Introduzione a Vim

a cura di **rceschel**

Obiettivi di oggi

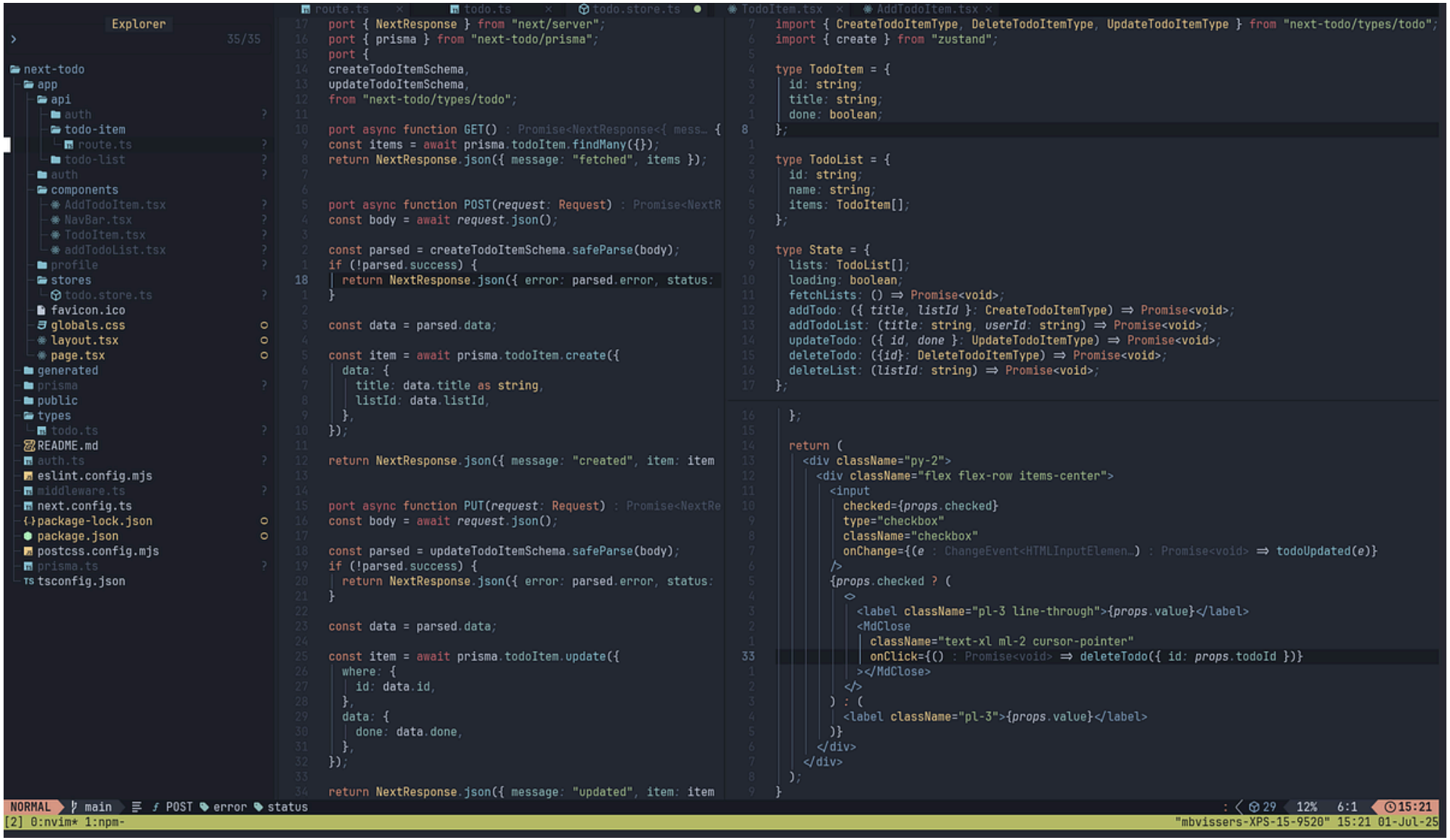
- Abbracciare il paradigma di **editor modale**
- Familiarizzare con Vim e i suoi concetti
- Imparare a muoversi con naturalezza in un file
- Imparare a **personalizzare l'ambiente** di Vim
- Scoprire come approfondire e allenarsi in autonomia

Perché Vim

- È leggero, veloce, sempre presente
- È integrato nel **terminale**, il nostro ambiente
- All'inizio è scarno, ma può essere stravolto

Vim appare vuoto e dispersivo, ma per questo potrà conformarsi ad ogni nostra esigenza.

Esempio di configurazione avanzata



Vim è diverso

Vim **non** è un editor classico.

