

各种抽奖活动来一遍系列 (一)

作者: [xiaoxiezaijia](#)

原文链接: <https://ld246.com/article/1567073588569>

来源网站: [链滴](#)

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

项目介绍

在日常生活中，各种抽奖活动都是很常见的，比如年会抽奖，刮刮乐，体育彩票，双色球，支付宝，信中的抢红包，微博抢红包，支付宝集福卡等，这些都大大丰富了我们的生活，当然，我基本上是没到什么奖，年会抽奖也把我漏了。。。 ，下面我就讲带大家来揭秘一下各种抽奖活动的实现，代码将以 golang 实现

抽奖系统的业务介绍

1. 抽奖活动的业务需求，既复杂又多变
2. 奖品类型和改路设置
3. 如何公平抽奖，安全发奖

抽奖系统的技术挑战

1. 网络并发编程,数据读写的并发安全性问题
2. 高效的抽奖和发奖,提高并发性能
3. 系统优化, 这么把 redis 更好的利用起来

抽奖活动之企业抽奖

```
package main

import (
    "fmt"
    "math/rand"
    "net/http"
    "sync"
    "strings"
    "time"
)

//全局变量,存储用户列表
var userList []string
//锁
var mu sync.Mutex

//当前系统当前抽奖人数
func count(w http.ResponseWriter, r *http.Request) {
    count := len(userList)
    countStr := fmt.Sprintf("当前总共参与人数: %d\n", count)

    w.Write([]byte(countStr))
}

//导入用户名单
func importUser(w http.ResponseWriter, r *http.Request) {
    strUsers := "小明,张三,李四,王五,周六,周日,卷福,斯瓦辛格,李小龙"
    users := strings.Split(strUsers, ",")
}
```

```

count1 := len(userList)
for _, u := range users {
    u = strings.TrimSpace(u)
    if len(u) > 0 {
        mu.Lock()
        userList = append(userList, u)
        mu.Unlock()
    }
}

//用户数量
count2 := len(userList)
str := fmt.Sprintf("当前总共参与人数: %d, 成功导入用户数: %d\n", count1, count2)

w.Write([]byte(str))

}

//抽奖
func lucky(w http.ResponseWriter, r *http.Request) {

    count := len(userList)

    if count > 1 {
        //生成随机数
        seed := time.Now().UnixNano()
        index := rand.New(rand.NewSource(seed)).Int31n(int32(count))
        user := userList[index]

        //重新组织 userList 删除取出的那个
        mu.Lock()
        userList = append(userList[0:index], userList[index+1:]...)
        mu.Unlock()

        w.Write([]byte("当前中奖用户: " + user))
        fmt.Sprintf("当前中奖用户: %s\n", user)
    }

}

func main() {

    //切片
    userList = []string{}

    //初始化 锁
    mu = sync.Mutex{}

    //定义访问路由
    http.HandleFunc("/", lucky)
    http.HandleFunc("/import", importUser)
    http.HandleFunc("/count", count)

    //监听端口,启动服务器

```

```
http.ListenAndServe(":8080", nil)  
}
```