1

Basic Technologies

- Hypertext Markup Language
- Cascading Style Sheets

2

Interactive Web

- JavaScript
- Critical Rendering Path
- Json & Ajax
- jQuery
- JavaScript ES6
- Websocket
- React

II.2

JavaScript ES6

What is ECMAScript 6 (or ES6)

- ECMAScript 2015 (or ES6)
  sixth and major edition of the ECMAScript language specification standard
  defines the standard for the JavaScript implementation
- features
  - ➢ const
  - ➢ let versus var
    - var
      function-scoped and hoisted at the top within its scope
    - let
      block-scoped ({}) and they are not hoisted
  - ➢ for ... of - loop

II.2

JavaScript ES6

What is ECMAScript 6 (or ES6)

- Template Literals

```
let result = `The sum of ${a} and ${b} is ${a+b}.`;
```

- Default Values for Function Parameters
- Rest Parameters

```
function myFunction(a, b, ...args)
```

- Arrows Functions

```
var sum = (a, b) => a + b;
```

- Destructuring Assignment for Arrays and Objects

```
let fruits = ["Apple", "Banana"];
let [a, b] = fruits;
```

II.2 JavaScript ES6

What is ECMAScript 6 (or ES6)

- Classes
  - to create objects, implement inheritance by using the extends keyword, and reuse the code

```
class Rectangle {
     constructor(...)
     ...
     getCoordinates()
}
class Square extends Rectangle {
     ...
}
```

1

Basic Technologies

- Hypertext Markup Language
- Cascading Style Sheets

2

Interactive Web

- JavaScript
- Critical Rendering Path
- Json & Ajax
- jQuery
- JavaScript ES6
- Websocket
- React

II.2

Websockets

Characteristics so far

- Websites maintain knowledge by cookies
- carries overhead and is open to security vulnerabilities
- all communication is client initiated and each request/response is isolated (stateless)

Requirements

- today's Web applications demand reliable, real time communications with near zero latency
- not just broadcast, but bidirectional communication
- but … classic HTTP does not scale

II.2

Websockets

Requirements

- with advances of WWW, some apps need more interactions and real time communications between a browser and a server
- some apps need to push data to browser without the request by browser
- Advantages
  less data transmission and short delay comparing with other methods such as
  - pulling
  - long pulling
  - streaming

II.2

Websockets

Pulling

- pull technology is a type of communication that takes place over the Internet when a client initiates a transaction by requesting information from a server
- used to deliver content to many different types of applications and devices
  - once the website has loaded, some content, such as automatically updated information, may be pushed to the browser
  - web applications employ pull technology if they must be manually refreshed in order to display updated content
  - computer, tablet, and smartphone applications employ pull technologies if they only check for updated content while they are being used

II.2

Websockets

HTTP-based Long Pulling

- a push-based approach that allows the server to send updates to the client as soon as they are available
  - the client initiates a request to the server, typically through an HTTP request
  - instead of immediately responding, the server holds the request open, keeping the connection active
  - if no new data is available, the server waits until it has something to send back
  - once the server has new data or a predefined timeout occurs, it responds to the client with the latest information
  - upon receiving the response, the client immediately sends another request to the server to maintain the connection
  - this cycle of sending requests and receiving responses continues, ensuring real-time updates

