

Convertire un file CSV in XML con Go

La conversione di un file CSV in formato XML può essere necessaria in molte situazioni, come l'integrazione di dati tra applicazioni o la trasformazione di dati per l'elaborazione in sistemi diversi. In questo articolo, vedremo come realizzare un programma in Go per convertire un file CSV in XML.

Supponiamo di avere un file CSV chiamato `data.csv` con il seguente contenuto:

```
name,age,city
Alice,30,New York
Bob,25,San Francisco
Charlie,35,Los Angeles
```

Quindi creiamo il codice principale:

```
package main

import (
    "encoding/csv"
    "encoding/xml"
    "fmt"
    "os"
)

type Record struct {
    Name string `xml:"name"`
    Age  string `xml:"age"`
    City string `xml:"city"`
}

type Records struct {
    XMLName xml.Name `xml:"records"`
    Record  []Record `xml:"record"`
}

func main() {
    // Apri il file CSV
    csvFile, err := os.Open("data.csv")
    if err != nil {
        fmt.Println("Errore nell'apertura del file CSV:", err)
        return
    }
    defer csvFile.Close()

    // Leggi il file CSV
    reader := csv.NewReader(csvFile)
    records, err := reader.ReadAll()
    if err != nil {
        fmt.Println("Errore nella lettura del file CSV:", err)
        return
    }

    // Verifica che ci siano righe
    if len(records) < 2 {
```

```

        fmt.Println("Il file CSV non contiene dati.")
        return
    }

    // Estrarre i dati dal CSV e costruire le strutture XML
    var data Records
    headers := records[0] // Intestazioni

    for _, row := range records[1:] {
        record := Record{
            Name: row[0],
            Age:  row[1],
            City: row[2],
        }
        data.Record = append(data.Record, record)
    }

    // Scrivi il file XML
    xmlFile, err := os.Create("data.xml")
    if err != nil {
        fmt.Println("Errore nella creazione del file XML:", err)
        return
    }
    defer xmlFile.Close()

    encoder := xml.NewEncoder(xmlFile)
    encoder.Indent("", " ")
    err = encoder.Encode(data)
    if err != nil {
        fmt.Println("Errore nella scrittura del file XML:", err)
        return
    }

    fmt.Println("Conversione completata! File XML generato: data.xml")
}

```

Ecco come appare il file data.xml:

```

<records>
  <record>
    <name>Alice</name>
    <age>30</age>
    <city>New York</city>
  </record>
  <record>
    <name>Bob</name>
    <age>25</age>
    <city>San Francisco</city>
  </record>
  <record>
    <name>Charlie</name>
    <age>35</age>
    <city>Los Angeles</city>
  </record>
</records>

```

Conclusione

In questa guida, abbiamo visto come convertire un file CSV in XML utilizzando Go. Questo approccio è altamente scalabile e può essere adattato a diverse esigenze. Se il tuo file CSV ha una struttura diversa, puoi semplicemente aggiornare le definizioni delle strutture Go per adattarle al tuo caso.

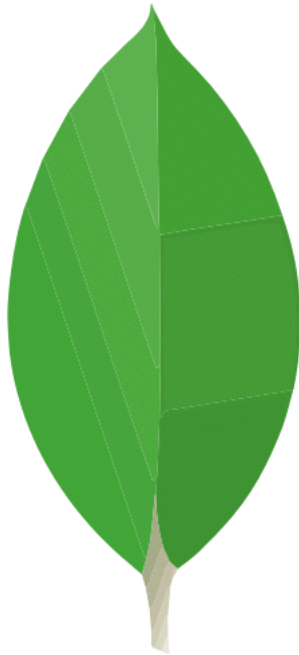
Applicazioni Correlate



-

Go Placeholder Image

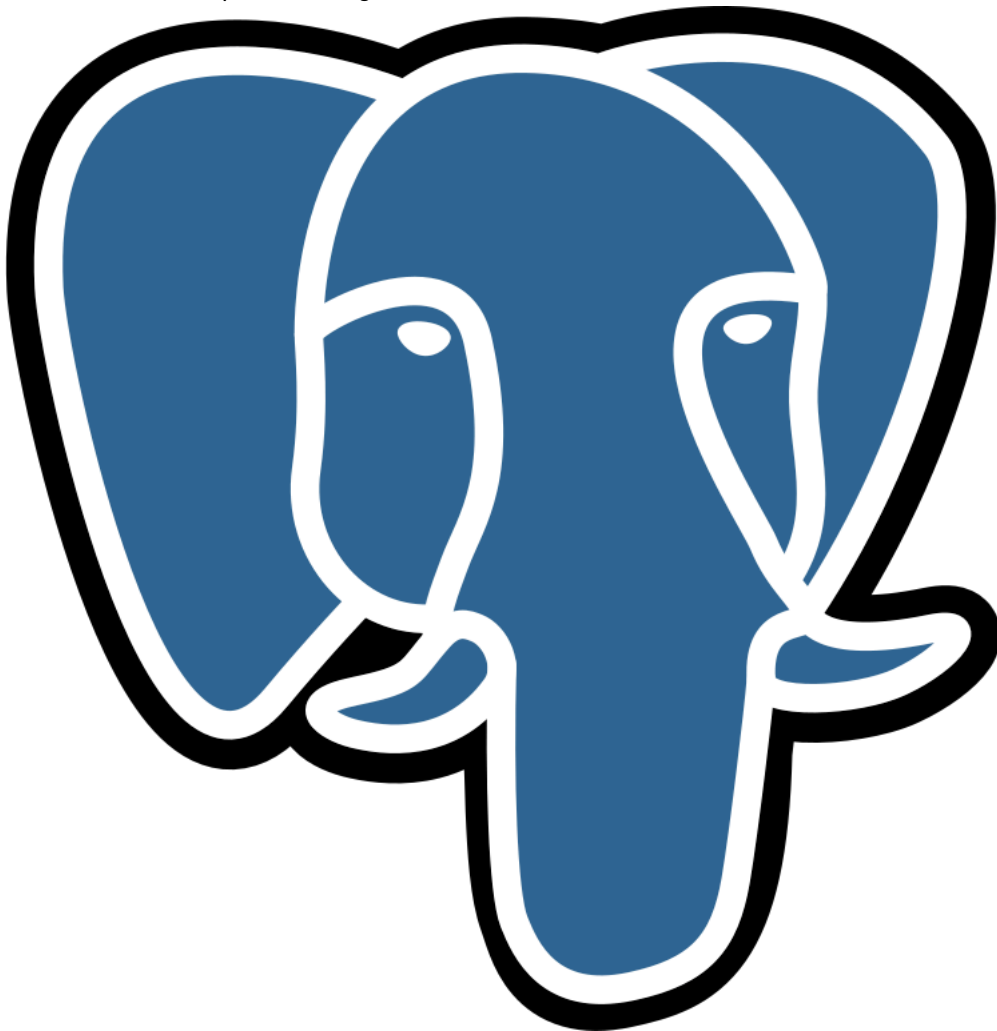
Applicazione in Go per la creazione di immagini segnaposto.
DockerDocker ComposeGoJavaScript



•

Go MongoDB App

Applicazione basata su MongoDB ed implementata in Go con il driver ufficiale.
DockerDocker ComposeGoMongoDB



•

Go PostgreSQL App

Applicazione basata su PostgreSQL e sviluppata in Go.

DockerDocker ComposeGoPostgreSQL