



DelphiDay
italian conference

Vue.js e Delphi

seminario

Marco Breveglieri



Software Developer |



Trainer and Consultant |



Tech Content Creator




MARCO BREVEGLIERI

 Software Developer |  Trainer and Consultant |  Tech Content Creator

 www.breveglieri.it

 marco.breveglieri@abls.it

 twitter.com/mbreveglieri

 github.com/marcobreveglieri

 linkedin.com/in/marcobreveglieri



11-12 Giugno 2024
Piacenza



OPEN-SOURCE PROJECTS

github.com/mbreveglieri



CompilaQuindiVa

www.twitch.tv/compilaquindiva

+  **YouTube**



Delphi Podcast

www.delhipodcast.com



Prometheus Client for Delphi

github.com/marcobreveglieri/prometheus-client-delphi



11-12 Giugno 2024
Piacenza



AGENDA

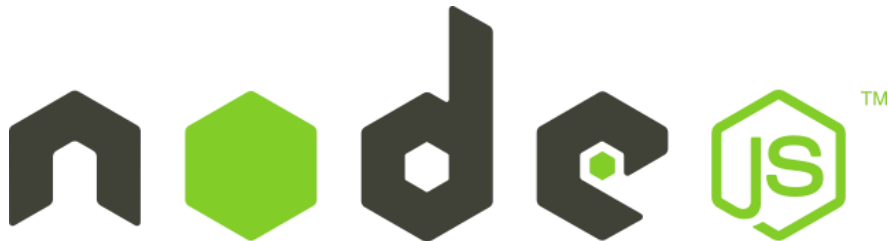
- Node.js e dintorni
- Panoramica di Vue.js
- Creiamo un progetto frontend
- Integriamo il backend in Delphi
- Conclusioni

+ coding interattivo e domande in ogni momento.

Node.js e dintorni

Che cos'è Node.js?

- Runtime per eseguire codice JavaScript
- Basato sul motore V8 di Google Chrome
- Creato da Ryan Dahl nel 2009



Caratteristiche principali

→ Event-Driven

→ Architettura «non blocking» e asincrona

→ Single-threaded

→ Singolo thread per gestire le richieste

→ Migliora la scalabilità generale

→ Ecosistema vastissimo

→ Package Manager (NPM) in bundle

→ Accesso a una pletora di package disponibili



Perché Node.js?

→ Prestazioni elevate

→ Garantite dal modello asincrono e non bloccante

→ Estrema scalabilità

→ Può gestire un numero ampio di connessioni

→ Linguaggio univoco

→ JavaScript viene utilizzato lato client e lato server

→ Facilita il cosiddetto sviluppo «Full Stack»

→ Versatilità estesa

→ App in tempo reale (*chat, giochi online, bet, ecc.*)

→ Servizi Web, microservizi, REST API

→ Applicazioni di tipo I/O bound



Un esempio

```
JS server.js

1  const http = require('http');
2
3  const server = http.createServer((req, res) => {
4    res.statusCode = 200;
5    res.setHeader('Content-Type', 'text/plain');
6    res.end('Hello, World!\n');
7  });
8
9  server.listen(3000, '127.0.0.1', () => {
10   console.log('Server running at http://127.0.0.1:3000/');
11 });
```





Let's code!

npm

→ npm = Node Package Manager

- Gestore ufficiale dei pacchetti per Node.js
- Consente di installare e condividere librerie JavaScript
- Gestisce i pacchetti e tutte le loro dipendenze
- Automatizza task di sviluppo implementati in JavaScript
- Elemento fondamentale per lavorare con *Node.js*
 - **Ampia comunità:** sono disponibili milioni di pacchetti
 - **Facilità d'uso:** semplice da utilizzare per installare e disinstallare pacchetti
 - **Modulare:** supporta la creazione di applicazioni e altri pacchetti riutilizzabili




<https://www.npmjs.com>

Esempi



bash

```
1 npm init
```



bash

```
1 npm install express
```



bash

```
1 npm run start
```




Let's code!

