



Introduzione a Node.js

Server Side Programming / MMM 2012
Daniele Pezzatini

Cos'è Node.js

Node.js è un framework per realizzare applicazioni Web in **JavaScript**, permettendoci di utilizzare questo linguaggio, tipicamente utilizzato nella "client-side", anche per la scrittura di applicazioni "server-side".

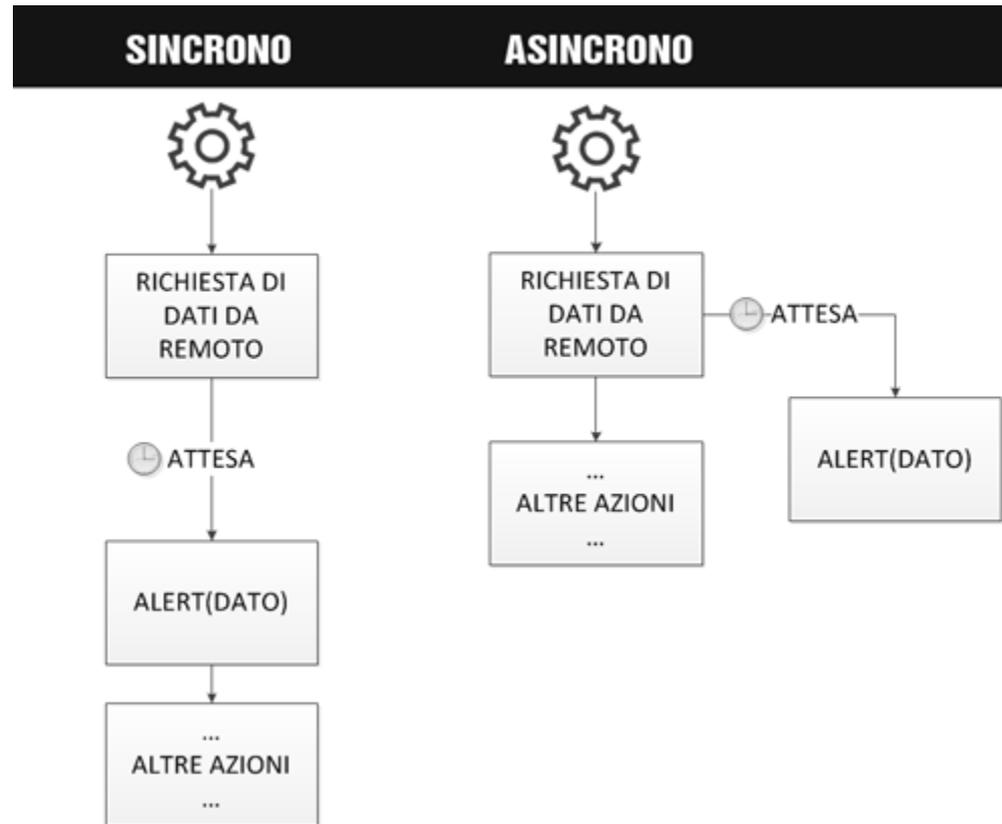
La piattaforma è basata sul **JavaScript Engine V8**, che è il runtime di Google utilizzato anche da Chrome e disponibile sulle principali piattaforme, anche se maggiormente performante su sistemi operativi UNIX-like.

Event-driven programming

La caratteristica principale di Node.js risiede nella possibilità che offre di accedere alle risorse del sistema operativo in modalità **event-driven** e non sfruttando il classico modello basato su processi o thread concorrenti, utilizzato dai classici web server.

Il modello event-driven, o "programmazione ad eventi", si basa su un concetto piuttosto semplice: si lancia una azione quando accade qualcosa. Ogni azione quindi risulta asincrona a differenza dei metodi di programmazione più comuni in cui una azione succede ad un'altra solo dopo che essa è stata completata.

Esecuzione asincrona



Sincrono vs. Asincrono

Approccio sincrono

```
var dato = caricaDati(url);  
alert(dato);
```

L' esecuzione aspetta la ricezione dei dati prima di effettuare l'alert.

