

# Sviluppare SPA con Vue.js

Fabrizio Bernabei

Senior Software Developer

@bernabei

bernabei@gmail.com











<https://github.com/bernabei>



# Un'altro framework javascript???

- Vue (pronounced /vju:/, like **view**) is a **progressive framework** for building user interfaces (<http://vuejs.org>)
- Più precisamente... una **libreria** per il rendering di viste
- Leggero (20K min+gzip) e di facile apprendimento
- Diventa framework grazie a moduli separati che è possibile aggiungere
- Progressive... “scales between a library and a full-featured framework”

# Sempre più popolare

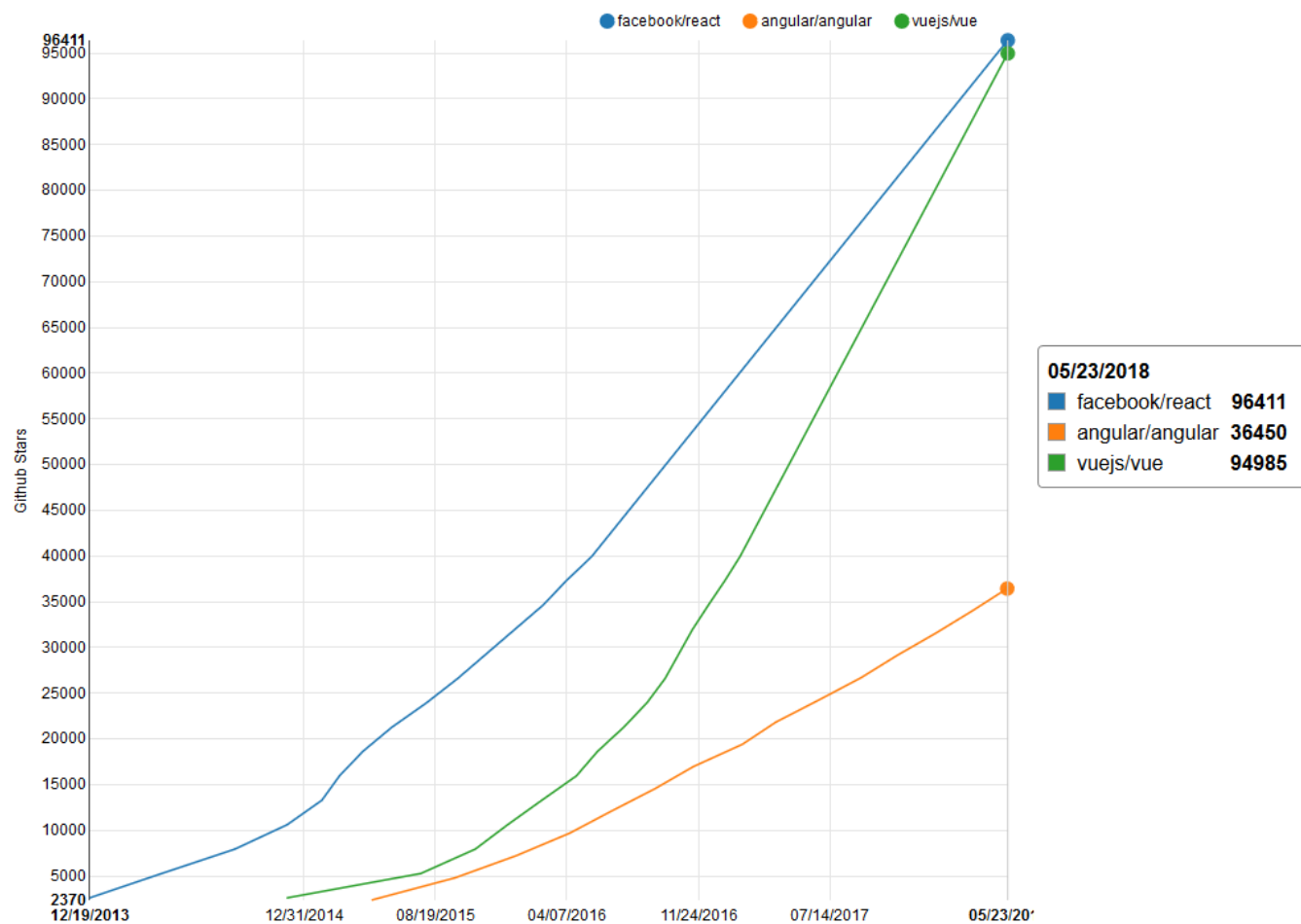
1		<b>Vue.js</b> A progressive, incrementally-adoptable JavaScript framework...	+40.0k ★
2		<b>React</b> A declarative, efficient, and flexible JavaScript library for buildi...	+27.8k ★
3		<b>Angular</b> One framework. Mobile & desktop.	+12.2k ★
4		<b>Preact</b> Fast 3kB React alternative with the same modern API. Com...	+10.4k ★
5		<b>Hyperapp</b> 1 KB JavaScript library for building web applications.	+8.1k ★
6		<b>dva</b> React and redux based, lightweight and elm-style framework...	+5.3k ★
7		<b>Inferno</b> An extremely fast, React-like JavaScript library for building mod...	+4.6k ★
8		<b>Moon</b> A minimal, blazing fast UI library.	+3.7k ★
9		<b>Angular 1</b> AngularJS - HTML enhanced for web apps!	+3.6k ★
10		<b>Rax</b> A universal React-compatible render engine.	+2.6k ★