Introduzione a Vue.js

Vue.js

→ E' un framework JavaScript per creare interfacce utente

- A differenza di Angular, non si occupa di altri servizi a supporto dell'applicazione Web
- E' possibile strutturare applicazioni complete usando toolkit completi come Nuxt

→ Creato da Evan You nel 2014 (release 1.0)

→ Perfetto per sviluppare SPA
(*Single Page Applications*)

Parere personale: rispetto ad altre soluzioni dello stesso tipo, è un ottimo compromesso tra semplicità e funzionalità.



👉 <https://vuejs.org>

Vue.js

→ Caratteristiche principali

- Reattività: data-binding per aggiornare automaticamente il DOM della pagina
- Componenti: basato su componenti riutilizzabili
- Facilità di integrazione anche in progetti esistenti (con diverse limitazioni in base agli scenari)

→ Perché Vue.js?

- Curva di apprendimento (relativamente) bassa
- Documentazione ampia: guide, corsi online
- Performance elevate: ottimizzato e veloce



Esempi (versione da CDN per pagine tradizionali)

1

```
index.html
1 <div id="app">
2   {{ message }}
3 </div>
4
5 <script src="https://cdn.jsdelivr.net/npm/vue@2"></script>
6 <script>
7   var app = new Vue({
8     el: '#app',
9     data: {
10       message: 'Hello, Vue.js!'
11     }
12   });
13 </script>
```

Esempi (versione da CDN per pagine tradizionali)

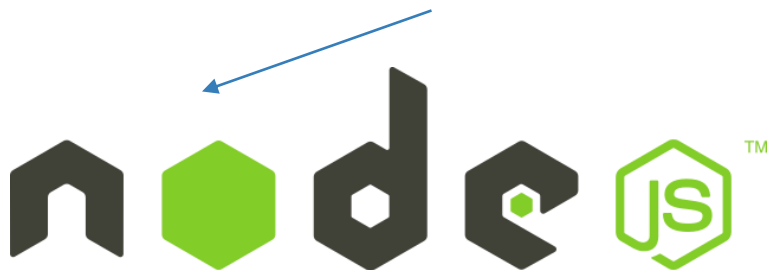
2

```
JS app.js

1 Vue.component('todo-item', {
2   props: ['todo'],
3   template: '<li>{{ todo.text }}</li>'
4 });
5
6 var app = new Vue({
7   el: '#app',
8   data: {
9     groceryList: [
10      { id: 0, text: 'Vegetables' },
11      { id: 1, text: 'Cheese' },
12      { id: 2, text: 'Whatever else humans are supposed to eat' }
13    ]
14  }
15 });
```


tuttavia

noi lo useremo in modo diverso!



Vite

- **Tool di build JavaScript di nuova generazione**
- Creato da Evan You (lo stesso di *Vue.js*!)
- Ambiente per sviluppo veloce e build
 - Supporta *Hot Module Replacement* (HMR), ovvero aggiorna la pagina durante lo sviluppo senza ricaricare
 - Build ottimizzata con bundle efficienti e minimizzati
 - Prestazioni elevate e tempi di avvio rapidi
 - Configurazione minima (anche «zero configuration»)
- Supporta tutti i framework JavaScript moderni (tra cui Vue, React, Svelte) tramite plugin



<https://vitejs.dev>

Creiamo un progetto!

Sviluppo frontend con Vue.js

→ **Vue è un framework «progressivo» che soddisfa le principali necessità di sviluppo frontend**

- Miglioramento dell'HTML statico senza una fase di creazione
- Incorporamento come componenti Web in qualsiasi pagina
- Applicazione a pagina singola (SPA)
- Full Stack/Rendering lato server (SSR)
- Jamstack/Generazione di siti statici (SSG)
- Targeting per desktop, mobile, WebGL e persino terminale!

Il Web è un universo eterogeneo e molto esigente:
come supportarlo con *Vue.js*?



