



# Go for C# developers

Ibraheem Osama (Ibby)

Senior Consultant - Readify Sydney

5X Microsoft Most Valuable Professional


@IbraheemOM



**Microsoft®**  
Most Valuable  
Professional



# Agenda

- Language specifications.
- Go concurrency.
- Why I like Go
- Why I don't like go 
- Open discussion about Go and C#.



**Microsoft®**  
Most Valuable  
Professional



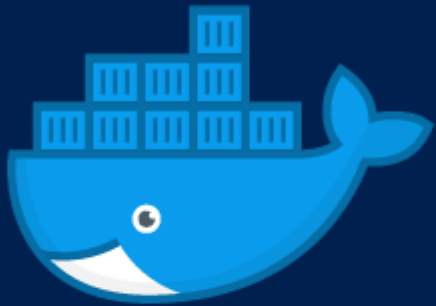
# What is Go ?

- Created by Google.
- C style System Programming Language (Unlike C# general puprpose).
- Open Source, Cross-platform.
- Statically typed, Compiled language.
- Memory safety, Garbage collection.



**Microsoft®**  
Most Valuable  
Professional

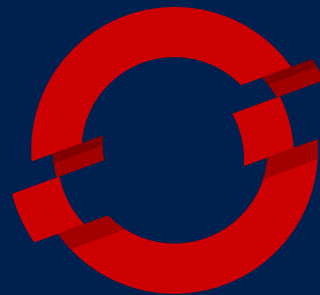
# Famous OS Projects



docker



kubernetes



**RED HAT®**  
**OPENSIFT**  
Container Platform



**Microsoft®**  
Most Valuable  
Professional



# Selected Language Specs



**Microsoft®**  
Most Valuable  
Professional

# Packages



- .vscode
- constants
- dao
- helpers
- internal
- models
- services
- GO app.go
- debug
- Dockerfile
- Dockerfile-test
- README.md
- TicketSales.exe

You, 11 days ago | 1 author (You)

```
1 package dao
2
3 import (
4     "TicketSales/constants"
5     "TicketSales/helpers"
6     "TicketSales/models"
7     "errors"
8     "log"
9
10    "gopkg.in/mgo.v2"
11    "gopkg.in/mgo.v2/bson"
12 )
```

