, oder kurz SSH, versteht man ein Netzwerkprotokoll mit dem es möglich ist eine verschlüsselte Verbindung über ein unsicheres Netzwerk zu einem entfernten Gerät aufzubauen. Bei diesen Geräten handelt es sich meist um Server, bei denen eine SSH-Verbindung oft dazu genutzt wird eine sich auf dem Server befindliche Kommandozeile lokal zu nutzen und damit den entfernten Computer zu verwalten. Es stellt damit eine Möglichkeit zur konsolengestützten Fernwartung von Computern dar.

1.10 Browser

Ein Browser, oder auch Webbrowser, ist ein Programm, das meist zur Fortbewegung im Internet und der Darstellung von Internetseiten genutzt wird.

Er beinhaltet dabei eine Engine zum Anzeigen der Webinhalte (Layout-Engine), sowie eine zum Ausführen von Javascript-Code. Da die verschiedenen Browser meist unterschiedliche Engines nutzen und die eine oder andere manche Funktionalitäten nicht unterstützt, kann es von Webbrowser zu Webbrowser zu unterschiedlichen Darstellungen gleicher Internetseiten oder anderen Fehlern kommen, weshalb von Web-Entwicklern bei der Erstellung von Webseiten auf die Eigenarten der einzelnen Engines besonders geachtet und darauf eingegangen werden muss.

Zu den bekanntesten Vertretern zählen Mozilla Firefox, Google Chrome, Microsoft Internet Explorer, Safari und Opera.

2 Konzepte

2.1 Slidy

Slidy wurde von der W3C entwickelt und dient der Darstellung von Präsentationen (engl. Slideshow) im Webbrowser.

Zur Anzeige der Inhalte sind die Dokumente in XHTML/HTML geschrieben, wobei detaillierte Formatierungsinformationen, wie z.B. farbliche Gestaltungen von Texten, in CSS-Dateien ausgelagert sind. Dabei stellt ein XHTML-Dokument genau eine Präsentation dar. Die Nutzung dieser Sprachen hat dabei den positiven Nebeneffekt, dass, anders als in Präsentationswerkzeugen, wie PowerPoint oder Impress, eine Bearbeitung der Foliensätze direkt in einem Texteditor möglich ist, man also nicht auf solche Programme angewiesen ist und völlig plattformunabhängig arbeiten kann.

Zusätzlich zur Darstellung wurden die Basisfunktionalitäten für eine Slideshow mittels Javascript implementiert. So ist es möglich durch einzelne Folien zu blättern, zur ersten und letzten Folie zu springen, die Fußzeile ein- und auszublenden und einen Vollbildmodus zu aktivieren.

Insgesamt bildet Slidy für dieses Projekt das Grundgerüst für die eigentlich laufende Präsentation.

2.2 node.js

node.js ist ein Framework, das es ermöglicht Javascript-Code auf einem Server auszuführen. Es setzt auf der von Google entwickelten V8-Javascript-Engine auf, die derzeit die schnellste Javascript-Implementierung auf dem Markt ist, was u.a. daran liegt, dass der Code vor der Ausführung in Maschinensprache umgewandelt wird.

Eine weitere Besonderheit von node.js ist die Nutzung des Konzepts der „non-blocking I/O“. Im Gegensatz zu anderen Sprachen, wo jede Ein- und Ausgabe (Zugriffe auf Netzwerk und Festplatte) die Abarbeitung der nachfolgenden Programmzeilen bis zum vollständigen Abschließen der Aktionen blockierte und die Programme damit viel Geschwindigkeit einbüßen, da diese Zugriffe wesentlich langsamer sind, als CPU- oder RAM-Operationen, setzt die nicht-blockierende Ein-/Ausgabe auf Callback-Funktionen, die als Parameter in I/O-Funktionen angegeben werden. Das bedeutet, dass während der Ausführung von Ein-/Ausgabe-Operationen der eigentliche Programmablauf nicht unterbrochen wird, da die I/O-Funktionen im Hintergrund ablaufen und nach der vollständigen Abarbeitung die angegebene Callback-Funktion ausführen.

