1 Einleitung

Einer der technischen Hauptakteure im Web ist HTTP, das *Hypertext Transfer Protocol* [FGM+99], und das ist so, weil HTTP den genauen Kommunikationsablauf zwischen Webbrowser und Webserver festlegt. Genau wie ein Protokoll für eine königliche Gala zu Hofe alle Regeln in Bezug auf Kleidung, Etikette und Umgangsformen zusammenfasst, so legt ein Kommunikationsprotokoll alle Regeln zum Nachrichtenaustausch fest. Anders als bei Hofe sind das bei einem Kommunikationsprotokoll aber vielmehr Regeln in Bezug auf den Aufbau, das Format und die Codierung der ausgetauschten Nachrichten. Außerdem muss in einem Protokoll festgelegt sein, wer die Kommunikation beginnt und wie diese dann Nachricht für Nachricht bis zum Kommunikationsende abläuft. HTTP verwendet hier einen einfachen Request-Response-Nachrichtenaustausch. Dieser wird immer vom Webclient durch eine Anfrage (engl. *Request*) initiiert und entsprechend vom Webserver mit einer Antwort (engl. *Response*) beantwortet. Sobald Sie also z. B. eine Webadresse – die URL – in die Adresszeile eines Webbrowsers eingeben oder einen Bookmark bzw. Link anklicken, setzen Sie damit eine Anfrage an einen Server ab, der diese dann bearbeitet und die Antwort daraufhin zurücksendet (siehe Bild 1.1).

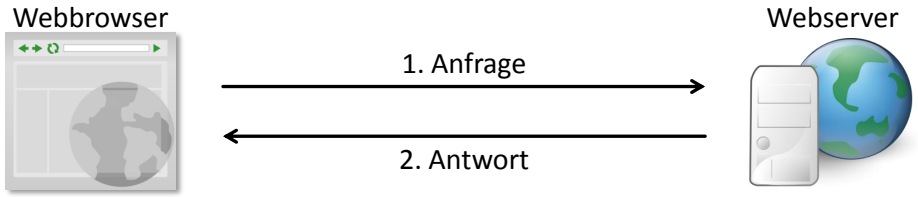

Bild 1.1 Grundlegender HTTP-Nachrichtenaustausch

In diesem einfachen Request-Response-Nachrichtenaustausch liegen viele Gründe für den großen Erfolg von HTTP und nicht zuletzt des Webs. Dazu zählen u. a. die guten Skalierungseigenschaften, denen wir die Größe des Webs verdanken. Wie Sie sich aber sicher jetzt schon denken, bringt das Nachrichtenpaar nicht nur Vorteile mit sich. Und tatsächlich ist die Protokollregel, dass der Request immer vom Client ausgehen muss, eine Zwangsjacke, die bestimmte Bewegungsfreiheiten in Form von Kommunikationsmustern nicht zulässt. Der Teil einer Webanwendung, der auf dem Server ausgeführt wird, kann sich nämlich nicht von sich aus an einen Client wenden. Diese Eigenschaft des Protokolls schränkt folglich Anwendungsfälle ein, in denen die Serverseite Statusänderungen an die betroffenen Clients weiterreichen können muss. Typische Beispiele hierfür sind Chatanwendun-

gen, Finanzdatenströme, Webmail, Sportticker, das Monitoring von Haushalt und Industrie oder auch Internetauktionen. All diese Anwendungen teilen die Gemeinsamkeit, dass (häufig in relativ kurzen Zeitabschnitten) Daten auf dem Server hinzukommen bzw. sich ändern und dies unmittelbar an die angeschlossenen Clients bekannt gegeben werden muss. Liegt Ihnen eine derartige Anforderung auf dem Tisch, dann würden Sie sich eine standardisierte Technik wünschen, mit der Sie die Schranken der Serverseite öffnen und die entsprechenden Clients ansprechen können.

