

R-Pi

Team Emertxe



Websockets - API

Introduction

Websockets

Introduction



- Web sockets are defined as a two-way communication between the servers and the clients, which mean both the parties communicate and exchange data at the same time.
- The key points of Web Sockets are true concurrency and optimization of performance, resulting in more responsive and rich web applications.

Websockets

Features

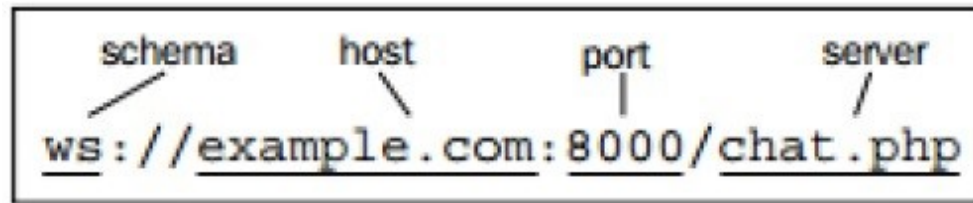
- Protocol is being standardized, which means real time communication between web servers and clients is possible with the help of this protocol.
- Web sockets are transforming to cross platform standard for real time communication between a client and the server.
- This standard enables new kind of the applications. Businesses for real time web application can speed up with the help of this technology.
- The biggest advantage of Web Socket is it provides a two-way communication (full duplex) over a single TCP connection.

Websockets

URL



- HTTP has its own set of schemas such as http and https. Web socket protocol also has similar schema defined in its URL pattern.



- The latest specification of Web Socket protocol is defined as RFC 6455 - a proposed standard.
- RFC 6455 is supported by various browsers like Internet Explorer, Mozilla Firefox, Google Chrome, Safari, and Opera.

Websockets - API

Duplex Communication



Websockets

Its Need

- Internet was conceived to be a collection of Hypertext Mark-up Language (HTML) pages linking one another to form a conceptual web of information.
- During the course of time, static resources increased in number and richer items, such as images and began to be a part of the web fabric.
- Server technologies advanced which allowed dynamic server pages - pages whose content was generated based on a query.
- Soon, the requirement to have more dynamic web pages lead to the availability of Dynamic Hypertext Mark-up Language (DHTML).

Websockets

Its Need



- All thanks to JavaScript. Over the following years, we saw cross frame communication in an attempt to avoid page reloads followed by HTTP Polling within frames.
- However, none of these solutions offered a truly standardized cross browser solution to real-time bi-directional communication between a server and a client.
- This gave rise to the need of Web Sockets Protocol. It gave rise to full-duplex communication bringing desktop-rich functionality to all web browsers.