Approccio ad eventi (asincrono)

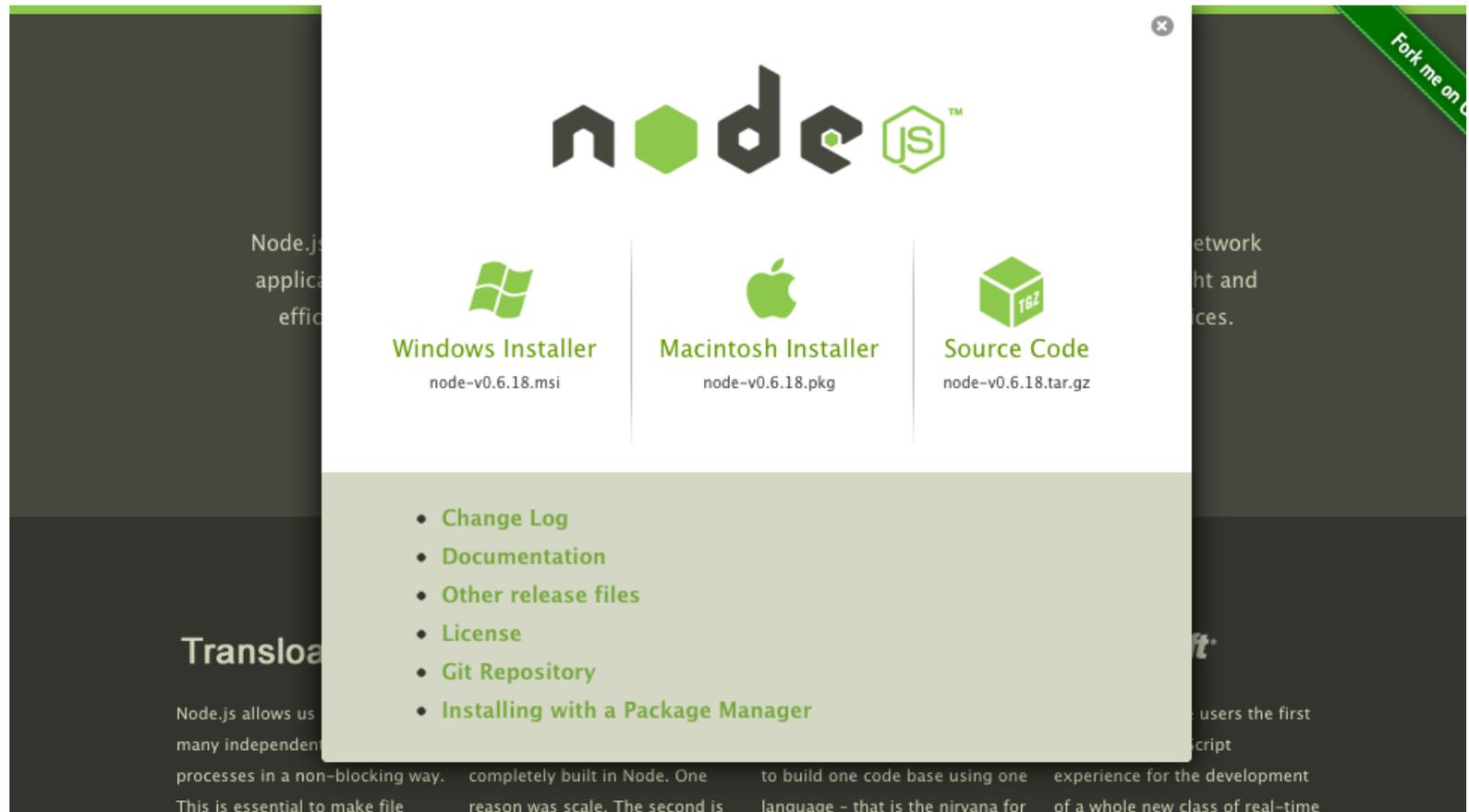
```
caricaDati(url, function(dato) {  
    alert(dato);  
});
```

L'azione da effettuare una volta ottenute i dati richiesti è sotto forma di funzione (detta **callback**).

La callback è eseguita solo quando i dati sono disponibili.

Installazione Node.js

Dal sito ufficiale è possibile scaricare l'installer per Windows e per Mac.