HTTP bietet für derartige Kommunikationsmuster keine native Unterstützung. Wenn Sie sich in den letzten Jahren aus dieser Zwangsjacke hätten befreien und die Kommunikationseinbahnstraße zur Umsetzung von Serverbenachrichtigungen hätten aufbrechen wollen, so wären Sie auf Grundlage des Verfügbaren auf eine (selbst) konstruierte Lösung angewiesen gewesen. Die Konstruktionen, die findige Webentwickler dabei gefunden haben, sind vielfältig. Prinzipiell können Sie den Standardfunktionsumfang durch Browser-Plug-ins und serverseitige Erweiterungen aufbohren und damit proprietäre Lösungen zur Serverbenachrichtigung implementieren. Der erste Ansatz überhaupt kam tatsächlich auch aus dieser Technologiegattung und bediente sich Java Applets und der LiveConnect-Schnittstelle zur Anbindung an JavaScript-Funktionen im Browser. Seit HTML5 besteht allerdings ein starker Trend weg von Browsererweiterungen à la Plug-in oder Add-on. Schützenhilfe kam zudem aus dem Umfeld der Smartphones und Tablets, das sich gegen eine Unterstützung von Flash und Silverlight entschieden hat, was schließlich selbst Adobe und Microsoft dazu bewegt hat, die Entwicklungen ihrer Technologien für mobile Plattformen einzustellen. Daher wollen wir auch nicht weiter auf das Auslaufmodell der Plug-in-basierten Ansätze eingehen. Weitere Konstruktionen basieren auf Ansätzen, die den Webbrowser in regelmäßigen Zeitabständen beim Webserver nach Veränderungen fragen lassen. Diese aktuell weit verbreiteten Verfahren haben Nachteile, was das Kommunikationsaufkommen anbelangt. Dennoch haben sie unter Umständen ihre Berechtigung. Sie lernen diese Ansätze daher kennen und insbesondere diese zu bewerten, um die richtige Technologie für Ihr Entwicklungsvorhaben auswählen zu können. Nichtsdestotrotz handelt es sich aber weitestgehend um Notlösungen, die durch die in der HTML5-Standardisierung befindlichen alternativen Lösungswege abgelöst werden. Dazu gehören zum einen *Server-Sent Events* (SSE) [Hic12a] und zum anderen *WebSockets* (WS) [FM11a]. Die verschiedenen Bestandteile des Standards, die HTML5 einführt, sind thematisch gruppiert und mit einem Logo versehen worden [W3C14]. SSE und WS sind in der sogenannten Connectivity-Gruppe. Das Logo sehen Sie in einer leicht abgewandelten Form in Bild 1.2.

Bild 1.2 Das Logo für die Connectivity-Standards Server-Sent Events und WebSockets, platziert auf dem HTML5-Schild [W3C14]

WebSockets stehen im Fokus dieses Buches, da Sie damit neben den Serverbenachrichtigungen eine Vielzahl weiterer Anwendungsfälle realisieren können, zu denen z. B. Spiele, jegliche Art von Konversationen (Text, Sprache, Sprache mit zusätzlichem Video), Kollaborationen sowie das Benutzermonitoring im Rahmen von Usabilitystudien zählen.

1.1 Wie ist das Buch strukturiert?

WebSockets und die damit einhergehenden neuen Entwicklungsmöglichkeiten stehen somit im Zentrum des vorliegenden Buches und an diese wollen wir uns mit Ihnen Schritt für Schritt ranpirschen. In Kapitel 2 führen wir uns dazu zunächst die notwendigen Grundlagen in Bezug auf HTTP zu Gemüte. Kennen Sie diese schon, können Sie getrost die Abkürzung zu Kapitel 3 nehmen, in dem wir uns mit den Mechanismen befassen, mit denen eine höhere Interaktivität und Echtzeitfähigkeit bei Webanwendungen erreicht werden kann. In Kapitel 4 widmen wir uns dann dem WebSocket-Protokoll und in Kapitel 5 geht es anschließend ans Eingemachte. Hier lernen wir die WebSocket API kennen und programmieren mit JavaScript erste Beispiele für WebSocket-Clientanwendungen in Webbrowsern. In Kapitel 6 wenden wir uns den WebSocket Implementierungen auf der Serverseite sowie gebräuchlichen Frameworks zu. Weitere wichtige Aspekte folgen in Kapitel 7, das das Testen von verteilten WebSocket-basierten Applikationen beschreibt und auf Eigenschaften der Performance eingeht. Was niemals vergessen werden darf, sind Sicherheitsaspekte, insbesondere wenn die Anwendung aus verteilten Komponenten zusammengesetzt ist, die über offene Netze miteinander gekoppelt sind, wie das im Web quasi immer der Fall ist. Daher wollen wir uns damit ebenfalls in Kapitel 7 auseinandersetzen und die Besonderheiten von WebSockets in diesem Kontext herausstellen. In Kapitel 8 werden wir dann das Gelernte einsetzen und verschiedene „größere“ und vollständige Anwendungen implementieren. Diese haben wir dabei so gewählt, dass damit ein möglichst großes Spektrum an Möglichkeiten aufgezeigt wird. Wir spannen den Bogen von einer generischen Fernsteuerung für Webanwendungen, über ein klassisches Chatsystem über eine Heatmap für Usability-Tests bis hin zu einer Überwachungskamera per Webcam.

