xlog日志库说明

特性

xlog是基于Golang标准库log的增强日志库，除支持log库的所有功能外，还提供了大量新特性，包括：

- 日志高亮(两种高亮方式)
- 调用者信息(包名.函数名)
- 日志级别(ERROR, WARN, INFO, DEBUG, TRACE)
- 更简单的调用方式(自动格式化)
- 日志异步持久化(单文件+多文件轮换)

用法

基本用法

最简单的使用方式是使用全局默认日志实例：

```
1  package main
2  import "./xlog"
3  func main() {
4    xlog.Info("Hello Golang")
5  }
```

```
✿ ebtech✦xshrim :: ~/git/minicode/admission(master|✗) ✶ » go run main.go
2020/06/29 19:50:00 [INFO] Hello Golang
```

全局默认日志实例的各项参数支持自定义：

```
1  package main
2  import "./xlog"
3  func main() {
4    xlog.SetLevel(xlog.TRACE)
5    xlog.SetFlag(xlog.Ldate | xlog.Ltime | xlog.Lshortfile | xlog.Lcolor)
6    xlog.Debug("Hello Golang")
7  }
```

```
✿ ebtech✦xshrim :: ~/git/minicode/admission(master|✗) ✶ » go run main.go
2020/06/29 19:52:16 main.go:8: [DEBUG] Hello Golang
```

此外用户也可以自行创建日志实例，或基于该实例作进一步封装：

```go
package main
import "os"
import "./xlog"
func main() {
  logger := xlog.New(os.Stdout, "[MyPrefix]", xlog.INFO, xlog.Ldate | xlog.Ltime | xlog.Lfullcolor)
  logger.Error("Hello Golang")
}
```

```
ebtech❖xshrim :: ~/git/minicode/admission(master|x) * » go run main.go
[MyPrefix]2020/06/29 19:55:05 [ERROR] Hello Golang
```

§主要特性

◊日志高亮

xlog支持**无高亮**，**仅Level高亮**和**全高亮**的方式显示日志，两种高亮方式的启用方式为在日志实例的`flag`字段加入`Lcolor`或`Lfullcolor`项，如：`xlog.SetFlag(xlog.Ldate | xlog.Ltime | xlog.Lfullcolor)`，效果分别如下：

```
ebtech❖xshrim :: ~/git/minicode/admission(master|x) * » go run main.go
2020/06/29 19:57:46 main.go:10: [DEBUG] Hello Golang
2020/06/29 19:57:46 main.go:12: [WARN] Hello Golang
2020/06/29 19:57:46 main.go:14: [INFO] Hello Golang
```

◊调用者信息

xlog在log库支持显示调用文件及所在行的基础上，还支持显示调用函数及其所在package，此特性需要在日志实例的`flag`字段加入`Lstack`项，如：`xlog.SetFlag(xlog.Ldate | xlog.Ltime | xlog.Lstack | xlog.Lshortfile | xlog.Lfullcolor)`，效果如下：

```
ebtech❖xshrim :: ~/git/minicode/admission(master|x) * » go run main.go
2020/06/29 20:01:35 <main.test> main.go:8: [DEBUG] Hello Golang
```

◊日志级别

xlog支持**ERROR**，**WARN**，**INFO**，**DEBUG**，**TRACE**五种日志级别，通过`SetLevel`函数设置可显示级别后，低于该级别的日志将不打印，如：

```go
package main
import "./xlog"
func main() {
  xlog.SetLevel(xlog.INFO)
  xlog.SetFlag(xlog.Ldate | xlog.Ltime | xlog.Lshortfile | xlog.Lstack | xlog.Lcolor)
  xlog.Warn("Debug Message")   // 打印
  xlog.Trace("Trace Message")  // 不打印
}
```

```
✿ ebtech✦xshrim :: ~/git/minicode/admission(master|✘) ✶ » go run main.go
2020/06/29 20:02:57 <main.main> main.go:13: [WARN] Debug Message
```

◊简化调用

调用xlog打印日志的方法分别为`Error`，`Warn`，`Info`，`Debug`和`Trace`，此外也支持`Fatal`和`Panic`两种异常方法．所有方法均能够自动完成格式化输出，也能够像标准fmt库一样支持可变参数，如：

```
1  package main
2  import "./xlog"
3  func main() {
4    xlog.SetLevel(xlog.INFO)
5    xlog.SetFlag(xlog.Ldate | xlog.Ltime | xlog.Lshortfile | xlog.Lstack |
   xlog.Lfullcolor)
6    xlog.Warn("Warn: %s", "unsafe")
7    xlog.Error("Error: division by ", 0, xlog.Err("error message"))
8    xlog.Fatal("exit")
9  }
```

xlog的日志打印相关函数均是换行打印，如需不换行打印，可使用`Log`函数，用法为：`xlog.Log("debug", "print %s without newline", "text")`．

此外，xlog还内置了`Print`，`Println`，`Sprint`和`Err`这些常用函数，分别用于不换行打印，换行打印，返回字符串和返回错误．类似功能的函数在fmt库中也存在，xlog内置的这些函数支持自动格式化输出，而且免于再import fmt．

```
✿ ebtech✦xshrim :: ~/git/minicode/admission(master|✘) ✶ » go run main.go
2020/06/29 20:04:06 <main.main> main.go:13: [WARN] Warn: unsafe
2020/06/29 20:04:06 <main.main> main.go:14: [ERROR] Error: division by 0 error message
2020/06/29 20:04:06 <main.main> main.go:15: [FATAL] exit
exit status 1
```

◊日志持久化

xlog支持将日志持久化输出到日志文件中，既可以指定单个日志文件，也可以指定日志目录和相关轮换参数进行日志文件的自动轮换．而且允许将日志同时输出到标准输出设备和日志文件中．xlog的日志持久化操作是异步完成的．使用方法为通过`NewLogSaverWithLogFile`或`NewLogSaverWithRotation`获取LogSaver并通过`SetSaver`绑定到日志实例．

```go
package main
import "./xlog"
func main() {
  xlog.SetLevel(xlog.TRACE)
  xlog.SetFlag(xlog.Ldate | xlog.Ltime | xlog.Lstack | xlog.Lshortfile | xlog.Lfullcolor)
  xlog.SetSaver(xlog.NewLogSaverWithRotation("./", 1024*1024, 3)) // 参数依次为：轮换日志目录，日志文件大小上限，轮换日志数量
  xlog.Info("日志持久化")
  defer xlog.Flush() // 保证日志全部落盘
}
```

[注意]：由于日志持久化是异步完成的，因此为了保证所有日志都写入文件中，建议在程序退出前调用Flush函数

此外，xlog同样支持按照log库的方式实现日志持久化，即将`*os.File`对象作为日志实例的`io.Writer`。

```
✿ ebtech✣xshrim :: ~/git/minicode/admission(master|✗) ✷ » go run main.go
2020/06/29 20:10:37 <main.main> /home/xshrim/git/minicode/admission/main.go:14: [INFO] 日志持久化
✿ ebtech✣xshrim :: ~/git/minicode/admission(master|✗) ✷ » \cat main.0.log
2020/06/29 20:10:37 <main.main> /home/xshrim/git/minicode/admission/main.go:14: [INFO] 日志持久化
```