- Most Popular Projects Overall 2017
- Most Popular Front-end Frameworks 2017 (e 2016)

<https://risingstars.js.org/2017/en/>



# E continua a crescere



<http://www.timqian.com/star-history/#facebook/react&angular/angular&vuejs/vue>

# Storia e punti di forza

- Nato nel 2014 da Evan You (ex Google)
- Reattivo e focalizzato sul core con moduli aggiuntivi
- V2 nel 2016... Miglioramento **prestazioni** (già tra le migliori) con Virtual DOM, scelto come framework ufficiale per Laravel
- Sviluppo “agile” ma solidi finanziatori (Ali Baba)
- Flessibile (template html e data binding, ma anche jsx... CLI, Single File Component, ES6, Typescript, Mobile con NativeScript e tanto altro)
- Curva di apprendimento minima e documentazione eccellente
- Larga community in crescita costante, supporto e tool di sviluppo

# Vediamolo all'opera

- Setup minimo: inclusione file javascript nella pagina html

```
<script src="https://cdn.jsdelivr.net/npm/vue"></script>
```

- Definizione template HTML nella pagina

```
<div id="app">  
  {{ message }}  
</div>
```

- Inserimento e collegamento al template dell'istanza Vue

```
var app = new Vue({  
  el: '#app',  
  data: {  
    message: 'Hello Vue!'  
  }  
})
```

