

Golang 介绍

Go 语言是谷歌推出的一种全新的编程语言，可以在不损失应用程序性能的情况下降低代码的复杂性。谷歌首席软件工程师罗布派克(Rob Pike)说：我们之所以开发 Go，是因为过去 10 多年间软件开发的难度令人沮丧。

谷歌资深软件工程师罗布·派克(Rob Pike)表示，“Go 让我体验到了从未有过的开发效率。”派克表示，和今天的 C++或 C 一样，Go 是一种系统语言。他解释道，“使用它可以进行快速开发，同时它还是一个真正的编译语言，我们之所以现在将其开源，原因是我们认为它已经非常有用和强大。”

Golang 语言的一些的功能：

具有现代的程序语言特色，如垃圾回收，帮助程序设计师处理琐碎和重要的内存管理等问题。Go 的速度也非常快，几乎和 C 或 C++程序一样快，且能够快速制作程序。

该软件是专为构建服务器软件所设计（如 Google 的 Gmail），Google 认为 Go 还可应用到其他领域，包括在浏览器内执行软件，取代 JavaScript 的角色。

Go 也可解决现今的一大挑战：多核心处理器。一般电脑程序通常依序执行，一次进行一项工作，但多核心处理器更适合并行处理许多工作。

相较于其他语言，Golang 之所以发展迅速，与该语言特有的特色密不可分：

- 简洁 快速 安全
- 并行 有趣 开源
- 内存管理，数组安全，编译迅速

Iris 简介

Iris 是一款 Go 语言中用来开发 web 应用的框架，该框架支持编写一次并在任何地方以最小的机器功率运行，如 Android、ios、Linux 和 Windows 等。该框架只需要一个可执行的服务就可以在平台上运行了。

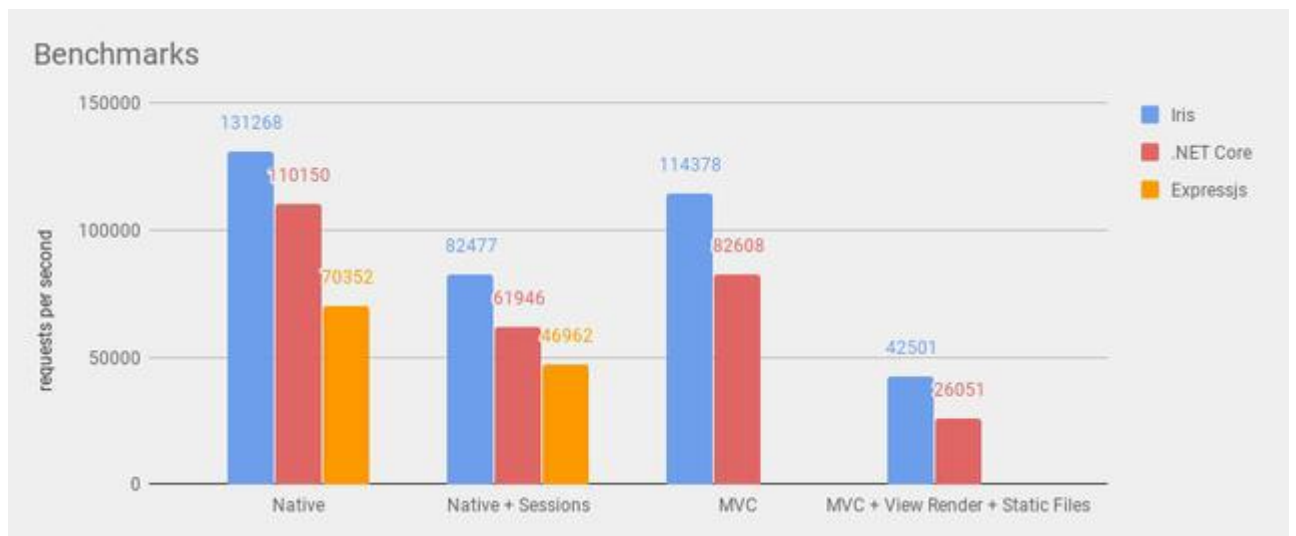
Iris 框架以简单而强大的 api 而被开发者所熟悉。iris 除了为开发者提供非常简单的访问方式外，还同样支持 MVC。另外，用 iris 构建微服务也很容易。

在 iris 框架的官方网站上，被称为速度最快的 Go 后端开发框架。在 Iris 的网站文档上，列出了该框架具备的一些特点和框架特性，列举如下：

- 1) 聚焦高性能
- 2) 健壮的静态路由支持和通配符子域名支持
- 3) 视图系统支持超过 5 以上模板
- 4) 支持定制事件的高可扩展性 Websocket API

- 5) 带有 GC, 内存 & redis 提供支持的会话
- 6) 方便的中间件和插件
- 7) 完整 REST API
- 8) 能定制 HTTP 错误
- 9) 源码改变后自动加载

等等还有很多特性, 大家可以参考 Iris 官方文档。在 GoWeb 开发的诸多框架中, 各个维度的性能比较如下:



go (1.11)	muxie (1.0)	6.99 ms	5.51 ms	10.93 ms
go (1.11)	iris (10.7)	7.08 ms	5.80 ms	11.45 ms
csharp (7.3)	aspnetcore (2.1)	6.71 ms	5.91 ms	9.62 ms
go (1.11)	gin (1.3)	8.05 ms	6.27 ms	12.93 ms
go (1.11)	echo (3.3)	7.82 ms	6.30 ms	12.93 ms
go (1.11)	beego (1.10)	7.87 ms	6.50 ms	12.96 ms
go (1.11)	gorilla-mux (1.6)	7.95 ms	6.64 ms	13.20 ms

