WebSocket facilitates complete duplex communication between the client and server. HTML5 made it possible to create enterprise-level applications considering the important parameters related to application maintainability. All these enhanced features help developers to remove the dependency on different plugins.

Starting with the basics of HTML5, you will work your way through the process of implementing different applications using WebSockets and HTML5 using the Node.js server. You will explore how to use WebSockets on the mobile and tablet platforms and build a real-time drawing application in HTML5 utilizing WebSockets. You will also learn how to set up the Node.js server and develop an app for presentation sharing.

By the end of this book, you will be familiar with the technology and will have learned how to enhance your HTML5 web application development using modern tools.

Who this book is written for

This book is for web developers who want to learn and implement WebSocket to create interesting apps for modern browsers, leveraging the capabilities of HTML5 with WebSockets.

What you will learn from this book

- Create a web app with the advanced features of the modern Web
- Expand the level of application by using different frameworks and libraries
- Improve HTML5 web application development with the help of up-to-date tools
- Structure your application and speed up development with modern tools
- Develop a WebSocket server using Node.js
- Discover all the basics of using WebSockets on mobile and tablet platforms
- Transmit real-time data between users by building a drawing application in HTML5
- Understand the importance of using framework Angular.js along with Bootstrap.

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Introducing the World of Web App'
- A synopsis of the book’s content
- More information on WebSocket Essentials – Building Apps with HTML5 WebSockets

About the Author

Varun Chopra has a lot of experience in the design and development of enterprise applications. He has worked as a consultant and has extensive experience in integrating different technologies. Besides his love of technology, he is also a singer and a guitarist and loves gadgets.
WebSocket Essentials – Building Apps with HTML5 WebSockets

HTML, the most important part of web development, was lacking somewhere, but now developers are returning to HTML5 because of its enhancements and features, giving them a new experience of development. WebSocket support on different browsers made it easier to develop web applications with a lot of features.

Data communication between the client and the server is one of the most important parts of any web application. Almost all browsers support WebSockets, which makes it more powerful and available. Developers always want to build their application on a concrete ground so that it is reliable for users. WebSocket makes this possible now. With HTML5 enhancements, it is being accepted and appreciated widely in the community.

In this book, you will learn and understand how WebSockets with HTML5 can create great applications, especially applications where data needs to be pushed from both the client side and the server side. With some basic sample applications that we will create in this book, you will understand how the client can be set up and how the Node.js-based WebSocket server can be created with ease.

This book is for developers who want to learn to create WebSocket-based applications. It gives you real-world scenarios for implementing different aspects of communication with WebSockets. It is simple to learn and easy to understand.
What This Book Covers

Chapter 1, *Introducing the World of Web App*, is an introduction to web applications, covering the basics of the Web. This chapter introduces HTML5, its new features and WebSockets.

Chapter 2, *Getting Started with WebSockets*, covers WebSockets in depth, including the benefits of WebSockets and how to create a sample application. Here, you learn to create your own basic WebSocket server using the Node.js platform.

Chapter 3, *Configuring the Server and Transferring Real–time Data*, shows how data can be sent across different users connected to the server using WebSockets. This chapter also covers the creation of an application using the JavaScript library to share a presentation and collaboratively change slides between different users.

Chapter 4, *Using WebSockets in Real Scenario*, demonstrates another application to explain more about how WebSockets can be used in real-world scenarios. This chapter also talks about the JavaScript framework and its uses.

Chapter 5, *WebSockets for Mobile and Tablet*, covers how WebSockets behaves with mobile devices, different libraries for mobile WebSockets, running the server on an Android mobile phone, and the use of the Express.js package for delivering content from within the server.

Chapter 6, *Enhancing HTML5 Web Application Development Using Modern Tools*, explains different tools and techniques that can be used to enhance web application development. This chapter illustrates speeding up development using different tools such as editors, package managers, version control, boilerplates, application frameworks, responsive web design, and more.
Web application development has reached the next level with HTML5 and WebSockets. The revolutionary enhancements in web development technologies have equipped developers with modern tools and techniques. Using WebSockets they can create web applications which can send data not only from the client side but from the server side as well. Web applications with real-time data transfer can be created with a lot lower use of bandwidth. WebSockets by complementing HTML5 enhanced feature can make future applications powerful. Let’s talk about the basics of the Web before understanding WebSockets in detail.

What is the Web?

