Le guide di Alv

Alessandro Varesi

06 ott 2023

INDICE

Benvenuto nella raccolta delle mie guide personali. Dopo anni si programmazione, studio e divertimento ho deciso di raccogliere in forma leggibile le informazioni che di volta in volta ritengo necessarie.

Gli argomenti principali sono di carattere informatico, come guide all'installazione ed uso di applicazioni, configurazione di sistemi, creazione di applicazioni web, programmazione.

Una delle prime guide sarà proprio quella relativa alla pubblicazione di documentazione proprio come quella che stai leggendo in questo momento.

CAPITOLO 1

HTML - I contenuti delle pagine Web [Prossimamente]

CAPITOLO 2

CSS - lo stile delle pagine Web [Prossimamente]

CAPITOLO 3

Javascript - Programmare il Web [Prossimamente]

CAPITOLO 4

Python - Un linguaggio per tutte le stagioni [Prossimamente]

CAPITOLO 5

Sveltekit e Airtable per la nostra applicazione web [WIP]

Proviamo a creare assieme una applicazione web completa, in gergo definita «full-stack», per la gestione dei libri della nostra libreria di casa.

Per far questo oltre alla pura creazione del programma per il quale utilizziamo il «framework» Sveltekit, abbiamo anche bisogno di gestire un database di dati per il quale useremo direttamente Airtable, ma ne parleremo a tempo debito.

Partiamo con le basi di Sveltekit.

5.1 Prima di iniziare

La guida vuole guidare il lettore nella creazione di una applicazione di esempio, utilizzata per lo più come istruzione per l'utilizzo del framework SvelteKit nelle sue principali funzionalità.

Nasce come raccolta personale di appunti durante la mia attività di studio dello strumento, quindi non vuole essere una guida completa, ma un semplice punto di riferimento per l'utilizzo del framework.

Nella scrittura si presuppone che il lettore conosca i linguaggi base dello sviluppo web, come HTML, CSS, JavaScript.

5.1.1 Predisposizione dell'ambiente di sviluppo

Per poter utilizzare Sveltekit, dobbiamo avere una versione di node.js installata sul nostro computer. In particolare dobbiamo avere la versione *v16.9* o superiore.

Nota: La procedura di installazione di *node.js* non ricade nello scopo della presente guida.

Per maggiori informazioni a riguardo si rimanda alla lettura della pagina dedicata al download del sito ufficiale: <https://nodejs.org/en/download/>.

Accertiamoci che *node.js* sia installato correttamente:

```
$ node -v
v16.16.0
```

Ovviamente il comando serve per verificare la versione installata, e risponderà con la versione corrispondente.

5.1.2 IDE consigliato

Sebbene i file sorgenti siano rappresentati da file testuali, facilmente editabili con un qualsiasi editor di testo, si consiglia vivamente di utilizzare un *Ambiente di Sviluppo Integrato* (IDE).

In particolare mi sento di consigliare Visual Studio Code che, grazie alle funzionalità integrate ed alla vasta disponibilità di estensioni dedicate, facilita enormemente la scrittura e la correzione del codice. Tra le estensioni, ovviamente , consiglio di installare Svelte for VS Code

5.2 Iniziamo

Come per praticamente tutti i programmi basati su framework, il modo più semplice per iniziare è quello di partire da un progetto base di esempio.

5.2.1 Creiamo il codice di esempio

A questo scopo gli sviluppatori di *SvelteKit* hanno predisposto un comando dedicato per la creazione della struttura di programma da cui partire:

```
npm create svelte@latest my-bookshelf
cd my-bookshelf
npm install
npm run dev
```

Il primo comando crea il vero e proprio progetto di esempio nella cartella *my-bookshelf*. Nel farlo richiede alcune informazioni di configurazione del progetto, come ad esempio l'utilizzo o meno di Typescript come linguaggio di programmazione.

Noi procediamo scegliendo le seguenti opzioni:

- «*Skeleton project*» come template: vogliamo partire da un progetto vuoto
- «*No*» all'utilizzo di *Typescript* per il controllo dei tipi
- «*Yes*» per l'utilizzo di *ESLint* per il controllo del codice
- «*No*» per l'uso di *Prettier* per la formattazione automatica del codice
- «*No*» all'uso di *Playwright* per il testing su multi browser

---

Suggerimento: La attivazione della funzionalità di controllo del codice tramite ESLint, integrato correttamente nell'editor utilizzato obbliga a mantenere una coerenza di scrittura. È anche stato scelto di evitare l'uso del formattatore automatico Prettier obbligando così a scrivere il codice corretto fin dall'inizio.