# Demo

Hello world

# Concetti principali

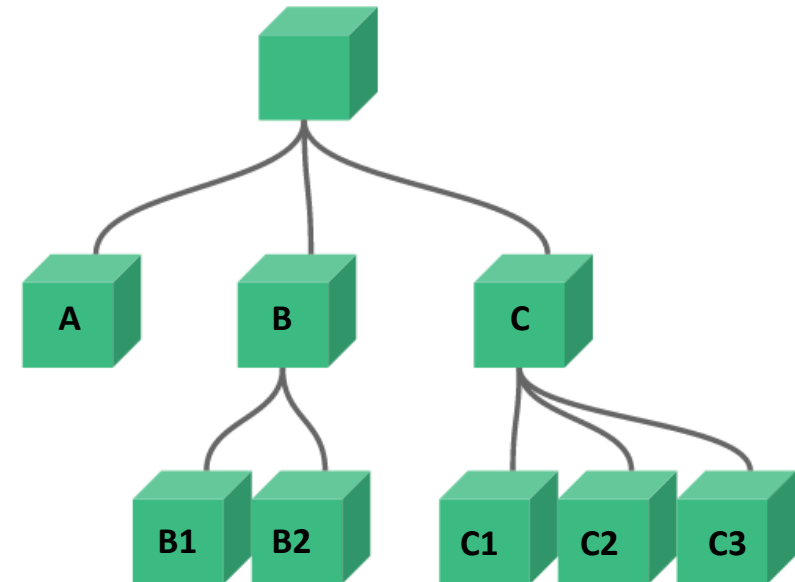
- MVVM (Model View ViewModel)
  - data (reactivity/observable state)
  - methods (logic)
- Espressioni (mustache text interpolation)  
{{ model }}
- Data binding e direttive
  - v-bind (one way... shortcut : )
  - v-model (two way)
  - v-if/v-show (conditional)
  - v-for (loop)
  - v-on (event listeners... shortcut @ )
  - ...
- Computed props
  - computed (reactivity tree)
  - Watch
- Molto altro
  - Class & styles binding
  - Filtri
  - ...eccellente documentazione ;)

# Demo

Hello world 2

# Incentiva la strutturazione a componenti

- Riutilizzabili ed essenziali in applicazioni complesse e SPA
- Esiste una vasta scelta di librerie di componenti (ElementUI, Vuefy, Buefy, ...)



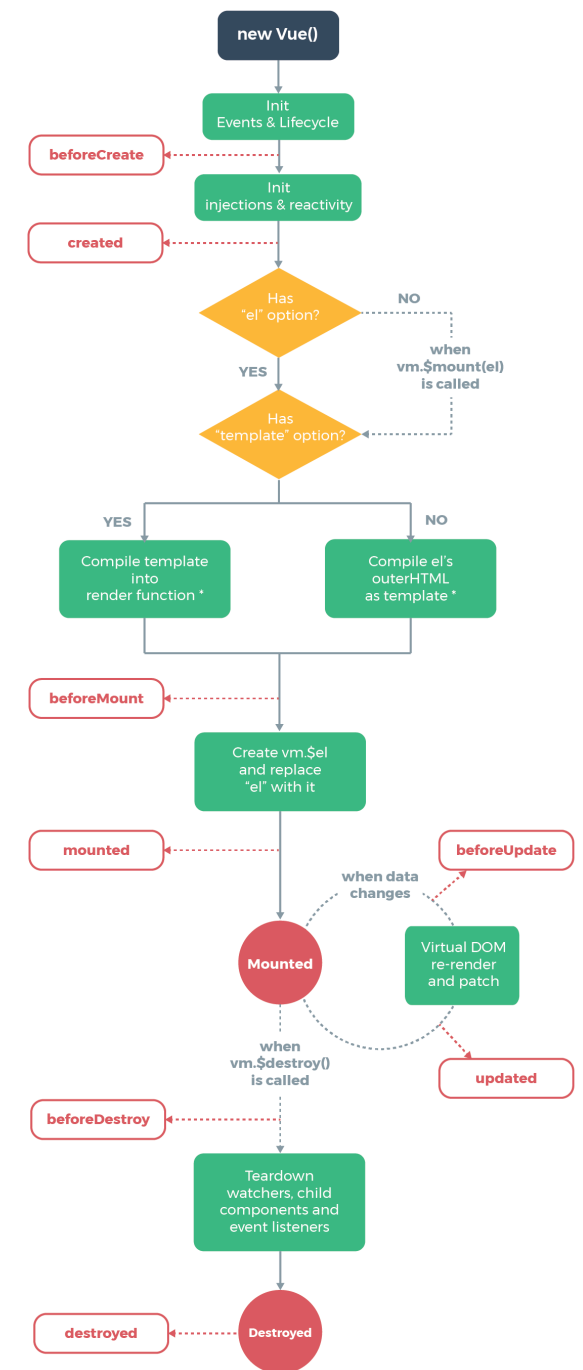
# Componenti

- Core concept di Vue (simile a React)

```
Vue.component('todo-item', {  
  props: ['todo'],  
  template: `- 
               v-bind:style="{ textDecoration: todo.done ?  
               'line-through' : 'none' }">  
               {{ todo.text }}  
             </li>`  
})
```