Iris 框架学习渠道

在学习 Iris 框架过程时，我们需要有相应的配套资料才能够完成我们的学习。下面是学习 Iris 框架过程中可能会用到的资料。

- 1) Iris 官网: <https://iris-go.com/>
- 2) Iris 框架源码地址: <https://github.com/kataras/iris>
- 3) Iris 框架中文学习文档: <https://studyiris.com/doc/>

当然，还有其他相关资料，比如说 xorm 框架等，这些我们在后面的课程文档中学习以后再做解释和说明。

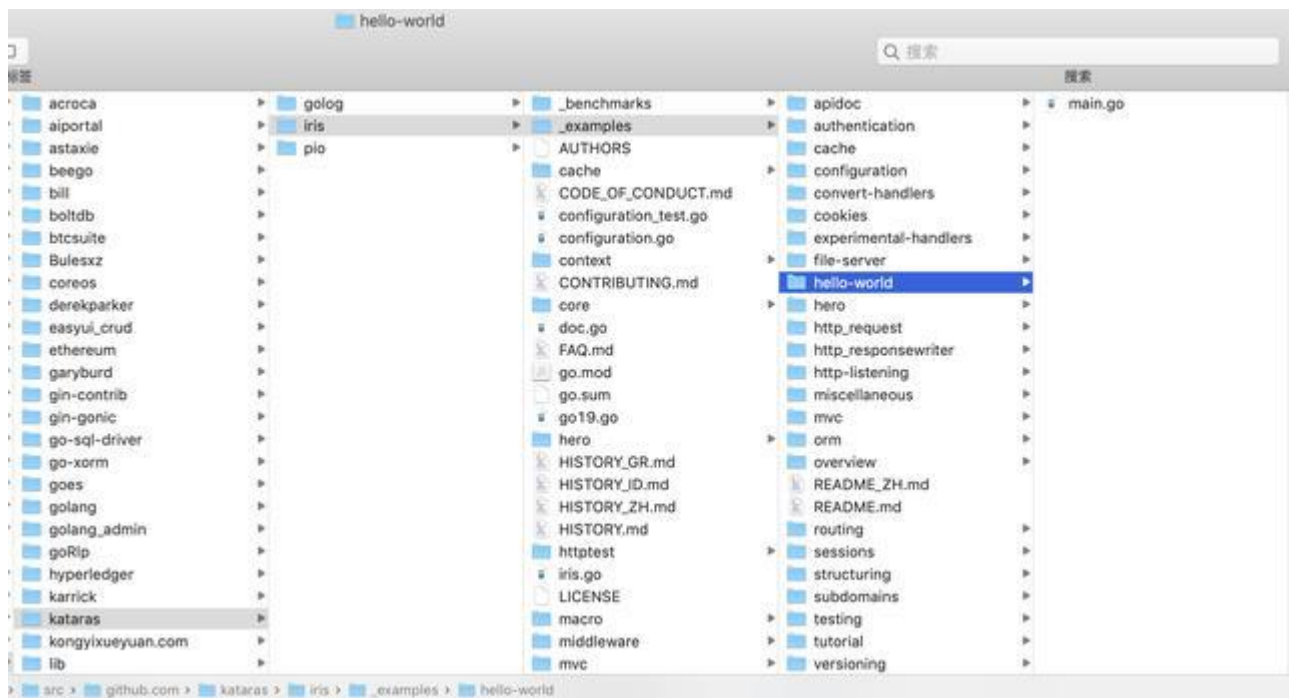
Iris 框架安装

环境要求: iris 框架要求 go 语言版本至少为 1.8。各位同学可以通过打开终端，执行: `go version` 命令来查看自己机器的 go 环境版本。

安装 Iris 框架非常简单，使用 go 语言的全局安装第三方代码的命令 `get` 即可。安装 Iris 框架的命令如下：

```
go get -u github.com/kataras/iris
```

在本地终端中执行如上的安装命令，等待命令执行成功，即表示 Iris 源码下载安装完成。安装完成 Iris 框架以后，能够在本地机器的 GoPath 环境目录中的 `src/github.com/` 目录下找到 iris 框架对应的包名，如下图所示：



如图所示的 kataras/iris 所在的目录就是 iris 框架的源码, 如上图所示即是安装成功。

源码案例

在 iris 源码安装完成以后, iris 框架为开发者提供了自己学习的实战案例, 供开发者自行学习。iris 提供的案例在 iris 框架目录中的 _example 目录下, 在学习时可以进行参考。

接下来我们就可以开始学习 iris 的相关内容, 进行代码开发了。

Iris 构造服务实例

在安装完成 Iris 的源码后, 我们就开始来编写最简单的一个 Iris 的服务。在 Iris 中, 构建并运行一个服务实例需要两步:

- 1、通过 iris.New() 方法可以实例化一个应用服务对象 app
- 2、通过 Run 方法开启端口监听服务, 运行服务实例

如下是一个最简单的服务案例 Demo

```
package main

import "github.com/kataras/iris"

func main() {

    //1. 创建 app 结构体对象

    app := iris.New()

    //2. 端口监听

    app.Run(iris.Addr(":7999"),

    iris.WithoutServerError(iris.ErrServerClosed))

    ///application.Run(iris.Addr(":8080"))//第一种

    //application.Run(iris.Addr(":8080"),

    iris.WithoutServerError(iris.ErrServerClosed)) //第二种
```