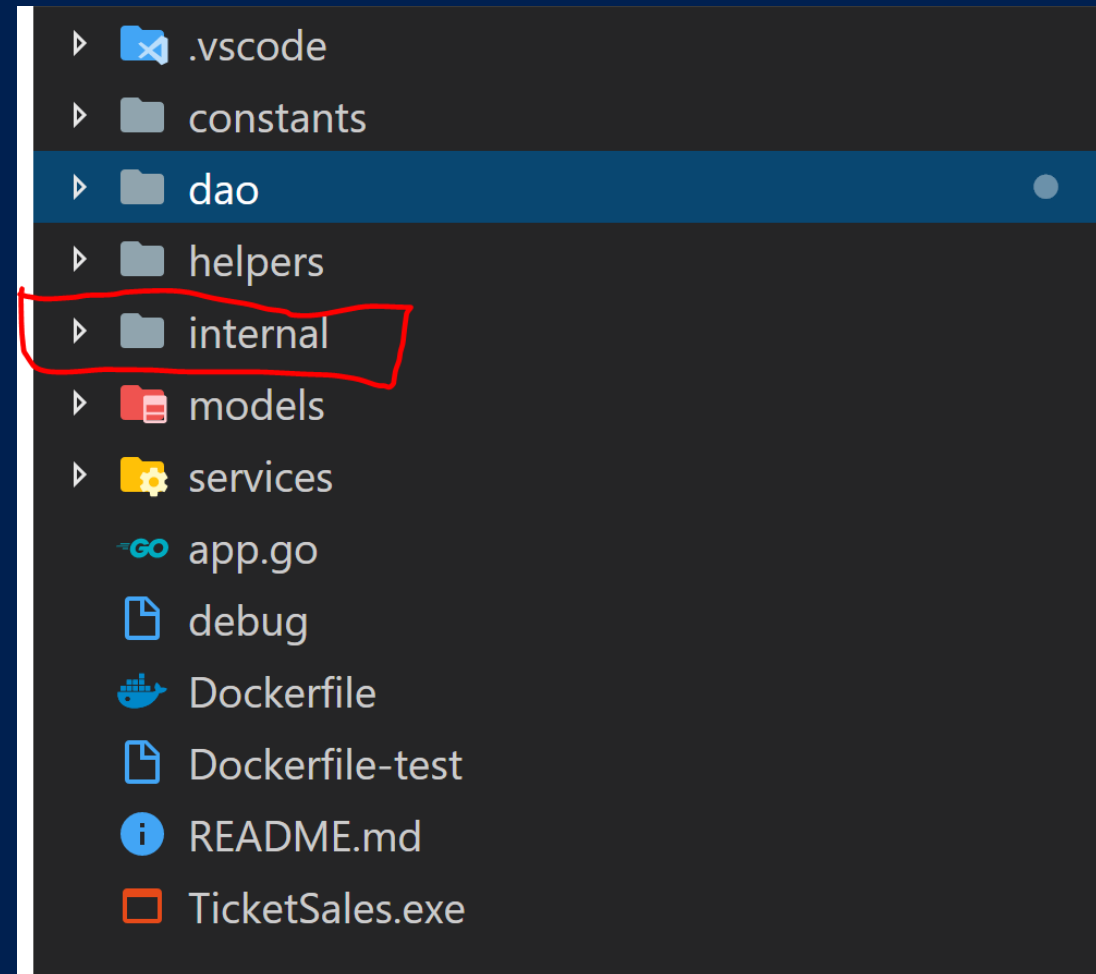


**Microsoft®**  
Most Valuable  
Professional



# Private/Public/Internal

```
You, a few seconds ago | 1 author (You)
1 package dao
2
3 import (
4     "TicketSales/constants"
5     "TicketSales/helpers"
6     "TicketSales/models"
7     "errors"
8     "log"
9     "gopkg.in/mgo.v2"
10    "gopkg.in/mgo.v2/bson"
11
12 )
13
14 var x = ""
15 var X = ""
16
```





# Structs (No classes ☹️)

You, 11 days ago | 1 author (You)

```
type MoviesDAO struct {  
    mainSession *mgo.Session  
    Config       helpers.IConfig  
}
```



**Microsoft®**  
Most Valuable  
Professional





# Functions & Methods(Receiver)

```
func registerRoutes(router *mux.Router, service services.IRegisterRouterService) {  
    service.RegisterService(router)  
}
```

```
func (m *MoviesDAO) FindAll(page int, count int) ([]models.Movie, error) {  
    if page == 0 {  
        return nil, errors.New("Paging index must begin with 1")  
    }  
    session := m.getMainSession().Copy()  
    defer session.Close()  
    var db = session.DB(m.Config.GetMongoDatabase())  
  
    var movies []models.Movie  
    err := db.C(constants.MoviesCollectionName).Find(bson.M{}).Skip((page - 1)  
    (&movies)  
    return movies, err  
}
```



**Microsoft®**  
Most Valuable  
Professional



# Interfaces

You, 6 days ago | 1 author (You)  
`package helpers`

You, 6 days ago | 1 author (You)  
`type IConfig interface {  
 GetMongoServer() string  
 GetMongoDatabase() string  
 GetHttpServerPort() string  
 GetInfoServiceUrl() string  
}`



# Interfaces



```
func (c *Config) GetMongoServer() string {
    if len(c.mongoServer) == 0 {
        c.mongoServer = os.Getenv("MongoServer")
    }
    return c.mongoServer
}

// Read and parse the configuration file
func (c *Config) GetMongoDatabase() string {
    if len(c.mongoDatabase) == 0 {
        c.mongoDatabase = os.Getenv("MongoDatabase")
    }
    return c.mongoDatabase
}

func (c *Config) GetHttpServerPort() string {
    if len(c.httpServerPort) == 0 {
        c.httpServerPort = ":" + os.Getenv("InfoServerPort")
    }
    return c.httpServerPort
}

func (c *Config) GetInfoServiceUrl() string {
    if len(c.infoServiceUrl) == 0 {
        c.infoServiceUrl = os.Getenv("InfoServerUrl")
    }
    return c.infoServiceUrl
}
```



# Interfaces

```
var config = helpers.Config{}  
var c helpers.IConfig  
c = &config  
c.GetHttpServerPort()
```





# Variables

```
var i, j, k int
```

```
x := 1
```

```
movies, err := service.Dao.FindAll(page, count)
```



Microsoft®  
Most Valuable  
Professional



# Package Management

```
Install packages `go get github.com/BurntSushi/toml gopkg.in/mgo.v2 github.com/gorilla/mux`
```

