

MOBILE DEVELOPER

The background is a solid blue color with a faint, white, abstract network pattern. This pattern consists of numerous small dots connected by thin, intersecting lines, creating a web-like structure that covers the entire slide. In the top right corner, there is a thin white curved line that starts from the edge and curves downwards and to the left. In the bottom right corner, there is a solid dark blue circular shape, partially cut off by the edge of the slide.

I processi in background

Sommario

- I Webworker come lavorano in background.

Processi in background

In passato era inconcepibile solo pensare che una pagina web potesse eseguire un processo in maniera asincrona o addirittura in background.

Ciò vuol dire che un'applicazione web costringeva sempre l'utente all'attesa di una risposta in virtù di una nuova richiesta inviata al server.

Limite delle web application

Quando veniva eseguito uno script che eseguiva una grossa logica di business, la pagina web rimaneva letteralmente "**bloccata**" per il tempo necessario per avere una risposta dal server.

Web Worker

I web Worker introducono la possibilità per le applicazioni web di eseguire processi in background, che vengono solitamente eseguire in thread separati.

Modalità operativa del Worker

Un Web Worker è uno script Javascript che viene eseguito in background completamente disgiunto dai task interattivi della pagina principale, evitando quindi di bloccare l'interfaccia presentata (*User Interface*).

Limiti dei Web Worker

Uno script in esecuzione all'interno del Web Worker non può accedere all'oggetto `window` della pagina (`window.document`) di conseguenza non avrà accesso al DOM della pagina.

Tipico impiego dei Web Worker

Un'ipotetica applicazione che resta in ascolto di messaggi broadcast inviati da un server di backend, pubblicandoli sulla pagina principale non appena vengono ricevuti (possibile impiego anche di una *Web Socket*).

Impiego dei Web Worker

Realizzare una pagina web che sfrutta un web worker è estremamente semplice:

1. Creazione oggetto Web Worker, con riferimento al file javascript;
2. Creazione del listener dei messaggi associato al web worker;
3. Uso della funzione `postMessage` per la comunicazione al Web Worker.

Supporto Web Worker

Il supporto da parte dei browser per i Web Worker può essere realizzato con la seguente funzione:

```
function webWorkerSupport () {  
    if (typeof(Worker) !== "undefined") {  
  
        document.getElementById("support").innerHTML  
TML =  
        "Il browser supporta i Web Worker!";  
    }  
}
```

Web Worker: creazione

Quando viene inizializzato un Web Worker è necessario passare il parametro URL al file javascript, che verrà eseguito.

```
Worker = new Worker ("script.js");
```

Web Worker: esecuzione script

La manipolazione del DOM, può passare attraverso l'uso di script detti Helper.

```
importScripts("helper.js");  
  
importScripts("helper.js", "helper2.js");
```

E' consentito importare più helper.

Web Worker: comunicazione

Dopo aver avviato un processo, si possono adoperare `postMessage` per inviare i dati al Web Worker o per farsi che esso stesso invii messaggi alla pagina.

`postMessage` invia solo oggetti javascript.

Web Worker: pagina principale

Il codice della pagina principale per la gestione del Web worker è il seguente:

```
document.getElementById ("helloButton").onclick = function () {  
    worker.postMessage ("invio messaggio!");  
}
```

```
worker.addListener("message", messageHandler, true);  
  
function messageHandler(e) {  
    //operazioni sul messaggio del Web Worker  
}
```

Web worker: codice javascript

Funzione messageHandler:

```
function messageHandler (e) {  
    postMessage ("worker :" + e.data );  
}
```

Aggiunta listener per gli eventi message della pagina principale

```
addListener("message", messageHandler, true);
```


Web Worker: gestione errori

E' necessario aggiungere un listener per la gestione degli errori:

```
function errorHandler(e) {  
    console.log(e.message, e);  
}  
  
Worker.addListener ("error", errorHandler, true);
```

Web Worker: interruzione

Il Web Worker può essere interrotto nel seguente modo:

```
worker.terminate();
```

MOBILE DEVELOPER