# Ciclo di vita istanza Vue

- Vue root instance
- Ogni componente è una istanza Vue
- Instance Lifecycle Hooks (principali)
  - created
  - mounted
  - beforeUpdate/update
  - destroyed



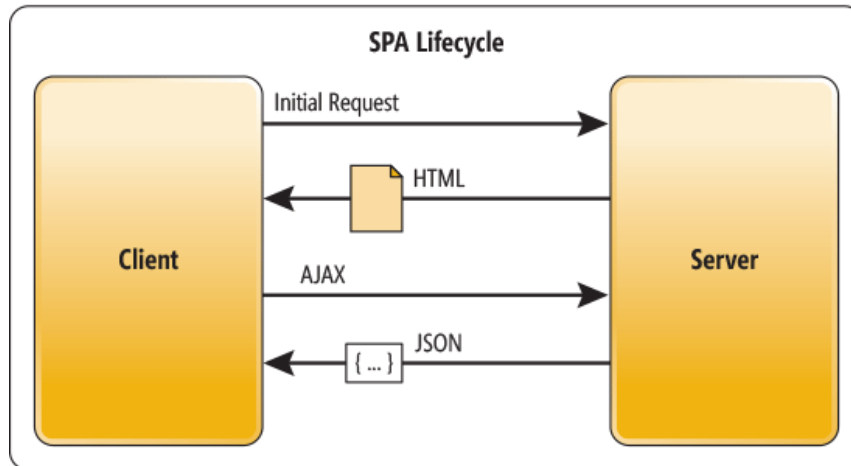
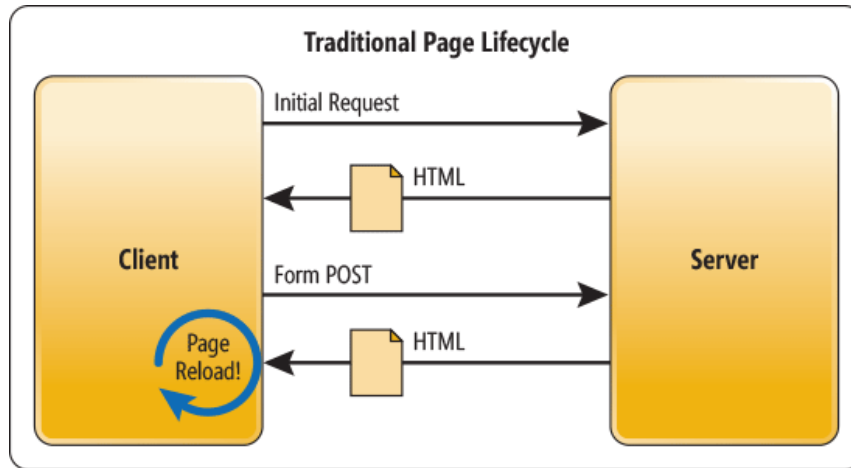
# Components are the building blocks of your SPA



# Demo

Hello world, with component

# SPA, un ripasso veloce



- Acronimo di **S**ingle **P**age **A**pplication
- Applicazione più veloci e responsive
- Server fornisce solamente dati (API)
- Client solitamente carica una singola pagina, javascript interagisce con server ed aggiorna la pagina con i dati recuperati dal server

# Applicazioni Vue con CLI

- Velocizza la creazione della struttura dell'app
- Permette di aggiungere plugin in qualsiasi momento (no eject)
- Si occupa della configurazione dei tool/plugin

```
npm install -g @vue/cli  
vue create my-project  
npm run serve  
  
(vue add [plugin])
```