The foundation of the Web was laid in the late eighties. The Web works on interlinked hypertext documents that we can access using the Internet. A browser plays a vital role in reading and converting these hypertext documents into a readable and more meaningful format, which we call web pages. HTML5 is the hypertext document which the browser reads and renders for us. The browser not only reads and renders but also creates a Document Object Model (DOM) for us so that we can read and manipulate the structure easily. Dynamic manipulation of the DOM can be achieved by the JavaScript language, which is a standard scripting language for HTML. A server plays a vital role in the functioning of the Web. We mainly consider the Web to be divided into two parts: client and server. Client is considered to be the browser whereas server is the one who gives client the data.
Introducing the World of Web App

Let's see how the Web works:

1. Browser requests a URL from server.
2. Server checks and returns the HTML file.
3. Browser engine draws the page.

In a nutshell, this is how the Web works. Browsers and servers are the most important entities of the Web. Browsers have engines that read HTML files and render web applications in the way HTML files are described. Different browsers such as Blink, Trident, Gecko, WebKit, and so on, have different engines to render the HTML of a page. Now servers are the one who are storing all the data and providing the same on user request.

Web applications

In the beginning of the Web, pages were static. They only used to show content with minimal interactivity and functionality. But with the advancement in the standards of the Web along with the evolution of computers, efficient browsers, enhanced tools and libraries, creating web applications has now become very easy, and a lot of functionalities can be added very quickly.

Here is a simple definition of web applications—any application that runs on a browser is a web application. There are many web applications that we use in our daily routine to check mails, read news, watch videos, and so on. Web applications run in your browser and do not require much of your computer resources.

The Web is growing at a very high rate. Many companies are building their applications on the Web. The first and most important benefit is that it is independent of the operating system. You can run it on Windows, Mac, or Linux and it works the same, because the work is mainly done by the browsers and they are available for most operating systems.

Here are some of the examples of web applications:

- Gmail
- Dropbox
- Flickr
- Facebook
Where does WebSockets fit?
As we have seen some examples of web applications, now the question here is where does WebSockets fit in these applications or any web application? Let us first understand something about application behavior; let us take an example of Gmail, which is basically a mailing client. The work of a mailing client is to fetch mails and display them. This sounds simple, but the problem arises when someone sends you a mail and you want that mail to be displayed right away. To implement such a functionality there are different ways, such as polling and long-polling, which are not efficient. So WebSockets solves the problem here by providing a server push facility. WebSockets provides functionality to push from both the client and server side, which makes it stand out.

WebSockets comes with some good features and great benefits over other methods of communication. Some of the features and benefits of WebSockets are:

- Full-duplex communication
- Low bandwidth consumption
- Security
- Low latency
- Works over Transmission Control Protocol (TCP) (although it needs HTTP for initial handshake)
- Supported by almost all the web browsers and web servers including mobile browsers

We can treat WebSockets as a feature which enhances the experience of web applications. And with HTML5-enhanced features, we can create a dynamic and real-time application.

WebSockets over other methods
There are different ways of implementing data communication between a client and server. Flash, Comet, PusherApp, and so on provide us with the features needed to implement the data communication which WebSockets provides. Then the question arises that why should we go with WebSockets? There are many reasons for picking WebSockets over other methods, some of which are as follows:

- In comparison to other means of data communication, WebSockets exhibits low latency, which decreases from nearly 150 ms to 50 ms.
- WebSockets is a lightweight connection and uses low bandwidth.
Introducing the World of Web App

- It requires lesser developer effort in terms of learning and implementation in different technologies.
- Ease of compilation when different technologies are used.
- Code maintenance becomes easy with WebSockets.
- WebSockets offers full-duplex connection support without much overhead.

Modern browsers
Modern browsers are equipped with advanced features to support web applications.

Web applications have a lot of different features, and to support those features, we need browsers—not just ordinary but modern browsers. For a modern browser to support the advanced features provided by HTML5, it has to implement the HTML5 standard, because it has the latest features and functionalities. There are some versions of browsers that do not support HTML5 majorly because they have not implemented the HTML5 standards, either because they were developed earlier or they choose not to.

Some of the advantages of modern browsers are as follows:

- Good performance
- Good security
- Lesser issues
- Faster page loading
- Experimental Application Programming Interface or API
- Support for latest features
- Access to native resources

HTML
HTML is a markup language which is used by browsers to render a webpage. It is the standard fixed by the World Wide Web (W3). This standard has some defined elements which different browsers implements.
Chapter 1

HTML5 – the modern Web standard
After eight years working on the HTML5 standard, W3 finalized the standard on October 28, 2014. This standard is going to be revolutionary for the future of the Web. The enhancements done to the HTML standard are revolutionary. Let us go through the main features of HTML5, which makes it a great standard for the Web:

Media – audio/video
One of the big features introduced in the HTML5 standard is media playback. We can now play audio/video directly using the browser. Earlier we used to use some plugins in order to play audio and video, which added another layer onto our web application. For example, YouTube used Flash player to play videos, but now we can play the videos directly. This feature has been a bigger advantage for applications which are completely built using HTML.