- Non scrivi sempre
- Non usi menu o mouse
- Non tieni premuti modificatori

Vim funziona per **modalità**.

Editor Modale

Editor classico vs Modale

Classico	Modale
Scrivi sempre	Ti muovi ed esegui azioni
Tasti = caratteri	Tasti = comandi
CTRL / menù	Modalità

Editor modale

Idee fondamentali

- **Digitare ≠ Scrivere**
- Scrivere è solo *una* modalità
- La modalità di lavoro è **Normal**

Visualizza Vim come un **sistema a stati**.

Complesse operazioni sul testo vengono eseguite con combinazioni **sequenziali** di tasti, secondo una logica sintattica, non solo mnemonica.

Questo si traduce in **comodità** ed **efficienza**.

Normal Mode

Una casa per tutti

Muoversi nel file

Movimenti fondamentali:

- `h j k l` → sinistra, giù, su, destra
- `w` → parola successiva
- `e` → fine parola
- `0` → inizio riga
- `$` → fine riga

Muoversi nel file

Movimenti globali:

- `G` → fine del file
- `gg` → inizio del file

Muoversi nel file

Movimenti mirati:

- `:` → entra in command line, digita il numero di riga per muoverti
- `/` → Search: digita un termine per trovarlo nel file
  - `n` → occorrenza successiva
  - `N` → occorrenza precedente

Il panic button

Se non sai cosa sta succedendo:

ESC

- Torni sempre in Normal mode
- Puoi premerlo più volte

Premilo senza paura, o quando ne hai.

Lavorare nel file

Operatori principali

In Normal mode:

- `d` → delete (cancella / taglia)
- `y` → yank (copia)
- `c` → change (cancella e scrivi)
- `p` → put (incolla)

Gli operatori **da soli non fanno nulla**.
Serve un obiettivo.

Vim è un linguaggio

La formula magica:

```
OPERATOR [+ COUNT] + MOTION
```

Traduce in comandi il nostro linguaggio naturale.

- "Delete Word"
- "Yank to the end of line"
- "Delete three lines"

Il modello combinatorio

Comando	Significato
`dw`	delete word
`d$`	delete to end of line
`3dd`	delete 3 lines
`yw`	yank word
`yy`	yank line

Stesso movimento, operatore diverso.

Nota: un operatore ripetuto due volte si applica a tutta la linea.

Insert mode

Finalmente si scrive

Comando	Dove scriverai
`i`	prima del cursore
`I`	inizio riga
`a`	dopo il cursore
`A`	fine riga
`o`	nuova riga sotto
`O`	nuova riga sopra

Uscire sempre con **ESC**.

Visual mode

Selezionare prima di agire

Fase	Azione
1	Entri in Visual
2	Ti muovi selezionando
3	Applichi un operatore

È familiare, ma meno potente del modello base.

Usala per eseguire operazioni complesse, avendo un riscontro visivo prima di applicare le modifiche.

Command line

Comando	Azione
`:w`	salva
`:q`	esci
`:x`	salva ed esci
`:42`	vai alla riga 42

Modalità diversa da Normal e Insert.

Cercare nel file

Comando	Azione
`/test`	cerca avanti
`?test`	cerca indietro
`n`	prossimo risultato
`N`	precedente

Sbagliare è permesso

Comando	Effetto
`u`	undo
`U`	undo su tutta la riga
`<C-r>`	redo
`.`	ripete ultima modifica

Riduce errori e fatica.

Cosa avete visto

- Vim è **modale**
- Normal mode è centrale
- Operatore + movimento
- Undo, ricerca, ripetizione

Questo è **il core di Vim**.

E adesso?

Per praticare:

- `vimtutor`
- `:help`

La velocità arriva **solo con la pratica**.

Personalizzazione

Creare il proprio ambiente

La selezione dello chef

```
" Righe e numeri di riga
set number                              " Abilita i numeri di riga
set relativenumber          " Imposta i numeri relativi (a partire dal cursore)

" Indentazione e tab
set tabstop=4                   " Numero di spazi visualizzati quando Vim incontra un carattere <Tab>
set softabstop=4                " Numero di spazi inseriti o rimossi premendo <Tab> o <Backspace> in modalità inserimento
set shiftwidth=4                " Numero di spazi usati per ogni livello di indentazione (>>, <<, autoindent)

set autoindent                  " Indentazione automatica attiva
set cindent                             " Imposta lo stile C-like di indentazione

" Utilizzo generale
set mouse=a                     " Abilita l'uso del mouse in tutte le modalità
syntax on                               " Abilita l'evidenziazione della sintassi
```