

# wiki for Vuetify Material Design & VueJs Frameworks

Indice:

Info sul progetto

Basato su

Per Cominciare

Requisiti

Sintassi

Componenti

## Info sul progetto

L'obiettivo di questo progetto è quello di rifare il frontend (la parte che vede l'utente finale) di un portale usato per la distribuzione dell'energia. Il nostro compito è quello di ricreare le pagine del sito attuale, usando i componenti già pronti dal framework scelto, in questo caso, Vuetify. Altre persone (più esperte) si occuperanno della parte di backend del progetto, probabilmente alcune saranno nella vostra stessa stanza. Se avete dubbi, domande o perplessità, non abbiate paura a fargli domande.

## Basato su

Vuetify framework per frontend, VueJs e NuxtJs per backend

## Per Cominciare

Bisogna aver installato NodeJs, Git (da usare insieme ad un account github ) e infine un editor che supporti la sintassi di javascript. Di seguito trovi tutto il necessario

## Requisiti

NodeJs- versione LTS

Git- Windows/Linux

Editor Generico o IDE:

Visual Studio Code (consigliato perchè possiede molte estensioni per JS/VueJS)

Visual Studio Community

Jetbrains IDE come Webstorm, PhpStorm ecc. (non gratuiti ma disponibili in versione studente)

## Sintassi

La sintassi di Vuetify non è altro che quella html, ovvero con tag di apertura e/o chiusura, tra i quali va il contenuto del tag. Il vantaggio di questo framework ui rispetto al html è che non bisogna scrivere ogni singola cosa da zero, ma si possono utilizzare componenti e stili già presenti in Vuetify o crearne dei nuovi personalizzati (sempre attenendosi alla documentazione), quindi troverai dei tag che non esistono in html e che iniziano tutti con v-nometag, ovviamente questo non impedisce di usare i tag “stock” di html. In questa wiki verranno citati solo i componenti che abbiamo usato più spesso; per l’elenco di tutti i componenti e le loro proprietà fate riferimento alla documentazione ufficiale disponibile cliccando sull’immagine sottostante.

## Componenti

### **v-card**

Le v-card sono dei componenti versatili, possono contenere molti elementi al loro interno, quali bottoni, icone, testo, tabelle e molti altri. Hanno tre sottocomponenti principali (se vengono usati devono per forza essere all’interno della v-card), che sono v-card-title, v-card-subtitle e v-card-text, che servono ad impostare rispettivamente il titolo, il sottotitolo e il testo (corpo) della v-card. Una proprietà che abbiamo spesso usato è la proprietà flat, che rimuove ombra(elevation in vuetify) della v-card.

### **v-data-table**

Le v-data-table sono fondamentali per mostrare dati sottoforma di tabelle. Il loro utilizzo è relativamente semplice; nella tag di apertura vanno inserite le prop, ovvero properties, (Ad esempio con i : davanti alla prop questa assumerà il valore della variabile indicata, altrimenti assumerà valore di stringa) la prop :headers, e :items sono fondamentali per caricare le intestazioni che devono essere usate (headers) e i dati che la tabella dovrà mostrare (items). Talvolta può capitare che in una colonna della tabella vogliate mostrare più di una cosa oppure vorrete inserire dei bottoni o delle icone ecc. Per fare ciò dovreste usare un v-slot. Quando si usa v-slot, questo viene passato nell’ambito degli slot, come da documentazione. Tuttavia, ci sono altri componenti più complessi, per i quali è necessario legare gli attributi ai componenti corretti, per garantire un supporto adeguato, andando poi a dire cosa mettere in quella colonna/e.

### **v-row & v-col**

Questi due componenti vengono usati per creare delle griglie. Il tag `v-col` deve essere dentro al tag `v-row`; dentro ad uno stesso tag `v-row` ci possono essere più tag `v-col` (questo vuol dire più colonne nella riga). Il tag `v-row` può essere accompagnato dalla proprietà `dense`, che permette di mostrare le righe “più attaccate” tra loro, risulta molto utile quando si hanno molte righe con dati; per quanto riguarda `v-col` la proprietà `cols` (`cols=“n”`), con un numero al posto di `n` (tra 1 e 12, dove 1 è il minimo e 12 il massimo), è molto utile in quanto permette di decidere quante colonne devono essere occupate da un certo elemento.

### **v-icon**

Potrebbe essere che vi serva di aggiungere delle icone alla vostra tabella/pagina/bottone. Per poterle aggiungere bisogna mettere un tag `v-icon`, e tra il tag di apertura e quello di chiusura va il nome dell'icona che volete mettere; le icone usate di default sono di Material Design Iconse potete trovarli cliccando [qui](#) e scegliere l'icona che più vi aggrada. È possibile cambiare il colore dell'icona attraverso la proprietà `color` `color=“coloreDesiderato”`. Alcuni di questi colori possono essere definiti nei file di config ad esempio possiamo usare le parole chiavi `primary` o `success` o `error` per applicare i colori definiti di default oppure personalizzarli a nostro piacimento.