Elementi fondamentali

- Uso di **Single File Components (SFC)**
 - Contengono il template (HTML), il codice (JavaScript) e lo stile (CSS) in un singolo file
 - Hanno estensione «.vue»
- API di programmazione disponibile con **diversi stili**
 - **Options API**: la logica del componente viene implementata creando un oggetto che espone una interfaccia specifica
 - **Composition API**: la logica del componente viene implementata chiamando delle funzioni speciali del framework
- Creazione e gestione del progetto via *npm* con tool *Vite* (e altri)

Tool per il coding

- Si utilizza un flavor moderno di JavaScript (grazie a Node)
 - Il progetto finale è comunque «transpilato» in una versione di JS comprensibile ai browser
- E' supportato il linguaggio TypeScript (facoltativo e anche parziale)
- E' possibile usare alternative di npm (e di altri tool, se compatibili)
- E' possibile utilizzare diversi ambienti di sviluppo per il Web
 - Visual Studio Code (ricordarsi di installare l'estensione «Vue Official», ex «Volar»)
 - JetBrains WebStorm (supporto integrato a Vue, Vite e tanti altri tool, estensibile via plugin)
 - Sublime Text, Atom, ecc.
- Test e debugging in browser: Firefox Developer Edition, Chrome, Edge
 - Installa le estensioni per lo sviluppo in Vue!

Vediamo assieme...

- Struttura di un progetto Vue
- Creazione di componenti, con proprietà ed eventi
- Sintassi dei template dei componenti
- Codifica usando Options/Composition API
- Data Binding, direttive e principali use-case
- Approfondimento della sintassi JavaScript (e TypeScript)
- Debug, build di produzione e deploy



Let's code!

...e Delphi?



Delphi come backend

- Delphi è perfetto per il **backend** della tua applicazione
- Crea una **Web API** per inviare e ricevere dati dalla tua app Vue.js
 - Usa il framework che preferisci (WiRL, RAD Server, DMVCFramework, Horse, ecc.)
 - Puoi usare l'approccio REST, o GraphQL, o quello che preferisci
 - Integra FireDAC per l'accesso ai dati sfruttando le funzionalità di serializzazione
- Esponi la tua applicazione Vue dal server Delphi, oppure caricala su un hosting separato
 - L'applicazione finale Vue.js si ottiene con il comando `npm build`
 - Ricorda che i file dell'applicazione in sé sono semplici «file statici»
 - Non è richiesto Node.js per poter far funzionare l'applicazione Vue.js
 - Per copiare i file nella giusta location, usa strumenti CI/CD e DevOps!



Let's code!

Conclusioni

Vue.js + Delphi

- Vue.js è un ottimo framework per lo sviluppo di applicazioni Web
 - Supporto di tutti gli scenari moderni
 - Build e debug efficiente grazie a Vite
 - Tooling e supporto degli IDE diffuso
 - Curva di apprendimento morbida e ampia documentazione
 - Community in espansione e concorrenziale rispetto a tool più blasonati Angular e React
- Rispetto ad altri framework Delphi-like
 - Maggiore integrazione con il mondo JavaScript
 - Framework più conosciuto e diffuso (in caso di difficoltà a individuare «frontendisti»)
 - Package e librerie di terze parti infinita e integrazione semplificata
- E' necessario comunque documentarsi un minimo (ma con questo seminario, in parte l'avete già fatto...) ;-)
- Lo sviluppo frontend con Vue può essere gestito separatamente rispetto al progetto di backend, che può essere realizzato al 100% in Delphi

Altre cose interessanti...

- Gestione dello stato dell'applicazione
 - Pinia (sostituto di Vuex)
- Strumenti di build e configurazione
 - Vite (sostituisce VueCLI)
- Librerie e framework aggiuntivi
 - Nuxt (e Nuxt UI)
 - Vuetify, Quasar, BootstrapVue
- Siti di approfondimento
 - VueMastery (<https://www.vuemastery.com>)
 - VueSchool (<https://vueschool.io>)

Domande? 🤔





THANK YOU