

Coworking App -Backend- Type Assertions

Ivan <<ossan>> Pesenti
Software Developer



HACKERSSEN

POWERED BY



COSA SONO?



HACKERSGEN

POWERED BY
SORINT

Scopo

Il tipo di una variabile può essere:

- ❑ Un tipo **primitivo**: int, string, bool, float32, float64
- ❑ Una **collezione**: array, slice e mappe
- ❑ Una **struct**: definita da noi, dai package importati o dalla standard library
- ❑ Un'**interfaccia**: definita da noi, dai package importati o dalla standard library

Nell'ultimo caso, potremmo avere la necessità di capire il tipo concreto che di fatto quella variabile rappresenta (struct, int, string, pointer, ecc.). In questo caso, la **type assertion** viene in nostro aiuto.



HACKERSGEN

POWERED BY

SORINT

Prima Tipologia: <<Rischiosa>>

1. La variabile **value** viene definita come interfaccia (empty)
2. Gli viene assegnato un valore letterale di tipo string ("**GeeksforGeeks**")
3. La type assertion sulla variabile **value** per inizializzare la variabile **value1** funziona correttamente perchè il tipo sottostante è **string**
4. La type assertion sulla variabile **value** per inizializzare la variabile **value2** fa panicare la goroutine perchè il tipo sottostante non è un **int**

```
package main

import (
    "fmt"
)

// main function
func main() {

    // an interface that has
    // a string value
    var value interface{} = "GeeksforGeeks"

    // retrieving a value
    // of type string and assigning
    // it to value1 variable
    var value1 string = value.(string)

    // printing the concrete value
    fmt.Println(value1)

    // this will panic as interface
    // does not have int type
    var value2 int = value.(int)

    fmt.Println(value2)
}
```



HACKERSGEN

POWERED BY

SORINT

Seconda Tipologia: <<Sicura>>

In questo caso, tutto si svolge come in precedenza, eccetto la type assertion. In questo caso viene usato il cosiddetto “**comma, ok**” idioma che permette di ottenere **due** risultati dalla type assertion:

1. La variabile **value2** che contiene il valore della type assertion o lo zero-value del tipo richiesto (in questo caso “” visto che il tipo è **string**)
2. La variabile di tipo **bool test** che indica se la type assertion è andata a buon fine o meno. Quest’ultima verrà poi usata nel resto della logica del programma

```
package main

import (
    "fmt"
)

// main function
func main() {

    // an interface that has
    // an int value
    var value interface{} = 20024

    // retrieving a value
    // of type int and assigning
    // it to value1 variable
    var value1 int = value.(int)

    // printing the concrete value
    fmt.Println(value1)

    // this will test if interface
    // has string type and
    // return true if found or
    // false otherwise
    value2, test := value.(string)
    if test {

        fmt.Println("String Value found!")
        fmt.Println(value2)
    } else {

        fmt.Println("String value not found!")
    }
}
```



HACKERSGEN

POWERED BY
SORINT

QUALCHE ESERCIZIO DI PRATICA...



HACKERSGEN
POWERED BY
SORINT

THANKS!

@ossan



HACKERSGEN

POWERED BY