Edit User Variable ✕

Variable name:


Variable value:





Microsoft®  
Most Valuable  
Professional



# Unit testing

▲  services

 movies\_service\_test.go

 movies\_service.go



Microsoft®  
Most Valuable  
Professional



# Unit testing

```
run test | debug test
func TestMain(m *testing.M) {
    setup()
    code := m.Run()
    shutdown()
    os.Exit(code)
}
```







# Unit testing

You, 13 days ago | 1 author (You) | run package tests | run file tests  
package helpers

```
import "testing"
```

run test | debug test

```
func TestGetHttpServerPort(t *testing.T) {  
    var config = Config{}  
    var httpServerPort = config.GetHttpServerPort()  
  
    if httpServerPort[0] != ':' {  
        t.Errorf("Go http server must start with")  
    }  
}
```



**Microsoft**  
Most Valuable  
Professional



# Specific Tests

```
4
run test | debug test
5 func TestIntegration_AllMoviesEndPoint_ReadMovies_PagingWorking(t *testing.T) {
5 |     // Arrange
```

go test -run TestIntegration ./...



Microsoft®  
Most Valuable  
Professional



# Dependency Injection

```
package inject
```

```
import "github.com/facebookgo/inject"
```

You, 13 days ago | 1 author (You)

```
type MoviesService struct {  
    Dao          *dao.MoviesDAO    `inject:""`  
    ResponseHelper *helpers.ResponseHelper `inject:"`  
}
```



Microsoft®  
Most Valuable  
Professional



# Defer (using)

```
func (service MoviesService) CreateMovieEndPoint(w http.ResponseWriter, r
*http.Request) {
    defer r.Body.Close()
```



**Microsoft**  
Most Valuable  
Professional



# Get started with main

You, 15 days ago | 1 author (You)

```
package main
```

```
▶ func init() {           You, 20 days ago • Adding movies rest service in go lang  
  
    }  
  
▶ func main() {  
  
▶ }
```



**Microsoft®**  
Most Valuable  
Professional



# GO Concurrency

Do not communicate by sharing memory; instead, share memory by communicating.



**Microsoft®**  
Most Valuable  
Professional



# GoRoutines

```
func say(s string) {  
    for i := 0; i < 5; i++ {  
        time.Sleep(100 * time.Millisecond)  
        fmt.Println(s)  
    }  
}
```

```
func main() {  
    go say("world")  
    say("hello")  
}
```

You, a few seconds ago • Uncommitted changes



Microsoft®  
Most Valuable  
Professional

# Channels



```
func sum(a []int, c chan int) {  
    sum := 0  
    for _, v := range a {  
        sum += v  
    }  
    c <- sum // send sum to c  
}
```

```
func main() {  
    a := []int{7, 2, 8, -9, 4, 0}  
  
    c := make(chan int)  
    go sum(a[:len(a)/2], c)  
    go sum(a[len(a)/2:], c)  
    x, y := <-c, <-c // receive from c  
  
    fmt.Println(x, y, x+y)
```

```
}
```

You, a few seconds ago • Uncommitted ch



Microsoft®  
Most Valuable  
Professional





# Buffered Channel

```
func main() {  
    c := make(chan int, 2)  
    c <- 1  
    c <- 2  
    fmt.Println(<-c)  
    fmt.Println(<-c)  
}
```

You, a few seconds ago •





Doesn't it remind you with Actor Model ?



**Microsoft®**  
Most Valuable  
Professional



# Why Go ?

- Easy to get started.
- There is no learning curve 😊.
- Concurrency features.
- Go generates binaries with dependencies.
- No Semicolon.



**Microsoft®**  
Most Valuable  
Professional



# Go Criticism

- No Generics.
- No Exceptions.
- No inheritance.
- No overloading.
- No this keyword.
- Single active workspace at a time.
- Poor dependency management.



**Microsoft®**  
Most Valuable  
Professional



# Questions ?



**Microsoft®**  
Most Valuable  
Professional



Thanks 😊

Ibraheem Osama  
@IbraheemOM



**Microsoft®**  
Most Valuable  
Professional



# Mini Hack

## Exercise Link

<https://github.com/IbraheemOsama/GoSession/tree/ToDo>

If you have any question please reach out to Suds or Me



**Microsoft®**  
Most Valuable  
Professional



# Final Code

<https://github.com/IbraheemOsama/GoSession>

Ibraheem Osama  
@IbraheemOM



**Microsoft**  
Most Valuable  
Professional