1.2 Wer sollte das Buch lesen?

In erster Linie richtet sich das vorliegende Buch an den Entwickler in Ihnen. Möchten Sie mehr über WebSockets wissen und verstehen, was Sie mit diesen so alles in Ihren Webprojekten anstellen können, dann sollten Sie hier fündig werden. Da der Koffer an Werkzeugen, Technologien und Sprachen kaum unterschiedlicher gefüllt ist als bei Webentwicklern, stellen wir auf keine spezifische Umgebung ab, sondern versuchen dieser farbenfrohen Vielfalt gerecht zu werden. So finden sich Beispiele auf Basis von Node.js, vertx und JSR 356 im Buch wieder. Als Programmiersprachen verwenden wir Java und JavaScript. Neben der gängigen Vorstellung und Erläuterung der Buchinhalte sind die zahlreichen Beispielanwendungen in Kapitel 8 eine Stärke des vorliegenden Buches, mit denen das Erlernte geübt und nachvollzogen werden kann. Webentwickler sollten somit in die Lage versetzt werden,

einen umfassenden Einstieg in die Thematik zu bekommen und anhand der exemplarischen Anwendungen einen tief gehenden Einblick in den Einsatz von WebSockets zu erhalten.

Neben Webentwicklern eignet sich das Buch auch für Lehrveranstaltungen an Hochschulen und damit für Studierende verschiedener Fachrichtungen, die mit WebSockets innovative Anwendungen im Web entwickeln wollen. Zudem profitieren Informations- und Softwarearchitekten sowie Konzepter und (technische) Projektleiter vom vorliegenden Buch, da ein Grundverständnis der Kommunikationsmuster, die mit WebSockets realisiert werden können, für Architekturen und Konzepte von Bedeutung sein können. Auch hierfür haben wir versucht, mit den in Kapitel 8 beschriebenen Beispielanwendungen den Horizont aufzuzeichnen.

1.3 Wie sollte mit dem Buch gearbeitet werden?

Das Buch führt sukzessive in die Thematik ein und setzt nur wenige Vorkenntnisse voraus. Wenn Sie über kaum bis wenige Erfahrungen in der Webentwicklung verfügen, raten wir Ihnen, sich an diesem Aufbau zu orientieren und sich Kapitel für Kapitel einzuarbeiten. Nach den Grundlagen werden in den einführenden Kapiteln erste kleinere Beispielanwendungen gezeigt und erläutert. Wir empfehlen, diese nachzuprogrammieren und die Quelltexte im Detail nachzuvollziehen. Ist das Fundament gelegt, geht es mit weiterführenden Aspekten weiter, die selektiv durchgearbeitet werden können. Das Durcharbeiten der Beispielanwendungen ist wiederum sehr empfehlenswert. Es trainiert auf der einen Seite die im Buch vermittelten Inhalte. Bei Bedarf können Unklarheiten nochmals dediziert nachgelesen werden. Auf der anderen Seite zeigen sie die Potenziale von WebSockets auf. Leser mit fundiertem Vorwissen können sicherlich die Grundlagenkapitel überspringen und sich direkt mit den originären Inhalten und den Beispielanwendungen befassen.

http://websocket101.org/

Wir freuen uns von Ihnen zu hören! Ob Lob oder Kritik, ob Frage oder Anregung, ob Hinweis oder Verbesserung, wir glauben, dass das Buch von all dem profitieren kann, wenn es konstruktiv eingebracht wird. Dazu wollen wir unser Nötigstes tun und freuen uns auf Ihr Mitwirken. Aktuelles wird dabei zeitnah auf der buchbegleitenden Webseite bereitgestellt. Schauen Sie doch häufiger mal auf

http://websocket101.org/

vorbei. Dort finden Sie auch unsere Kontaktinformationen.

An dieser Stelle bleibt uns nur noch, Ihnen viel Spaß beim Lesen und Nachprogrammieren sowie viel Erfolg bei den eigenen Entwicklungen zu wünschen.