



链滴

快速导出多个 csv 文件并下载

作者: [ghqemperor](#)

原文链接: <https://ld246.com/article/1569059150165>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

- 将数据分文件存储，每个 csv 文件一定量条数据，再将 **所有 csv 文件 zip 压缩**
- 再将 zip 文件直接通过网络传输到浏览器，全程 **不经过磁盘**
- (可将demo中生成数据的部分改为数据库查询)

```
package main
```

```
import (
    "archive/zip"
    "bytes"
    "encoding/csv"
    "fmt"
    "net/http"
    "strconv"
    "time"
)

func main() {
    http.HandleFunc("/", func(writer http.ResponseWriter, request *http.Request) {
        fmt.Println(time.Now().Format("15:04:05"))
        zipBuffer := bytes.NewBuffer(nil)
        zipWriter := zip.NewWriter(zipBuffer)
        for i := 0; i < 10; i++ {
            fmt.Printf("zip: %v.csv\n", i+1)
            csvBuffer := Csv()
            // 将数据存入csv文件，并压缩
            csvFile, err := zipWriter.Create(strconv.Itoa(i+1) + ".csv")
            if err != nil {
                panic(err)
            }

            err = csvFile.Write(csvBuffer.Bytes())
            if err != nil {
                panic(err)
            }
        }
        if err := zipWriter.Close(); err != nil {
            panic(err)
        }

        writer.Header().Add("Content-Type", "application/form-data")
        // 设置文件名
        writer.Header().Add("Content-Disposition", fmt.Sprintf(`attachment; filename="%v.zip"`, t
me.Now().Unix()))
        err := zipBuffer.WriteTo(writer)
        if err != nil {
            panic(err)
        }
    })

    if err := http.ListenAndServe(":8080", nil); err != nil {
        panic(err)
    }
}
```

```

func Csv() *bytes.Buffer {
    csvBuffer := bytes.NewBuffer(nil)
    csvWriter := csv.NewWriter(csvBuffer)
    if err := csvWriter.WriteAll(getData()); err != nil {
        panic(err)
    }

    return csvBuffer
}

type SQLValue struct {
    ID      uint
    CreateAt time.Time
    UpdateAt time.Time
    DeleteAt *time.Time

    Username string
    Sex      int // 1.男, 2.女, 3.未知
    Age      int // year
    Height   int // cm
}

// 模拟数据库操作
func getData() [][]string {
    // 此处可为数据库操作
    sqlValue := make([]*SQLValue, 1000000)
    for i := range sqlValue {
        sqlValue[i] = &SQLValue{
            ID:      uint(i),
            CreateAt: time.Now(),
            UpdateAt: time.Now(),
            DeleteAt: nil,
            Username: strconv.Itoa(i),
            Sex:     3,
            Age:     20,
            Height:  170,
        }
    }
}

data := make([][]string, len(sqlValue))
data[0] = []string{
    "ID", "CreateAt", "UpdateAt", "DeleteAt", "Username", "Sex", "Age", "Height",
}

for i, v := range sqlValue {
    if i == 0 {
        continue
    }
    id := strconv.Itoa(int(v.ID))
    createAt := v.CreateAt.Format("2006-01-02 15:04:05")
    updateAt := v.UpdateAt.Format("2006-01-02 15:04:05")
    deleteAt := ""
    if v.DeleteAt != nil {

```

```
    deleteAt = v.DeleteAt.Format("2006-01-02 15:04:05")
}
sex := ""
switch v.Sex {
case 1:
    sex = "男"
case 2:
    sex = "女"
case 3:
    sex = "未知"
default:
}
age := strconv.Itoa(v.Age)
height := strconv.Itoa(v.Height)
data[i] = []string{
    id,
    createAt,
    updateAt,
    deleteAt,
    v.Username,
    sex,
    age,
    height,
}
}
return data
}
```