# CLI: Scelta plugin e configurazioni

```
Vue CLI v3.0.0-beta.11
? Please pick a preset: Manually select features
? Check the features needed for your project:
(*) Babel
(*) TypeScript
(*) Progressive Web App (PWA) Support
(*) Router
(*) Vuex
(*) CSS Pre-processors
> (*) Linter / Formatter
(*) Unit Testing
(*) E2E Testing
```

- progetto configurato con webpack
- usa i file .vue per raggruppare template html, codice js e css
- Configura pacchetti npm e file di configurazione



```
├─ public
│  └─ ★ favicon.ico
│     └─ <> index.html
├─ src
│  └─ assets
│     └─ components
│        └─ ▼ HelloWorld.vue
│           └─ ▼ App.vue
│              └─ JS main.js
│                 └─ .eslintrc.js
│                    └─ .gitignore
│                       └─ JS .postcssrc.js
│                          └─ JS babel.config.js
│                             └─ {} package-lock.json
│                                └─ {} package.json
│                                   └─ ⓘ README.md
```

# Demo

SPA - basic

# Un passo avanti verso la SPA: vue-router

- Gestisce routing lato client

```
export default new Router({
  routes: [
    {
      path: '/',
      name: 'home',
      component: Home
    },
    {
      path: '/quiz/:quizId',
      name: 'quiz',
      component: Quiz
    },
    {
      path: '/about',
      name: 'about',
      component: About
    }
  ]
})
```

- Permette di mappare path di navigazione su componenti
- Possibilità di passare parametri ai component
- Possibilità di definire routing annidati
- Mette a disposizione componenti e servizi per l'interazione con il routing:
  - `router-view` e `router-link`
  - `$route` e `$router`

<https://router.vuejs.org/en/>

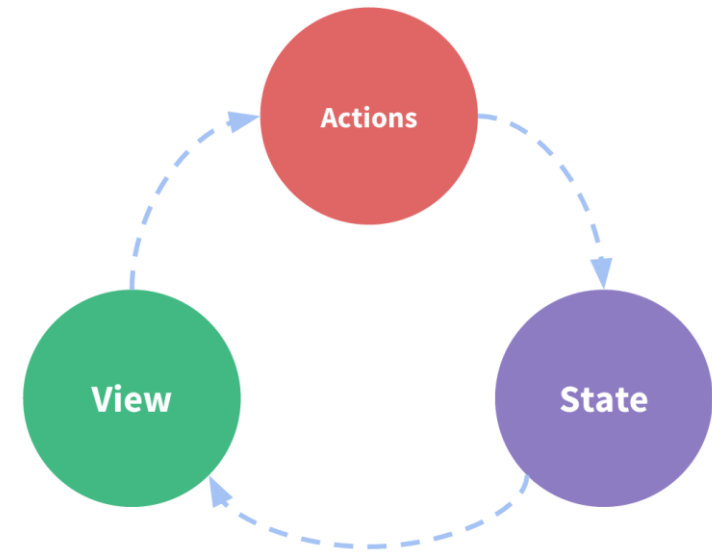
# Demo

SPA - routing

# Lo stato dell'applicazione: Vuex

- Vuex è un **pattern per la gestione dello stato**, ma anche una **libreria** fatta su misura per l'utilizzo con Vue.
- Flusso di base di un'app Vue:

