

GABRIELE ROMANATO

Introduzione al Test-Driven Development (TDD) in Go

Il Test-Driven Development (TDD) è una metodologia di sviluppo che prevede la scrittura dei test prima del codice effettivo. In Go, grazie al supporto nativo per i test tramite il pacchetto `testing`, è particolarmente semplice adottare questa pratica.

Cos'è il TDD?

Il TDD segue un ciclo in tre fasi:

- **Red**: scrivere un test che fallisce.
- **Green**: scrivere il codice minimo per far passare il test.
- **Refactor**: migliorare il codice mantenendo i test verdi.

Primo esempio in Go

Supponiamo di voler implementare una funzione che somma due interi. Iniziamo scrivendo il test.

```
package sum

import "testing"

func TestAdd(t *testing.T) {
    result := Add(2, 3)
    expected := 5

    if result != expected {
        t.Errorf("Add(2, 3) = %d; want %d", result, expected)
    }
}
```

Questo test fallirà poiché la funzione `Add` non è ancora stata implementata.

Implementazione minima

Scriviamo ora il codice per far passare il test.

```
package sum

func Add(a, b int) int {
    return a + b
}
```

Ora rieseguendo il test con `go test`, dovrebbe passare.

Strumenti utili

Go fornisce diversi strumenti per il testing:

- `go test` per eseguire i test.
- `-v` per output dettagliato.
- `go test -cover` per visualizzare la copertura del codice.

Conclusione

Il TDD aiuta a progettare codice affidabile e manutenibile. In Go, grazie alla semplicità del linguaggio e alla buona integrazione dei test, è facile adottare questa metodologia fin da subito.