

Inhalt

Vorwort	IX
Danksagung	XI
1 Einleitung.....	1
1.1 Wie ist das Buch strukturiert?	3
1.2 Wer sollte das Buch lesen?	3
1.3 Wie sollte mit dem Buch gearbeitet werden?	4
2 HTTP: Hypertext Transfer Protocol	5
2.1 Grundlegendes.....	5
2.1.1 ISO/OSI-Referenzmodell für Kommunikationssysteme.....	6
2.1.2 Vereinfachtes OSI-Modell	8
2.1.3 Dienst	9
2.1.4 Protokoll.....	10
2.1.5 Kommunikationsfluss	11
2.2 HTTP en détail	13
2.2.1 Kommunikationsablauf	13
2.2.2 Textbasiert.....	14
2.2.3 Aufbau von Request-Nachrichten und Protokollelemente	16
2.2.4 Aufbau von Response-Nachrichten und Statuscodes	20
2.2.5 Zustandslos	24
3 Höhere Interaktivität und Echtzeitfähigkeit	25
3.1 XMLHttpRequest (XHR)	25
3.2 Polling	27
3.3 Long-Polling	28
3.4 Comet	29
3.5 Server-Sent Events	31
3.6 Bewertung der Verfahren	33

4	Die Leitung: Das IETF WebSocket-Protokoll	35
4.1	Was bisher geschah	35
4.2	Opening-Handshake – WebSocket-Verbindung aufbauen	36
4.3	WebSocket-Frames	39
4.4	Fragmentierung	42
4.5	Maskierung	43
4.6	Datenframes	45
4.6.1	Frames mit Textdaten	45
4.6.2	Frames mit Binärdaten	47
4.7	Control-Frames	48
4.7.1	Ping-Frame	48
4.7.2	Pong-Frame	49
4.7.3	Close-Frame	49
4.8	Closing-Handshake – WebSocket-Verbindung schließen	53
4.9	Tools zur Protokollanalyse	54
4.9.1	Wireshark	54
4.9.2	Fiddler	60
4.9.3	Chrome Developer Tools	68
4.9.4	Chrome Network Internals	70
5	Der Client: Die W3C WebSocket-API	75
5.1	Was bisher geschah	75
5.2	Browserunterstützung	76
5.3	Namensschema	79
5.4	Objekterzeugung und Verbindungsaufbau	80
5.5	Zustände	81
5.6	Event-Handler	83
5.7	Erstes vollständiges Programmchen	86
5.8	Attribute	89
5.9	Datenübertragung	92
5.9.1	Übertragung textbasierter Daten	93
5.9.2	Übertragung binärer Daten	94
5.10	Verbindungsabbau	96
5.10.1	Verbindung beenden	96
5.10.2	Close-Event	96
5.11	Ausblick: HTTP 2.0/SPDY	97

6	Der Server: Sprachliche Vielfalt	101
6.1	Übersicht verfügbarer WebSocket-Implementierungen	101
6.2	Node.js	102
6.2.1	Installation von Node.js	104
6.2.2	WebSocket.io	106
6.2.3	Socket.io	109
6.2.4	WebSocket-Node	113
6.3	Vert.x	119
6.4	Play Framework	122
6.4.1	Installation	122
6.4.2	Anlegen eines neuen Play-Projekts	123
6.4.3	Anatomie einer Play-Anwendung	123
6.4.4	Die Play-Konsole	124
6.4.5	Controller in Play	124
6.4.6	Views in Play	125
6.4.7	Routes in Play	126
6.4.8	Vom Controller zur View	127
6.4.9	Erstellen eines Echo-Servers in Play	129
6.5	JSR 356	132
6.5.1	JSR 356-basierender WebSocket-Server	136
6.5.2	JSR 356-basierender WebSocket-Client	140
7	WebSockets in der Praxis	147
7.1	Performance und Skalierbarkeit	147
7.1.1	Test-Echo-Server	148
7.1.2	Testclient für die Auslastung	151
7.1.3	Testclient für die Zeitmessung	152
7.1.4	Einrichtung der Testumgebung	153
7.1.5	Ergebnisse	157
7.2	Sicherheit	158
7.2.1	Same Origin Policy	158
7.2.2	Vertrauliche Kommunikation	159
7.2.3	Authentifizierung	162
7.2.4	Firewalls und Proxys	183
7.2.5	Mögliche Gefährdungen	186
7.2.6	Bewertung der Sicherheitslage	195
7.3	Mit Haken und Ösen	196
7.3.1	Pings und Pongs	196
7.3.2	Das Cache-Problem im Internet Explorer 10	197

8	Beispielanwendungen	201
8.1	Fernbedienung von Webanwendungen mit einer Smartphone-/Tablet-Fernbedienung	202
8.2	Chat-System	211
8.3	Heatmap für Usability-Tests	216
8.4	Überwachungskamera per Webcam	221
	Schlusswort	231
A	XHR-Objekt	233
B	Chrome Developer Tools auf Android	235
C	Umgebungsvariablen definieren	241
C.1	Mac OS/Linux mit Bash	241
C.2	Windows	242
D	Express.js	247
D.1	Anatomie einer Express.js-Anwendung	248
D.2	Die Anwendungslogik in der Datei app.js	250
D.3	Die Jade-Template-Engine	252
D.4	Express.js mit WebSocket-Servern verbinden	254
	Literatur	257
	Stichwortverzeichnis	263