



《手写OS操作系统》小班二期招生，全程直播授课，大牛带你掌握硬核技术！

点此查看

首页 > 慕课教程 > Go工程师体系课全新版 > 2. go的配置文件管理-viper

全部开发者教程

9. 服务器命名规范

10. 内网穿透

11. 项目文档和资源

第13周 库存服务和分布式锁

1. 库存服务架构设计

2. 库存服务表结构

3. 为什么需要分布式锁

4. mysql的for update实现悲观锁

5. 基于redis实现分布式锁

6. redlock详解

7. 常见的分布式锁实现方案

第14周 订单和购物车微服务

1. 表结构设计和proto文件定义



bobby · 更新于 2022-11-08

◀ 上一节 1. go 日志库 zap 3. 自定义验证器 下一节 ▶

1. 介绍

Viper是适用于Go应用程序的完整配置解决方案。它被设计用于在应用程序中工作，并且可以处理所有类型的配置需求和格式。它支持以下特性：

- 设置默认值
- 从 JSON、 TOML、 YAML、 HCL、 envfile 和 Java properties 格式的配置文件读取配置信息
- 实时监控和重新读取配置文件（可选）
- 从环境变量中读取
- 从远程配置系统（etcd或Consul）读取并监控配置变化
- 从命令行参数读取配置
- 从buffer读取配置
- 显式配置值

2. yaml教程

[教程地址](#)

3. 安装

[viper项目地址](#)

<> 代码块

```
1 go get github.com/spf13/viper
```

4. 将配置文件映射成struct

<> 代码块

```
1 package main
2
3 import (
4     "fmt"
5     "github.com/spf13/viper"
6 )
7
8
9 type ServerConfig struct{
10     ServiceName string `mapstructure:"name"`
11     Port int `mapstructure:"port"`
```

意见反馈

收藏教程

标记书签

```
13
14 func main(){
15     v := viper.New()
16     //文件的路径如何设置
17     v.SetConfigFile("viper_test/ch01/config.yaml")
18     if err := v.ReadInConfig(); err != nil{
19         panic(err)
20     }
21     serverConfig := ServerConfig{}
22     if err := v.Unmarshal(&serverConfig); err != nil{
23         panic(err)
24     }
25     fmt.Println(serverConfig)
26     fmt.Printf("%V", v.Get("name"))
27 }
28
```

配置文件config.yaml

<> 代码块

```
1 name: 'user-web'
2 port: 8021
```

5. 使用

<> 代码块

```
1 package main
2
3 import (
4     "fmt"
5     "github.com/fsnotify/fsnotify"
6     "github.com/spf13/viper"
7     "time"
8 )
9
10 type MysqlConfig struct {
11     Host string `mapstructure:"host"`
12     Port int `mapstructure:"port"`
13 }
14 type ServerConfig struct{
15     Name string `mapstructure:"name"`
16     MysqlInfo MysqlConfig `mapstructure:"mysql"`
17 }
18
19 func GetEnvInfo(env string) string {
20     viper.SetEnvPrefix(env)
21     return viper.GetString(env)
22 }
23
24 func main(){
25     data := GetEnvInfo("Debug")
26     var configFileName string
27     configFileNamePrefix := "config"
28     if data == "true" {
29         configFileName = fmt.Sprintf("viper_test/%s-debug.yaml", configFileNamePrefix)
30     }else{
31         configFileName = fmt.Sprintf("viper_test/%s-pro.yaml", configFileNamePrefix)
32     }
33
34     serverConfig := ServerConfig{
```

意见反馈

收藏教程

标记书签



```
37
38     v := viper.New()
39     v.SetConfigFile(configFileName)
40     err := v.ReadInConfig()
41     if err != nil {
42         panic(err)
43     }
44     if err := v.Unmarshal(&serverConfig); err != nil {
45         panic(err)
46     }
47     fmt.Println(serverConfig)
48
49     go func() {
50         v.WatchConfig()
51         v.OnConfigChange(func(e fsnotify.Event) {
52             fmt.Println("Config file changed:", e.Name)
53             _ = v.ReadInConfig() // 读取配置数据
54             _ = v.Unmarshal(&serverConfig)
55             fmt.Println(serverConfig)
56         })
57     }()
58
59     time.Sleep(time.Second*3000)
60
61 }
```

配置文件内容

<> 代码块

```
1     name: 'user-web'
2     mysql:
3         host: '127.0.0.1'
4         port: 3306
```

手机号码正则表达式

<> 代码块

```
1     ^1([38][0-9]|14[579]|5[^4]|16[6]|7[1-35-8]|9[189])\d{8}$
```

1. go 日志库 zap ◀ 上一节

下一节 ▶ 3. 自定义验证器

✎ 我要提出意见反馈