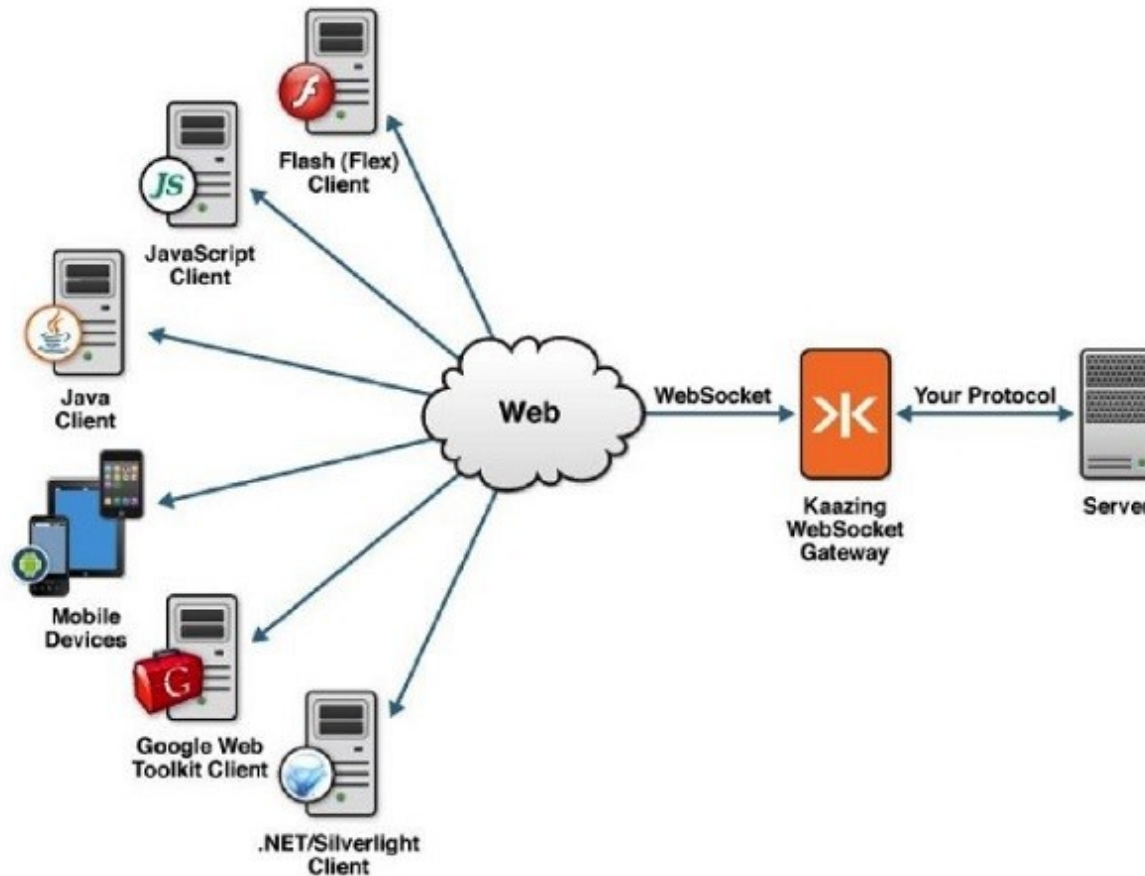
Websockets - API

Functionalities



Websockets

Functionalities



Websockets

Functionalities



- Web Sockets provide a connection between the web server and a client such that both the parties can start sending the data.
- The steps for establishing the connection of Web Socket are as follows -
 - The client establishes a connection through a process known as Web Socket handshake.
 - The process begins with the client sending a regular HTTP request to the server.
 - An Upgrade header is requested. In this request, it informs the server that request is for Web Socket connection.
 - Web Socket URLs use the ws scheme. They are also used for secure Web Socket connections, which are the equivalent to HTTPS.

Websockets - API

Roles



Websockets

Events



- There are four main Web Socket API events
 - Open
 - Message
 - Close
 - Error
- Each of the events are handled by implementing the functions like `onopen`, `onmessage`, `onclose` and `onerror` functions respectively.
- It can also be implemented with the help of `addEventListener` method.

Websockets

Actions



- Events are usually triggered when something happens.
- On the other hand, actions are taken when a user wants something to happen.
- Actions are made by explicit calls using functions by users.
- The Web Socket protocol supports two main actions,
 - `send()`
 - `close()`

Websockets

Opening Connections



- Once a connection has been established between the client and the server, the open event is fired from Web Socket instance.
- It is called as the initial handshake between client and server.
- The event, which is raised once the connection is established, is called the onopen.
- Creating Web Socket connections is really simple.
- All you have to do is call the WebSocket constructor and pass in the URL of your server.

```
// Create a new WebSocket.  
var socket = new WebSocket('ws://echo.websocket.org');
```

Websockets

Opening Connections



- onopen refers to the initial handshake between client and the server which has lead to the first deal and the web application is ready to transmit the data.

```
socket.onopen = function(event) {  
    console.log("Connection established");  
    // Display user friendly messages for the successful establishment of connection  
    var.label = document.getElementById("status");  
    label.innerHTML = "Connection established";  
}
```

The demo of the Web Socket connection established is documented in the given
URL - <https://www.websocket.org/echo.html>

Websockets

Error handlings



- Once a connection has been established between the client and the server, an open event is fired from the Web Socket instance.
- Error are generated for mistakes, which take place during the communication.
- It is marked with the help of onerror event.
- Onerror is always followed by termination of connection.
- The onerror event is fired when something wrong occurs between the communications.
- The event onerror is followed by a connection termination, which is a close event.
- A good practice is to always inform the user about the unexpected error and try to reconnect them.