Websockets

HTTP-based Long Pulling

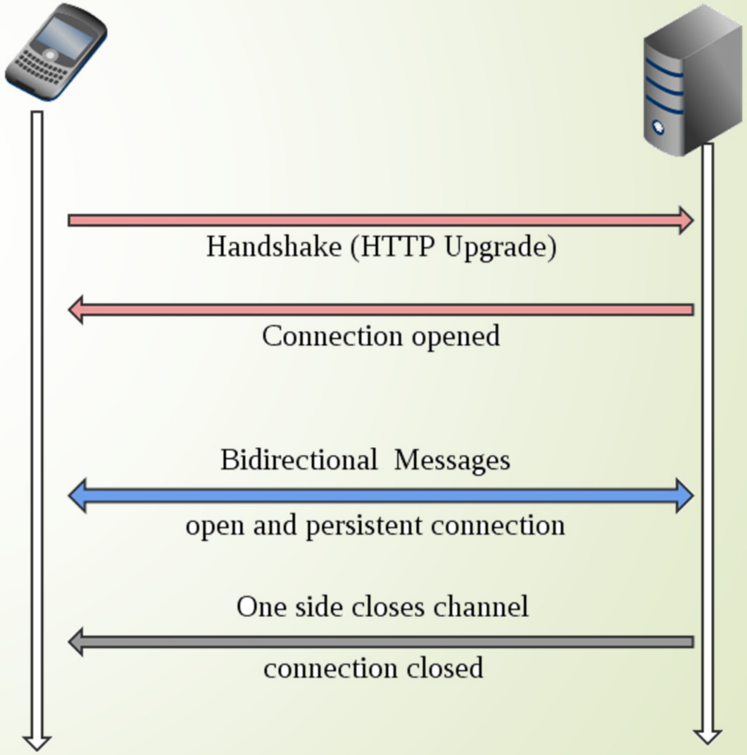
```
(function poll(){
   setTimeout(function(){
      $.ajax({
        url: 'https://api.example.com/endpoint',
        success: function(data) {
          // Do something with `data`
          // ...
          //Setup the next poll recursively
          poll();
        },
        dataType: 'json'
      });
  }, 10000);
})();
```

II.2

Websockets

Websockets

- a web technology providing full-duplex bidirectional communications channels over a single TCP connection

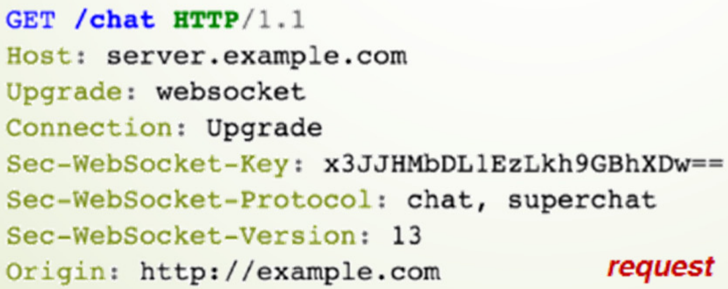


II.2

Websockets

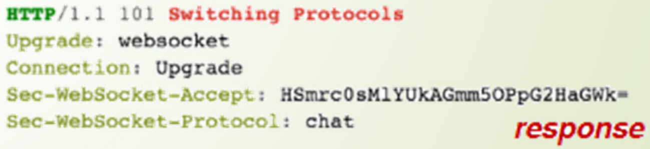
Websockets

- it starts off as a HTTP request, which indicates that it wants to “upgrade” to the WebSocket protocol
- if the server can understand it, then the http connection is switched into a WebSocket connection
- once connected, data is transmitted (bidirectionally) via frames



```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
Origin: http://example.com
```

request



```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: HSmrc0sMlYUkAGmm5OPpG2HaGWk=
Sec-WebSocket-Protocol: chat
```

response

Websockets

Client: JavaScript-Websockets

1. Create a Websocket-connection

```
// Connecting to the server with a protocol called myProtocol
let ws = new WebSocket("ws://echo.websocket.org", "myProtocol");
```

2. Websocket object attributes

   - readyState - represents the state of the connection
     - 0 connection is in progress and has not yet been established
     - 1 connection is established and messages can be sent
     - 2 connection is going through the closing handshake
     - 3 connection has been closed or could not be opened

Websockets

Client: JavaScript-Websockets

3. WebSocket object dispatches four different events

   - ➢ Open
     The server responds to the WebSocket connection request. It indicates that the handshake has taken place and the connection is established.

```
// Event handler for the WebSocket connection opening
ws.onopen = function(e) {
   ...
};
```

   - ➢ Message
     The client receives data from the server. WebSocket messages contain the data from the server.

II.2

Websockets

Client: JavaScript-Websockets

3. WebSocket object dispatches four different events
   - Error
     There is any error in communication.
   - Close
     The connection is closed.

II.2

Websockets

Client: JavaScript-Websockets

4. WebSocket methods

   - send
     transmits data using the connection. If for some reasons the connection is not available or the connection is closed, it throws an exception about the invalid connection state.

   - close
     used to terminate any existing connection. If the connection is already closed, then the method has no effect. The close() method has two optional arguments: code (a numerical status code) and reason (a text string).