The screenshot displays the Node.js download page. At the top center is the Node.js logo. Below it, three installation options are presented in a grid:

- Windows Installer**: node-v0.6.18.msi (with a Windows logo icon)
- Macintosh Installer**: node-v0.6.18.pkg (with an Apple logo icon)
- Source Code**: node-v0.6.18.tar.gz (with a TGZ icon)

Below these options is a list of links:

- [Change Log](#)
- [Documentation](#)
- [Other release files](#)
- [License](#)
- [Git Repository](#)
- [Installing with a Package Manager](#)

Additional text visible in the background includes "Node.js allows us to run many independent processes in a non-blocking way. This is essential to make file", "completely built in Node. One reason was scale. The second is", "to build one code base using one language - that is the nirvana for", and "experience for the development of a whole new class of real-time". A "Fork me on GitHub" banner is visible in the top right corner.

Installazione Node.js

Una volta installato avremo a disposizione due nuovi comandi:

- **node** che permette di eseguire un'applicazione contenuta in un file passato come parametro
- **npm** permette di scaricare, ricercare ed aggiornare i pacchetti software da includere nelle nostre applicazioni.

Per controllare che l'installazione sia stata eseguita con successo è possibile richiamare il terminale tramite il comando `node` senza nessun parametro.

```
$ node
```

```
> console.log(1)
```

```
1
```

```
.
```

Quando usare Node.js

- Tante richieste e risposte di piccola dimensione
- Applicazioni real-time
- Giochi on-line, chat, sistemi di messaggistica
- Social networks e sistemi di notifica

Quando **NON** usare Node.js

- Risposte HTTP di grandi dimensioni
- Applicazioni CRUD (es. gestionali)
- Server per gestione di file statici (es. immagini)
- Blogs, CMS, E-Commerce

Creiamo un web server: Hello World!

```
var http = require('http');

var server = http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.write('Hello World');
  res.end();
})

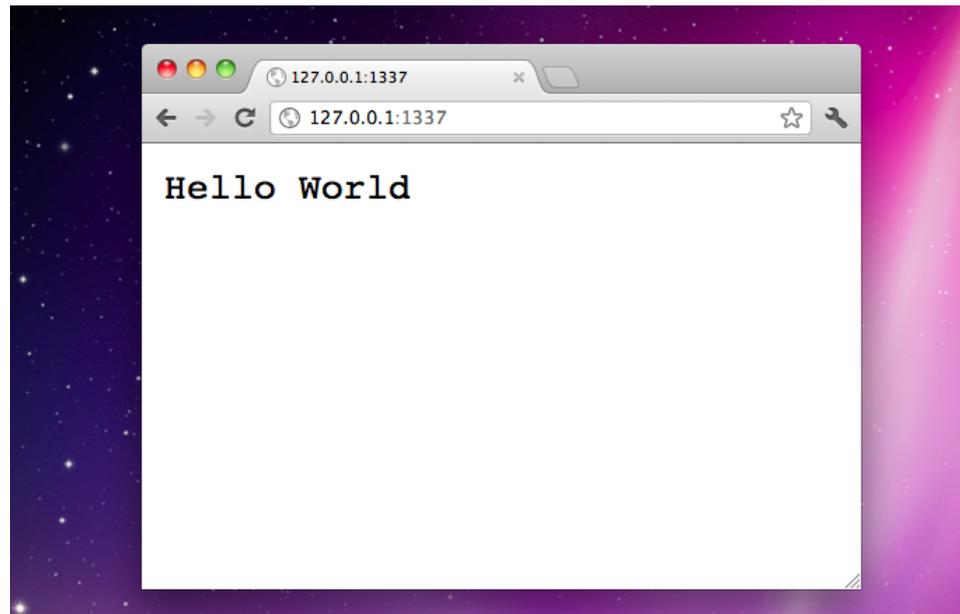
server.listen(1337, '127.0.0.1');

console.log('Server running at http://127.0.0.1:1337/');
```

Creiamo un web server: Hello World!

Salviamo in un file **helloworld.js** il contenuto del file, avviamolo con **node helloworld.js** e facciamo puntare un browser a **http://localhost:1337**.

Dovremmo vedere il primo Hello World inviato da Node.js.



Usare i moduli

Le funzionalità di Node.js sono scritte all'interno di moduli.

Alcuni di essi sono rilasciati insieme all'installazione di Node, altri possono essere aggiunti tramite il comando **npm**

I moduli possono essere richiamati tramite il comando **require()**

Socket.io

Socket.io è un modulo di Node che permette di creare connessioni attive tra il client e il server.

In questo modo il client può ricevere aggiornamenti dal server senza dover eseguire nessuna richiesta.