Websockets

Send + Recieve Msgs

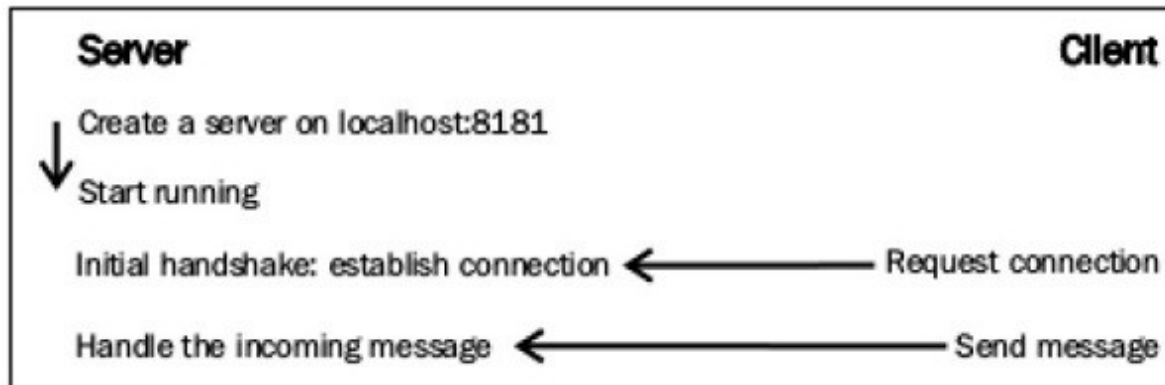


- The Message event takes place usually when the server sends some data.
- Messages sent by the server to the client can include plain text messages, binary data, or images. Whenever data is sent, the onmessage function is fired.
- This event acts as a client's ear to the server.
- Whenever the server sends data, the onmessage event gets fired.
- It is also necessary to take into account what kinds of data can be transferred with the help of Web Sockets.
- Web socket protocol supports text and binary data.
- In terms of Javascript, text refers to as a string, while binary data is represented like ArrayBuffer.

Websockets

Working

- A Web Socket server is a simple program, which has the ability to handle Web Socket events and actions.
- It usually exposes similar methods to the Web Socket client API and most programming languages provide an implementation.



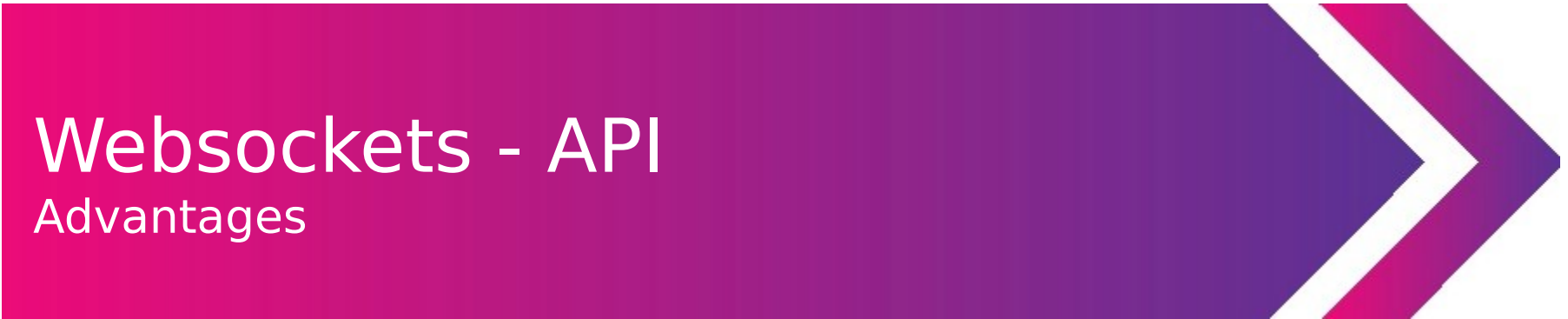
Websockets

Working: Connecting to the Web Server

- The Web Socket server works in a similar way to the Web Socket clients.
- It responds to events and performs actions when necessary.
- Regardless of the programming language used, every Web Socket server performs some specific actions.
- It is initialized to a Web Socket address. It handles OnOpen, OnClose, and OnMessage events, and sends messages to the clients too.

Websockets - API

Advantages



Websockets

Advantages



- Web Socket solves a few issues with REST, or HTTP in general
 - Bidirectional
 - Full Duplex
 - Single TCP Connection

THANK YOU