I comandi successivi installano le dipendenze necessarie per il funzionamento del progetto e lo avviano in modalità di sviluppo.

In particolare l'ultimo comando : `npm run dev` avvia il server di sviluppo e rende disponibile l'applicazione web su localhost:5173.

5.2.2 Analizziamo il progetto

SvelteKit è un progetto attualmente ancora in fase di sviluppo e durante il processo di progettazione ha subito differenti modifiche anche significative.

Nota: Questa guida fa riferimento alla versione disponibile al momento della scrittura (agosto 2022) che raccoglie modifiche sostanziali rispetto alle versioni precedenti. Per maggiori informazioni fare riferimento alla guida ufficiale di conversione : Migration guide #5774.

Ci sono due concetti base da tenere conto:

- Ogni pagina dell'applicazione è un *componente* Svelte
- Ogni pagina creata sarà un file aggiunto alla directory `src/routes`.

Capiremo più avanti il significato intrinseco di questi concetti.

Analizziamo insieme i file che sono stati generati dai comandi utilizzati precedentemente, evidenziando quelli che sono parte dei sorgenti del progetto e non artefatti della compilazione.

Nella cartella principale di progetto troviamo:

package.json

Il codice generato dovrebbe essere molto simile a quello riportato di seguito, con eventuali differenze sulle versioni delle dipendenze.

```
{
  "name": "my-bookshelf",
  "version": "0.0.1",
  "private": true,
  "scripts": {
    "dev": "vite dev",
    "build": "vite build",
    "preview": "vite preview",
    "lint": "eslint ."
  },
  "devDependencies": {
    "@sveltejs/adapter-auto": "next",
    "@sveltejs/kit": "next",
    "eslint": "^8.16.0",
    "eslint-plugin-svelte3": "^4.0.0",
    "svelte": "^3.44.0",
    "vite": "^3.1.0"
  },
  "type": "module"
}
```

Si tratta del file di dipendenze del progetto **Node**, e si può notare che allo stato attuale esistono solo dipendenze di sviluppo (`devDependencies`) e nessuna dipendenza di «produzione» (`dependencies`). Questo è una caratteristica

basiliare di **Svelte** che, per quanto possibile trasforma tutto il codice in *javascript* nativo rendendo il progetto portabile e ottimizzato.

Il file `package.json` deve includere `@sveltejs/kit`, `svelte` e `vite` come dipendenze di sviluppo.

Voglio farti notare che il file `package.json` creato contiene la configurazione `"type": module`. Questo fa si che i file `.js` sono interpretati come moduli nativi JavaScript in fase di `import` e `export`.

svelte.config.js

File di configurazione di **Svelte** e **SvelteKit**

vite.config.js

Un progetto **SvelteKit** in realtà si tratta di un progetto Vite che usa il plugin *@sveltejs/kit/vite* assieme a qualsiasi altra configurazione inserita in questo file.

static

In questa cartella sono inseriti gli asset statici che vengono pubblicati dal server in modo diretto; è qui che metteremo file tipo `favicon.png` o `robots.txt`.

src

È in questa cartella che troviamo tutti i fale che definiscono la nostra applicazione.

```
src/
├ routes/
│ └ +page.svelte
└ app.html
```

Per ora il nostro progetto vuooto contiene solo due file:

- `app.html` la pagina template - un documento HTML, vedremo in seguito da cosa è composto
- `+page.svelte` all'interno della cartella `routes` che rappresenta la pagina iniziale del nostro progetto.

CAPITOLO 6

Creare un sito web con Django [Prossimamemente]

CAPITOLO 7

Pubblicare su Read The Docs

Read The Docs è un servizio online dedicato alla pubblicazione di documentazione di varia natura.

Esistono due versioni distinte di servizio:

readthedocs.org
: versione gratuita del servizio, permette la pubblicazione di documenti pubblicamente accessibili

readthedocs.com
: versione premium del servizio, dedicato principalmente ad aziende o corporazioni che necessitano di poter pubblicare in forma ristretta la propria documentazione

Ovviamente per questa guida useremo la versione gratuita più che sufficiente per i nostri scopi.

7.1 Prossimamente su questi schermi...