Web Communication

adrian.olaru@1and1.ro

HTML

Connectivity

Web Messaging

Web Workers

Server Sent Events

Web Sockets

```
//send a message
postMessage('message');


//receive a message
onmessage = function(e) {
    //the sent message
    e.data;
}
```

The way of the force

```
postMessage(JSON.stringify(message));
```

```
onmessage = function(e) {
  result = JSON.parse(e.data);
};
```

Sending complex data

Don’t trust client e.data

Always check e.origin

Web Messaging

allows documents from different domains to communicate with each other in a secure way

Browser Support*

* source: http://caniuse.com

5.0+

3.0+

8.0

4.0+

10.1+

```
// http://example.com

var iframe = document.getElementsByTagName('iframe')[0],
    win = iframe.contentWindow;

win.postMessage('Where are you?', 'http://example.net');
```

```
//for same domain
win.postMessage('Where are you?', document.location);
```

```
// http://example.net (iframe)

onmessage = function(e){
  if ( e.origin !== 'http://example.com' ) {
    return;
  }

  do_something_with(e.data);

  //you can also send back messages
  e.source.postMessage('In your document!', e.origin);
};
```

Use Cases

portals

widgets

ads

traffic counters

Web Workers

run JavaScript in parallel on a web page, without blocking the user interface

Browser Support*

* source: http://caniuse.com

5.0+

3.5+

4.0+

10.6+

```
// index.html

var worker = new Worker('path/to/my_worker.js');


worker.onmessage = function(e) {
  result.innerHTML = e.data;
};


//calculate 11!
worker.postMessage(11);
```

```
//my_worker.js

onmessage = function(e) {
  var n = parseInt(e.data, 10);
  postMessage(fact(n));
};

//calculate n!
function fact(n) {
  return n ? n * fact(n-1) : 1;
}
```

Notes

can’t access parent & it’s DOM

‘this’ == current worker

can use XHR, SSE or WebSocket

can importScripts(‘script.js’)

can use other workers

can use timers (setTimeout, setInterval)

Server Sent Events

push data from the server to the client over HTTP

Browser Support*

* source: http://caniuse.com

6.0+

4.0

5.0

11

```
// on the client

var source = new EventSource('/events');

source.onmessage = function(e) {
    process(e.data.split('\n').join(''));

    //you can also access the id if one was sent
    //e.lastEventId;
};
```

```
source.onopen = function(e) {};
source.onerror = function(e) {};

source.readyState == source.CONNECTING ||
source.readyState == source.OPEN ||
source.readyState == source.CLOSE

source.close();
```

```
//on the server

if (req.headers.accept &&
    req.headers.accept == 'text/event-stream' &&
    req.url == '/events') {

  res.writeHead(200, {
    'Content-Type': 'text/event-stream',
    'Cache-Control': 'no-cache'
  });

  res.write('data: Server time is:\n');
  res.end('data: ' + (new Date()) + '\n\n');
}
```

```
//you can also associate an id with the message
res.write('id: 1\n');
```

```
//change reconnection timeout; default is around 3000ms
res.write('retry: 10000\n');
```

```
//send JSON data
res.write('data: {\n');
res.write('data: "name": "Adrian Olaru",\n');
res.write('data: "age": 26\n');
res.write('data: }\n\n');
res.end();


//parse message
var result = JSON.parse(e.data);
```

Use Cases

subscribe to live score

real time polls

bidding updates for actions

stock quotes

live news

Web Sockets

a bi-directional connection between the server and the client

Browser Support*

* source: http://caniuse.com

5.0+

4.0*

5.0

11*

* deactivated by default

```
//on the client

var ws = new WebSocket('ws://example.com/');

ws.onmessage = function(e) {
  process(e.data);
};

//send instead of postMessage
ws.send('How are you?');
```

```
ws.onopen = function(e) {};
ws.onclose = function(e) {};

ws.readyState == ws.CONNECTING ||
ws.readyState == ws.OPEN ||
ws.readyState == ws.CLOSED

ws.close();
```

```
//Web Socket Protocol (works on port 80)
new WebSocket('ws://example.com/');

//Secure Web Socket protocol (works on port 443)
new WebSocket('wss://example.com/');
```

Notes

full duplex communication

removes the overhead (only 2 bytes)

dramatically reduces complexity

Use Cases

online chat

online games

realtime geolocation

package tracking

Demos

http://github.com/adrianolaru/webcom


HTML

Danke