shared web workers error handling

How to handle/throw errors in a `shared web worker`?

If you throw an error like this in a `dedicated web worker` the error appears in the console in the browser.. but not when the worker is shared..!?

With a dedicated web worker you can also use `console.log()` in the worker.. but not in a shared web worker

main page

```
var worker = new SharedWorker('js/webworker.js');

worker.port.onmessage = function(e){
    console.log('From worker: '+e.data);
};

worker.port.onerror = function(e){
    console.log(e.message+'\nLine: '+e.lineno+'\nFilename: '+e.filename);
};

worker.port.start();

worker.port.postMessage('This message should throw an error in the web worker');
```

shared web worker

```
var ports = [];

self.onconnect = function(e){
    var port = e.ports[0];
    ports.push(port);

    port.onmessage = function(e){
        port.postMessage(e.data);

        throw Error('hehehe');
    };

    port.start();

    port.postMessage('Worker connected!');
};
```

javascript web-worker

edited Sep 10 '14 at 22:48

asked Sep 10 '14 at 22:41
clarkk
3,299 ● 23 ● 85 ● 159

1 Answer

As the documentation says:

> Whenever an uncaught runtime script error occurs in one of the worker's scripts, if the error did not occur while handling a previous script error, the user agent must report the error at the URL of the resource that contained the script, with the position (line number and column number) where the error occurred, in the origin of the scripts running in the worker, using the WorkerGlobalScope object's onerror attribute.

http://www.w3.org/TR/workers/

So the error will be reported to `self` and then sent to all the ports:

```
self.onerror = function (e) {
  ports.forEach(function (port) { port.postMessage(e); });
};
```

Here is jsfiddle that demonstrates it: http://jsfiddle.net/nhrfgd1L/

answered Sep 10 '14 at 23:31

AlexAtNet
5,982 ● 2 ● 21 ● 41