Damit arbeitet node.js extrem schnell und da Javascript in nur einem Thread ausgeführt wird, muss man sich als Entwickler keine Gedanken über sich überschneidende Zugriffe auf Objekte, o.ä. machen.

Der von uns verwendete node.js-Server wird deshalb die Zugriffe der socket.IO verwalten, die gesendeten Daten verarbeiten und die jeweiligen Aktionen an die angemeldeten Clients pushen, wie z.B. das Weiterblättern einer Folie, das durch den Präsentierenden ausgeführt wurde.

2.3 socket.IO

socket.IO ist eine Javascript-Bibliothek, die es ermöglicht Echtzeit-Web-Anwendungen in jedem Browser (Firefox, Internet Explorer, ...) zu erstellen. Dabei bestimmt socket.IO die Art und Weise der Verbindung, abhängig vom jeweiligen Browser, selbständig (z.B. über Websockets).

Vor der Nutzung muss socket.IO auf dem Server installiert werden. Beim Server selbst handelt es

sich um einen node-Server, der für die gesamte Kommunikation verwendet wird. Socket.IO ist deshalb stark an das node.js-Framework gebunden.

Auf Clientseite müssen lediglich die socket.io.js-Datei eingebunden und ein paar Zeilen Javascript-Code ausgeführt werden:

```
<script src="http://{node_server_url}/socket.io/socket.io.js"></script>
<script>
  var socket = new io.Socket({node_server_url});
  socket.connect();
  socket.on('connect', function(){ ... })
  socket.on('message', function(){ ... })
  socket.on('disconnect', function(){ ... })
</script>
```

Danach ist eine Kommunikation zwischen Client und Server über die verschiedenen von socket.IO gelieferten Befehle möglich.

In diesem Projekt wird socket.IO für die Synchronisierung der angezeigten Präsentationsinhalte und -aktionen zwischen Vortragendem und Zuhörern genutzt (z.B. Folienwechsel durch den Vortragenden hat einen Wechsel auf die gleiche Folie bei allen Zuhörenden zur Folge).

3 Zu studierende Aspekte

3.1 Installation socket-chat

1. Die unter <https://github.com/arunjitsingh/socket-chat> gelisteten Dateien downloaden, entpacken und auf den Zielsever hochladen
2. Da der Chat node, sowie socket.IO nutzt, müssen diese Komponenten auf dem Zielsever installiert werden. Dies geschieht über folgende UNIX-Befehle:
 - a) `git clone git://github.com/LearnBoost/Socket.IO-node.git bin/socket.io -recursive` (Installation von Socket.IO-node)
 - b) `git clone git://github.com/bentomas/node-mime.git bin/mime -recursive` (Installation von node-mime)
3. In nachfolgenden Zeilen in den Dateien app.js und index.html die Ports des Chats anpassen:
 - a) app.js – Zeile 133 - `app.listen(4150);`
 - b) index.html – Zeile 107 - `PORT = 4150;`
4. Host in der Datei index.html anpassen:
 - a) index.html – Zeile 107 - `const HOST = pcai042.informatik.uni-leipzig.de`
5. Pfad zu den genutzten Frameworks in der Datei app.js ändern:
 - a) app.js – Zeile 4 - `var io = require('/home/swp11-5/bin/socket.io-node/lib/socket.io');`
 - b) app.js – Zeile 12- `mime = require('/home/swp11-5/bin/mime');`
6. Um zu Verhindern, dass der sich Server beim schließen der SSH-Konsole beendet, muss

eine virtuelle Shell Console (mit Screen) erstellt werden.