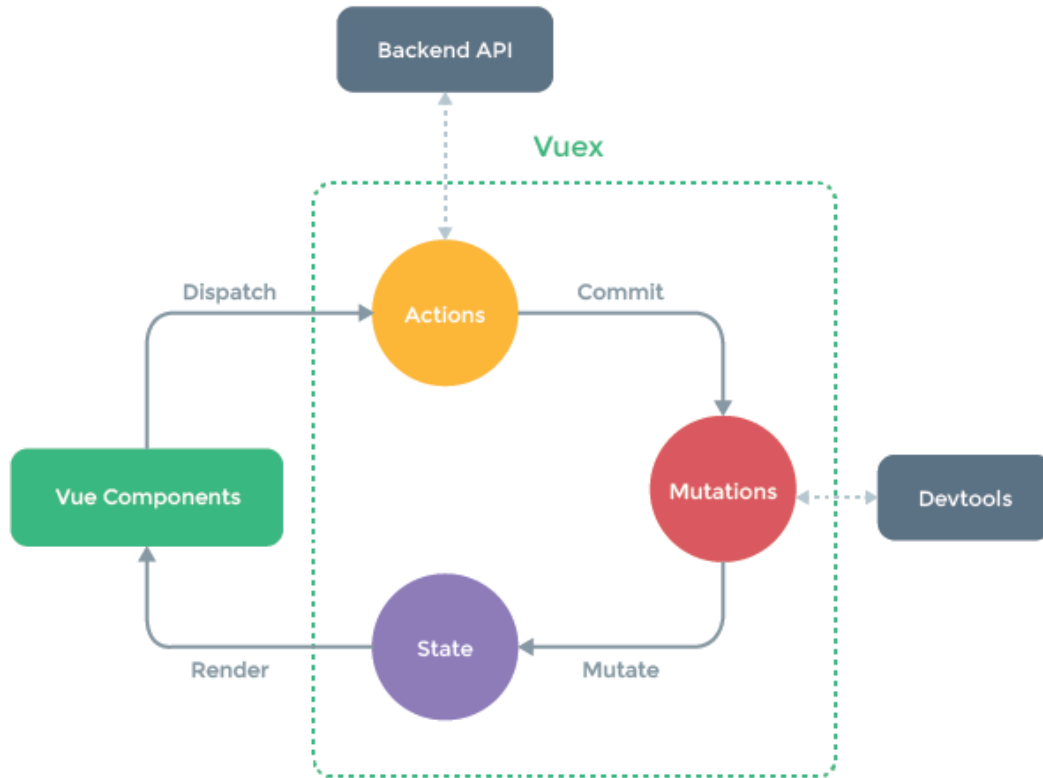
```
new Vue({  
  // state (the source of truth that drives our app)  
  data () {  
    return { count: 0 }  
  },  
  // view (just a declarative mapping of the state)  
  template: `<div>{{ count }}</div>`,  
  // actions (the possible ways the state could  
  // change in reaction to user inputs from the view)  
  methods: {  
    increment () { this.count++ }  
  }  
})
```



# Perchè Vuex

- All'aumentare della complessità, aumentano le necessità di:
  - Condividere/visualizzare lo stesso stato fra view diverse  
(passing props can be tedious for deeply nested components, and simply doesn't work for sibling components)
  - Azioni di componenti diversi possono mutare lo stesso pezzo di stato  
(we often find ourselves resorting to solutions such as reaching for direct parent/child instance references or trying to mutate and synchronize multiple copies of the state via events)
- Idea: Estrarre lo stato dai component e mantenerlo in un unico singleton, in modo che tutti i component possano accedervi o inviare modifiche.
- Inoltre separando le responsabilità e definendo delle regole comuni di utilizzo, è possibile forzare la scrittura di codice manutenibile.
- Idea alla base di Flux, Redux e altri, ma con implementazioni che sfruttano le caratteristiche di Vue (helper mapState, reactivity, ...)!

# Vuex: la soluzione



- Stato - immutabile
- Azioni – per notificare la volontà di aggiornamento dello stato
- Mutazioni – l’unico “attore” in grado di modificare lo stato.

<https://github.com/vuejs/vuex>

Flux libraries are like glasses: you'll know when you need them. (Dan Abramov, creatore di Redux)



# Demo

SPA - vuex

# Grazie! 😊

- Il materiale sarà online nei prossimi giorni su <http://www.communitydays.it>
- Il codice è disponibile anche su <https://github.com/bernabei/cdays18-vue>
- Per saperne di più  
<https://vuejs.org/v2/guide/>  
<https://vuejs.org/v2/cookbook/>