

# dogo

---

当源文件发生改变时，自动重新编译并运行(或重启)。适用于开发服务端程序时快速调试。

## 特点

---

- 当监控目录的源文件发生改变时，自动重新编译，并运行(重启)
- 同时监控多个目录，包括子文件夹
- 可同时运行多个实例，互不影响
- 详细日志输出
- 占用内存,CPU 资源少

## 安装

---

```
go get github.com/liudng/dogo
```

## 创建配置文件

---

dogo 的配置文件格式如下:

```
{
  "WorkingDir": "{GOPATH}/src/github.com/liudng/dogo/example",
  "SourceDir": [
    "{GOPATH}/src/github.com/liudng/dogo/example"
  ],
  "SourceExt": ".go|.c|.cpp|.h",
  "BuildCmd": "go build github.com/liudng/dogo/example",
  "RunCmd": "example.exe"
}
```

**WorkingDir:** 工作目录, dogo 会自动切换到此目录.

**SourceDir:** 监控源文件目录清单.

**SourceExt:** 监控的文件类型.

**BuildCmd:** 编译命令.

**RunCmd:** 运行命令.

# 开始监控

输入下面的命令(如果当前目录下存在 dogo.json 文件, 会自动载入):

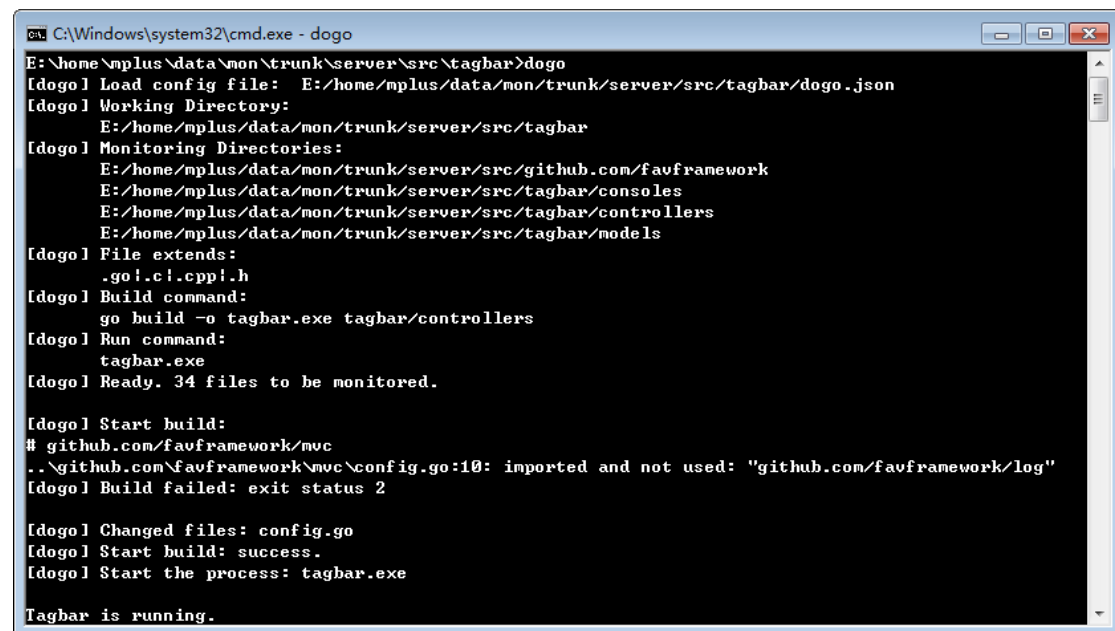
```
dogo
```

或者用 -c 参数指定配置文件路径:

```
dogo -c=/path/to/dogo.json
```

文件路径允许包含{GOPATH}, dogo 会自动替换为环境变量 GOPATH 的值。

## screen capture



```
C:\Windows\system32\cmd.exe - dogo
E:\home\mplus\data\mon\trunk\server\src>tagbar>dogo
[dogo] Load config file: E:/home/mplus/data/mon/trunk/server/src/tagbar/dogo.json
[dogo] Working Directory:
E:/home/mplus/data/mon/trunk/server/src/tagbar
[dogo] Monitoring Directories:
E:/home/mplus/data/mon/trunk/server/src/github.com/favframework
E:/home/mplus/data/mon/trunk/server/src/tagbar/containers
E:/home/mplus/data/mon/trunk/server/src/tagbar/controllers
E:/home/mplus/data/mon/trunk/server/src/tagbar/models
[dogo] File extends:
.go!.c!.cpp!.h
[dogo] Build command:
go build -o tagbar.exe tagbar/controllers
[dogo] Run command:
tagbar.exe
[dogo] Ready. 34 files to be monitored.

[dogo] Start build:
# github.com/favframework/mvc
..\github.com\favframework\mvc\config.go:10: imported and not used: "github.com/favframework/log"
[dogo] Build failed: exit status 2

[dogo] Changed files: config.go
[dogo] Start build: success.
[dogo] Start the process: tagbar.exe

Tagbar is running.
```

## 问题反馈

在使用中遇到任何问题, 请通过 <https://github.com/liudng/dogo/issues/new> 提出。