7. Letztendlich kann der Node-Server durch Aufrufen der Datei „./node public_html/SocketChat/app.js“ ausgeführt werden

3.2 Browser-Tests des socket-chats

In nachfolgenden Tabelle ist eine Auflistung über die verschiedenen Browser zu finden und ob und wie das Chat-System im jeweiligen Programm funktioniert hat:

Browser	Eingabe Nickname	Nachrichten senden/ empfangen	Kommentar/Besonderheiten/ Sonstiges
Google Chrome 10.0	ja	ja	Komplett funktionstüchtig
Internet Explorer 9.0	nein	nein	Es erscheint kein Anmeldefenster → Benutzername kann nicht angegeben werden; Keine Reaktion beim Absenden von Nachrichten
Firefox 4.0	ja	nein	Anmeldefenster erscheint; Anmeldung möglich; Nachrichten können gesendet werden und werden auch angezeigt, aber werden nicht an andere Teilnehmer gesendet; Nachrichten von anderen Teilnehmern werden nicht empfangen
Opera 11.10	ja	nein	Anmeldefenster erscheint; Anmeldung möglich; Nachrichten können gesendet werden und werden auch angezeigt, aber werden nicht an andere Teilnehmer gesendet; Nachrichten von anderen Teilnehmern werden nicht empfangen
Safari 5.0	ja	ja	Komplett funktionstüchtig

3.3 Architektur des Chat-Systems

Die Architektur des Chat-Systems basiert auf einer Client-Server-Struktur, wobei es sich bei dem Server um einen node-Server handelt, der auf dem node.js-Framework basiert. Jede Verbindung wird über die socket.IO-Bibliothek erstellt und verwaltet.

Der Server wird dabei an einen Port gebunden. Danach wird der Listener über den Befehl (`io = io.listen(httpServer)`) aktiviert. Dieser „belauscht“ den Server und die eingehenden Ereignisse, wie das Senden einer Nachricht von einem Client an den Server und reagiert entsprechend darauf.

Zu den Basisfunktionen, die der Chat benötigt zählen:

1. `io.on('connection', function(client){...})` - Funktion, die ausgeführt wird, wenn sich ein Client mit dem Server verbindet. Dabei hat jeder Client eine eigene Callback-Funktion.

2. `client.on('message', function(...){...})` - Methode wird ausgeführt, wenn ein Client eine Nachricht abschickt.
3. `client.broadcast(message)` - Mittels dieser Funktion wird die vom Client abgeschickte Nachricht an den Server geleitet, der diese an die anderen angemeldeten Nutzer sendet.
4. `client.on('disconnect', function(...){...})` - Funktion, die ausgeführt wird, wenn sich Client vom Chat abmeldet.

3.4 Funktionsweise Slidy

Slidy ist ein sehr kompaktes Javascript-Framework, das erlaubt, auf einfache und standardkonforme Art und Weise webbasierte Präsentationen zu erstellen.

Der Inhalt der Präsentation wird dabei in HTML und CSS definiert, wahlweise mit eigenem Javascriptcode

Grundsätzlich ist für einfache Präsentationen gar kein eigener Javascriptcode erforderlich, denn selbst dynamische Inhalte in der Präsentation können durch normales Markup, in erster Linie durch spezielle CSS-Klassen und Metatags definiert werden.

Um Slidy in das HTML einzubinden sind zwei Zeilen nötig: Das Includen der Javascript-Bibliothek und des Slidy-CSS-Sheets. Eigene Inhalte können beliebig hinzugefügt werden.

Alle Folien sind in einer HTML-Datei gespeichert, Slidy zeigt aber nur diejenige, die über die Adresszeile angegeben wurde, bildschirmfüllend an. Die Größe der Schriftart wird dabei automatisch entsprechend der Bildschirmauflösung bestimmt.

Slidy simuliert das typische Bedienungsverhalten mit den bekannten Tastenkürzeln bestehender Präsentationsprogramme.

Dadurch entsprechen die mit Slidy erstellten Präsentationen völlig normalen, standardisierten Webseiten, die auf allen gängigen Browsern funktionieren.

Da sämtliche Funktionalitäten clientseitig ausgeführt werden, ist es auch problemlos möglich, die Präsentationen offline anzuschauen.

Die Initialisierung der Slidy-Bibliothek geschieht beim Laden der Seite automatisch, sodass nicht einmal Initialisierungs-Code erforderlich ist.