Along with playback of audio and video, we can also capture audio and video resources of the device. Accessing the camera and microphone can be done using the getUserMedia() API, but it is still not available to all browsers because it is an experimental feature; it is, however, a feature that is greatly needed. This API not only gives access to the desktop computer, but also to the camera and microphone in mobile and tablet devices. This is another feature which will remove the dependency on different plugins for media access and capturing.

Canvas
Canvas gives you per pixel access for manipulation at runtime. So you can draw shapes, render graphs, color them, manipulate them, and even manipulate bitmap images per pixel, along with many more features. The canvas feature gives us an upper hand in drawing and making web applications just like Microsoft Paint (formerly Microsoft Paintbrush) or Adobe Photoshop.

The canvas element has a different set of methods to create drawings using lines, circles, boxes, text, graphics, and so on. JavaScript is used to draw in the canvas container.
Introducing the World of Web App

Form elements
There are many enhancements in form elements which help us create a great experience for the users and are easy to manage from the developers' perspective. Validation was a big problem earlier; we had to write our own code for it, but now it is a part of the elements. There are some enhancements which are made keeping mobile devices in mind, such as field type keyboard—for example, a dedicated keyboard for numeric fields. Some of the new elements are:

- **Input**: The following are the input types:
  - `type="email"`: A field with inbuilt email validator
  - `type="url"`: A field with inbuilt URL validator
  - `type="number"`: A field with inbuilt number input restriction and validator
  - `type="range"`: A range slider with max and min function

- **Datalist**: It specifies a predefined list of options for list control.

- **Keygen**: This element provides secure data submission using the public/private key method. From the security point of view, it is an excellent enhancement.

- **Output**: This element helps in showing the output value during form filling.

Semantics
Semantics are elements which have a meaning. Every developer wants to code in a language that is easy to understand and implement. Semantics is what makes it easier to read and understand the code because it defines the meaning of that piece of element or tag. Some examples of semantic elements are `<audio>`, `<video>`, `<form>`, and `<table>`. Examples of non-semantic elements include `<div>` and `<span>`. We can see from the examples that non-semantic elements don't tell us about the content, while semantic elements tell us clearly about the content.

Some of the new semantics in HTML5 are as follows:

- `<section>`
- `<nav>`
- `<article>`
- `<aside>`
- `<hgroup>`
- `<header>`
- `<footer>`
The addition of these new elements will help in making the code more readable and meaningful. And now let me introduce you to custom elements. Yes, now we can make our own custom elements using JavaScript, either by creating them from scratch or extending the default set of DOM elements by adding new behaviors to them. This way we can create different sets of reusable web components and use them across our web application. This feature adds meaning to the code and is a big advantage for large-scale applications to make reusable web components.

**Mobile first**

HTML5 and CSS3 standards are made keeping mobile devices in mind. There are many enhancements that optimize the code for mobile/tablet devices. Mobiles have evolved to an extent where they have become a part of our daily lives. We have started browsing the Internet more on mobile/tablet devices. And HTML5 has given a lot of power to the Web to match up to the modern Web requirements. HTML5 and CSS3 have some excellent features which can deliver the same content for all devices: desktop, mobile, and tablets. Some of the important features include the following:

- **Viewport**: This helps in adjusting the view of webpages based on devices. We can set a different scale level and so on.
- **Media queries**: CSS as per the screen size; isn't it a brilliant feature? Now by using media queries the CSS styling can be changed at runtime. Responsive web design is a very important feature of modern Web. We need the content to be displayed as per the screen size, and it should adapt and show appropriate content eliminating the not so important content from the page for smaller size devices.
- **Touch events**: These are vital events for mobile/tablet devices. **Swipe** is one of the important events that is now a part of HTML5 DOM.

**Offline storage**

The world is emerging with different technologies and we widely use online and web services in order to create an effective work space and a web world that will cater to our professional and personal needs. There are scenarios where you need websites to be accessible offline, that is, without an active internet connection on your device. This can be achieved using the offline storage functionality. Once you have opened a webpage, it is possible to put the data in cache so that next time you open it or for some reason your connection is lost, you can still open and use it.

The offline system is quite important when the data needs to be stored locally for the user. Mainly when it comes to reloading or restoring the pages if the system is in offline mode.
So, whenever we open a URL, it basically hits the server and then the server returns the requested file. Then, the browser renders the file which was given by the server. Now in case we are offline, the browser will take control, and instead of hitting the server to get the file, it loads the files from its local copy which was cached when we opened it earlier. There is also an API which tells us that we are online or offline. It is very helpful in case of mobile/tablet devices where the connectivity can be lost at any point of time.

