

Go: gestire strutture JSON complesse nelle richieste alle REST API

Come possiamo effettuare il parsing di una struttura JSON arbitraria? In Go si può utilizzare una soluzione alternativa alle struct.

È possibile rappresentare la struttura JSON con una map le cui chiavi sono di tipo stringa e i cui valori sono di tipo `interface`.

```
var results map[string]interface{}
```

Questa rappresentazione può essere ad esempio utilizzata con un'API REST:

```
package main

import (
    "encoding/json"
    "errors"
    "fmt"
    "io"
    "net/http"
    "net/url"

)

func apiRequest(query string)
(map[string]interface{}, error) {
```

```
var results map[string]interface{}

baseUrl := "https://api.unsplash.com"
resource := "/search/photos"

params := url.Values{}
params.Add("query", query)
params.Add("client_id", "api_key")

uri, _ := url.ParseRequestURI(baseUrl)
uri.Path = resource
uri.RawQuery = params.Encode()

urlStr := fmt.Sprintf("%v", uri)

res, err := http.Get(urlStr)

if err != nil {
    return results, err
}

defer res.Body.Close()

if res.StatusCode != 200 {
    return results, errors.New("Request
error")
}

body, e := io.ReadAll(res.Body)

if e != nil {
    return results, e
}
```

```

        jsonData := string(body)

        jsonErr := json.Unmarshal([]byte(jsonData),
&results)

        if jsonErr != nil {
            return results, jsonErr
        }

        return results, nil
    }

func main() {
    results, err := apiRequest("cat")
    if err != nil {
        panic(err)
    }
    fmt.Println(results)
}

```

Nell'esempio stiamo effettuando una ricerca per parola chiave con le API di Unsplash. Il formato di risposta è un oggetto JSON molto complesso che non riusciamo a mappare preventivamente con le struct. La nostra map generica, invece, ci consente di avere comunque una struttura su cui poter eventualmente iterare dopo che la stringa JSON della risposta viene passata come slice di byte alla funzione `Unmarshal()`.

Si tratta in definitiva di una soluzione utile quando la struttura JSON è molto complessa.