kubebuilder 项目开发 (Windows)

AUTHOR: 彭玲 TIME: 2021/9/29

环境准备

Go 环境

```
C:\Users\pengl>go version
go version go1.16.7 windows/amd64

# Windows 开启 Go Modules 功能
C:\Users\pengl>set GO111MODULE=on

# 通过 go env 验证 GO111MODULE 设置
C:\Users\pengl>go env | find "GO111MODULE"
set GO111MODULE=on
```

GoLand 设置

对 GoLand IDE 环境进行 `Go Modules` 相关设置：

- 启用 go modules 功能。
- 设置 Go 代理 `GOPROXY=https://goproxy.cn,https://proxy.golang.org,direct`。

项目打包传输

在 Linux 平台下创建 `guestbook` 项目，将该项目打包，并传输到 Windows 平台。

```
# 打包
pengling@FSZJ-PENGLING:~$ tar -zcvf ~/myprojects/guestbook.tar.gz -C ~/myprojects guestbook
...
pengling@FSZJ-PENGLING:~/myprojects$ ll -t
total 15604
drwxrwxr-x  5 pengling pengling     4096    Sep 14 14:41    ./
-rw-rw-r--  1 pengling pengling  15955570    Sep 14 14:39    guestbook.tar.gz
drwxr-xr-x 12 pengling pengling     4096     Sep 14 14:33    ../
drwxrwxr-x  7 pengling pengling     4096    Sep 13 16:32    guestbook/
...
# 传输至 Windows 平台
pengling@FSZJ-PENGLING:~/myprojects$ sz guestbook.tar.gz
```

GoLand 开发项目

使用 GoLand 打开`guestbook`目录，该 IDE 根据`go.mod`文件自动下载并安装项目依赖的包。

`go.mod`文件

```
module my.domain/guestbook

go 1.13

require (
    github.com/go-logr/logr v0.1.0
    github.com/onsi/ginkgo v1.11.0
    github.com/onsi/gomega v1.8.1
    k8s.io/apimachinery v0.17.2
    k8s.io/client-go v0.17.2
    sigs.k8s.io/controller-runtime v0.5.0
)

```

运行`main.go`

编译失败：`SchemeBuilder.Register(&Guestbook{}, &GuestbookList{})`中`Guestbook`和`GuestbookList`未实现`runtime.Object`接口。

```
GOROOT=C:\Program Files\Go #gosetup
GOPATH=C:\Users\pengl\go #gosetup
"C:\Program Files\Go\bin\go.exe" build -o C:\Users\pengl\AppData\Local\Temp\___go_build_main_go.exe E:/iFS-DevOps/guestbook/main.go #gosetup
go: my.domain/guestbook/controllers: package github.com/go-logr/logr imported from implicitly required module; to add missing requirements, run:
    go get github.com/go-logr/logr@v0.1.0

Compilation finished with exit code 1
```

新建`deep_copy.go`

创建`api/v1/deep_copy.go`文件，**解决**编译错误：`Guestbook`和`GuestbookList`未实现`runtime.Object`接口。

```go
package v1

import "k8s.io/apimachinery/pkg/runtime"

// Guestbook implement runtime.Object (DeepCopyObject method)
func (in *Guestbook) DeepCopyObject() runtime.Object {
    if c := in.DeepCopy(); c != nil {
        return c
    }
    return nil
}

func (in *Guestbook) DeepCopy() *Guestbook {
    if in == nil {
        return nil
    }
    out := new(Guestbook)
    in.DeepCopyInto(out)
    return out
}

func (in *Guestbook) DeepCopyInto(out *Guestbook) {
    *out = *in
    out.TypeMeta = in.TypeMeta
    in.ObjectMeta.DeepCopyInto(&out.ObjectMeta)
    out.Spec = in.Spec
    out.Status = in.Status
}

// GuestbookList implement runtime.Object (DeepCopyObject method)
func (in *GuestbookList) DeepCopyObject() runtime.Object {
    if c := in.DeepCopy(); c != nil {
        return c
    }
    return nil
}

func (in *GuestbookList) DeepCopy() *GuestbookList {
    if in == nil {
        return nil
    }
    out := new(GuestbookList)
    in.DeepCopyInto(out)
    return out
}

func (in *GuestbookList) DeepCopyInto(out *GuestbookList) {
    *out = *in
    out.TypeMeta = in.TypeMeta
    in.ListMeta.DeepCopyInto(&out.ListMeta)
    if in.Items != nil {
        in, out := &in.Items, &out.Items
```

```
        *out = make([]Guestbook, len(*in))
        for i := range *in {
            (*in)[i].DeepCopyInto(&(*out)[i])
        }
    }
}

```

项目打包

`guestbook` 项目编译通过后，将 `guestbook` 项目打包，并传输至 node38 服务器，以部署项目。

```
julin@FSZJ-PENGLING:~/myproject$ scp guestbook.tar
anxin@10.8.30.38:/home/anxin/pengling/k8s/kubebuilder
anxin@10.8.30.38's password:
guestbook.tar       100%   37MB  11.0MB/s   00:03
```