



《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

从所有教程的词条中查询...

首页 > 慕课教程 > Go工程师体系课全新版 > 1. go 日志库 zap



bobby · 更新于 2022-11-08

上一节 8. 优雅重启或停止 2. go的配置文... 下一节

zap官方地址

## 1. 安装和基本使用

<> 代码块

```
1 go get -u go.uber.org/zap
```

<> 代码块

```
1 package main
2
3 import (
4     "go.uber.org/zap"
5 )
6
7 func main() {
8     logger, _ := zap.NewProduction()
9     defer logger.Sync() // flushes buffer, if any
10    url := "https://imooc.com"
11    sugar := logger.Sugar()
12    sugar.Infow("failed to fetch URL",
13        // Structured context as loosely typed key-value pairs.
14        "url", url,
15        "attempt", 3,
16    )
17    sugar.Infof("Failed to fetch URL: %s", url)
18 }
19
```

Zap提供了两种类型的日志记录器— **Sugared Logger** 和 **Logger** 。

在性能很好但不是很关键的上下文中，使用 **SugaredLogger** 。它比其他结构化日志记录包快4-10倍，并且支持结构化和printf风格的日志记录。

在每一微秒和每一次内存分配都很重要的上下文中，使用 **Logger** 。它甚至比 **SugaredLogger** 更快，内存分配次数也更少，但它只支持强类型的结构化日志记录

## 2. 写入日志文件

<> 代码块

```
1 package main
2
3 import (
4     "go.uber.org/zap"
5     "time"
6 )
7
8 func main() {
9     // Create a logger that writes to a file
10    logger, err := zap.NewProduction()
11    if err != nil {
12        panic(err)
13    }
14    defer logger.Sync()
15    url := "https://imooc.com"
16    sugar := logger.Sugar()
17    sugar.Infow("failed to fetch URL",
18        "url", url,
19        "attempt", 3,
20    )
21    sugar.Infof("Failed to fetch URL: %s", url)
22 }
```

意见反馈

收藏教程

标记书签



```
8
9 func NewLogger() (*zap.Logger, error) {
10     cfg := zap.NewProductionConfig()
11     cfg.OutputPaths = []string{
12         "./myproject.log",
13     }
14     return cfg.Build()
15 }
16
17 func main() {
18     //logger, _ := zap.NewProduction()
19     logger, err := NewLogger()
20     if err != nil {
21         panic(err)
22         //panic("初始化logger失败")
23     }
24     su := logger.Sugar()
25     defer su.Sync()
26     url := "https://imooc.com"
27     su.Info("failed to fetch URL",
28         // Structured context as strongly typed Field values.
29         zap.String("url", url),
30         zap.Int("attempt", 3),
31         zap.Duration("backoff", time.Second),
32     )
33 }
```