

Coworking App -Backend- Lettura File

Ivan <<**ossan**>> Pesenti
Software Developer



HACKERSSEN

POWERED BY

SORINT

PRIMA O POI
DOVEVAMO PASSARCI...



Infatti...

Leggere i files archiviati nel File System è uno dei task più comuni nella vita quotidiana di ogni programmatore. Quando ci troviamo in questa situazione, Go mette a disposizione varie modalità. Le principali sono:

- ❑ Lettura di un file intero con la funzione **os.ReadFile(string)**. Ideale se si ha la certezza che il file è di dimensione contenute
- ❑ Lettura di un file pezzo per pezzo. A sua volta, questa categoria si suddivide in: lettura file riga per riga, lettura parola per parola e lettura in chunk (blocchi) di lunghezza fissa. Solitamente, vengono utilizzate le funzioni **os.Open(string) (*File, error)** e **bufio.NewScanner(io.Reader) *Scanner**. Questa modalità è adatta alla lettura di file di grandi dimensioni.



HACKERSGEN

POWERED BY
SORINT

Lettura con Caricamento in Memoria

- ❑ Questo tipo di lettura è la più semplice da implementare
- ❑ La funzione **os.ReadFile** si aspetta il path su FileSystem del file da leggere
- ❑ I valori che vengono ritornati sono:
 - ❑ Una **slice** di **bytes** con l'eventuale contenuto del file (altrimenti **nil**)
 - ❑ Un **error** con l'eventuale errore occorso (altrimenti **nil**)
- ❑ Se si usa una versione precedente alla 1.16 la funzione **ReadFile** si trova nel package **ioutil** (che ora è deprecato)

```
package main

import (
    "fmt"
    "log"
    "os"
)

func main() {
    content, err := os.ReadFile("file.txt")
    if err != nil {
        log.Fatal(err)
    }
    fmt.Println(string(content))
}
```



HACKERSGEN

POWERED BY
SORINT

LEGGIAMO QUALCHE
FILE...



HACKERSGEN

POWERED BY
SORINT

THANKS!

@ossan



HACKERSGEN

POWERED BY

