

golang 目录遍历

获取目录列表用 `ioutil.ReadDir()`，遍历目录用 `filepath.Walk()`，使用方法请参考文章示例。

示例代码：

```
package main

import (
    "fmt"
    "io/ioutil"
    "os"
    "path/filepath"
    "strings"
)

//获取指定目录下的所有文件，不进入下一级目录搜索，可以匹配后缀过滤。
func ListDir(dirPth string, suffix string) (files []string, err error) {
    files = make([]string, 0, 10)

    dir, err := ioutil.ReadDir(dirPth)
    if err != nil {
        return nil, err
    }

    PthSep := string(os.PathSeparator)
    suffix = strings.ToUpper(suffix) //忽略后缀匹配的大小写

    for _, fi := range dir {
        if fi.IsDir() { // 忽略目录
            continue
        }
        if strings.HasSuffix(strings.ToUpper(fi.Name()), suffix) { //匹配文件
            files = append(files, dirPth+PthSep+fi.Name())
        }
    }

    return files, nil
}

//获取指定目录及所有子目录下的所有文件，可以匹配后缀过滤。
func WalkDir(dirPth, suffix string) (files []string, err error) {
    files = make([]string, 0, 30)
    suffix = strings.ToUpper(suffix) //忽略后缀匹配的大小写
```

```
err = filepath.Walk(dirPth, func(filename string, fi os.FileInfo, err error) error { //遍历目录
    //if err != nil { //忽略错误
    //    return err
    //}

    if fi.IsDir() { // 忽略目录
        return nil
    }

    if strings.HasSuffix(strings.ToUpper(fi.Name()), suffix) {
        files = append(files, filename)
    }

    return nil
})

return files, err
}

func main() {
    files, err := ListDir("D:\\Go", ".txt")
    fmt.Println(files, err)

    files, err = WalkDir("E:\\Study", ".pdf")
    fmt.Println(files, err)
}
```

TAG

[golang \(/article/category/golang\)](/article/category/golang)[php \(/article/category/php\)](/article/category/php)[nginx \(/article/category/nginx\)](/article/category/nginx) [js \(/article/category/js\)](/article/category/js)[linux \(/article/category/linux\)](/article/category/linux)[android \(/article/category/android\)](/article/category/android)[jquery \(/article/category/jquery\)](/article/category/jquery)

[mysql \(/article/category/mysql/\)](/article/category/mysql/)

[测试 \(/article/category/测试/\)](/article/category/测试/)

[memcache \(/article/category/memcache/\)](/article/category/memcache/)

最 热 文 章

[go 中url编码和字符转码\(类似php中的urlencode 和 htmlspecialchars\) \(/article/55.html\)](/article/55.html)

[golang 字符串截取 \(/article/6.html\)](/article/6.html)

[golang中的并发 sync.WaitGroup 和channel \(/article/77.html\)](/article/77.html)

[golang 环境的搭建和安装 \(/article/33.html\)](/article/33.html)

[golang 字符转义 \(/article/27.html\)](/article/27.html)

[golang 随机字符串 \(/article/7.html\)](/article/7.html)

[golang 抓取网页信息（用goquery） \(/article/25.html\)](/article/25.html)

[golang map排序 \(/article/54.html\)](/article/54.html)

[android 横屏、竖屏切换禁止 \(/article/79.html\)](/article/79.html)

[nginx 出现a client request body is buffered to a temporary file 错误 \(/article/46.html\)](/article/46.html)

最 新 文 章

[golang模板语法简明教程 \(/article/92.html\)](/article/92.html)

[nginx rewrite 参数和例子 \(/article/91.html\)](/article/91.html)

[关于全局唯一ID生成方法 \(/article/90.html\)](/article/90.html)

[PHP升级导致系统负载过高问题分析 \(/article/89.html\)](/article/89.html)

[go语言channel的别样用法 \(/article/88.html\)](/article/88.html)

[Nginx ssi 设置 \(/article/87.html\)](/article/87.html)

[Linux tcpdump命令详解 \(/article/86.html\)](/article/86.html)

linux之sed用法 (/article/85.html)

php 导出excel , 同样稍微修改一下适合其他语言

(/article/84.html)

golang 目录遍历 (/article/83.html)

© Copyright © 2013-2014

网站地图 (<http://coolaf.com/article/m>)



(<http://tongji.baidu.com/hm-web/welcome/ico?s=18f18be6e58f13d87192835c8c15fdca>)