**Geolocation**

There are many applications which use geolocation, such as Twitter, Facebook, Foursquare, Google Maps, and so on. The introduction of this feature as a part of HTML5 has made it easier for developers to get the location of their device.

Mobile and tablet devices have **Global Positioning System (GPS)**, and using this API, the hardware of the device can be accessed. Let’s take an example of an application where you want to find nearby hotels. Using GPS, your location can be detected and a corresponding list of nearby hotels can be provided. This feature has reduced the effort of developers in implementing features related to geolocation. And yes, it is a feature which needs users’ permission to work. A prompt is given to the user to allow the web application to access their location details.

**Drag and drop**

Drag and drop is a feature which was always there but could only be implemented using some plugins. The good news is that now it is a part of HTML5 standard. By leveraging this feature, a lot of new controls can be defined, as we also have the custom semantics feature which we can use to define our own custom controls.

Web applications use a lot of different controls or widgets to display the data in a more user-friendly way. For large-scale applications where lists and grids are the most important controls to display the data, drag and drop plays a very important role. Controls that show calendars or the timeline of a project need the drag-and-drop feature to make it more usable. Some of the basic interactions are:

- Rearranging items in a list
- Moving items from one list to another
- Rearranging layouts
- Dragging items around the canvas
- Dragging a file from the computer to the browser
There are many good examples of drag-and-drop features. Different companies have implemented and made their own component library, which implements the drag-and-drop feature. Some examples are Sencha, jQueryUI, KineticJS, Kendo UI, and so on.

**Web workers**

Web workers are just JavaScript running in background. JavaScript is mainly used to manipulate the HTML of a webpage at runtime and uses only one main thread. Web workers have made it possible to run a piece of JavaScript code in the background without affecting the current process. Normally, whenever we run a process in JavaScript, it runs in a queue fashion, which means that one process is executed at a time. It blocks the whole UI for some time and you can't click on buttons as well. This has had a huge impact on the application performance. That is one of the reasons why bigger web applications hesitate in choosing HTML, but web workers will surely change this.

**JavaScript**

HTML pages are static; to make them dynamic and interactive, JavaScript is used. JavaScript is called the language of the Web. It is based on **ECMAScript**, and every browser runs JavaScript. All the interactivity from the clicking of a button, navigation to pages, calling services, and so on is done by JavaScript.

There are many frameworks built using JavaScript to make scripting easier to use: one of the majorly used frameworks is jQuery. It gives a user the flexibility to use DOM events, features, and API in a readable and meaningful way.

**Modern servers**

JavaScript is improving at a quick rate. Most developers are now using JavaScript for client-side handling. Introduction of the Node.js server has changed the scope of work of developers. Earlier, developers used different servers, and for that, they had to learn lot of different languages. Node.js removed that gap and gave developers a platform where they could build a server which is JavaScript based.

JavaScript servers built on the Node.js platform are quite simple to use and increase productivity as well. Developers can make a server and run it in very less time. Creating a server in Node.js is very easy and has many features, such as real-time data transfer using different packages available. There are many frameworks which are built for Node.js, such as **Express.js**, which helps in speeding up the development process.
Introducing the World of Web App

Node.js is free platform and provides a lot of different packages which can be distributed freely. The Node Package Manager (NPM) manages the dependencies for an application. It also is a version manager.

**WebSockets**

With the growth in web applications, the need for real-time data which supports full-duplex communication has also increased. Real-time communication is always hard to implement, and people used Flash for the same. The reason Flash-like plugins are used is because this feature was missing in HTML standard. So whenever we wanted to implement such mechanisms in HTML, we used the polling mechanism, which is a very costly process in terms of performance.

HTML5 comes prepared for all the required features needs for a good web application. WebSockets is a part of HTML5 standard and the WebSocket API is fully available to be utilized.

WebSockets gives a full-duplex communication between the client and server, which basically allows data transfer easily and on need basis, unlike the polling mechanism where we keep hitting the server on an interval to check for changes. WebSockets can send data from the server or client side—basically a connection bridge is opened which allows data transfer from both sides. WebSockets has eliminated the use of third-party plugins giving HTML developers the ability to implement it directly using the WebSockets API.

**Summary**

We have seen what the important elements of modern Web are in this chapter, and the enhanced features HTML5 standard has brought to us. We have been introduced to WebSockets, and in the next chapter, we will see the implementation of WebSockets from both the client and server side.
Where to buy this book

You can buy WebSocket Essentials – Building Apps with HTML5 WebSockets from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.