

code-generator介绍

如何操作自定义资源

`client-go` 为每种K8S内置资源提供对应的 `clientset` 和 `informer`。那如果我们要监听和操作自定义资源对象，应该如何做呢？这里我们有两种方式：

- 方式一：使用 `client-go` 提供的 `dynamicClient` 来操作自定义资源对象，当然由于 `dynamicClient` 是基于 `RESTClient` 实现的，所以也可以使用 `RESTClient` 来达到同样的目的。
- 方式二：使用 `code-generator` 来帮助我们生成我们需要的代码，这样我们就可以像使用 `client-go` 为K8S内置资源对象提供的方式监听和操作自定义资源了。

code-generator

[code-generator](#)是K8S官方提供的一组代码生成工具，它主要有两个应用场景：

- 为CRD编写自定义controller时，可以使用它来生成我们需要的 `versioned client`、`informer`、`lister` 以及其他工具方法
- 编写自定义API Server时，可以用它来 `internal` 和 `versioned` 类型的转换 `defaulters`、`internal` 和 `versioned` 的 `clients` 和 `informers`

我们本套课程只会涉及为CRD编写自定义controller的场景。

code-generator(续)

1. 获取code-generator的代码，并切换到[v0.23.3](#)的tag上

```
git checkout v0.23.3
```

2. 编译项目，安装代码生成工具，这里我们只安装我们接下来会用到的工具

```
go install code-generator/cmd/{client-gen,lister-gen,informer-gen,deepcopy-gen}
```

code-generator(续)

3. 使用工具code-generator/generate-groups.sh

```
code-generator/generate-groups.sh deepcopy,client,informer MOD_NAME/pkg/generated MOD_NAME/pkg/apis foo.example.com:v1 --output-base MOD_DIR/.. --go-header-file "code-generator/hack/boilerplate.go.txt"
```

常用code-generator标记

deepcopy相关标记

```
//关闭  
// +k8s:deepcopy-gen=false  
//打开  
// +k8s:deepcopy-gen=true  
//生成DeepCopyObject方法  
// +k8s:deepcopy-gen:interfaces=k8s.io/apimachinery/pkg/runtime.Object
```

常用code-generator标记(续)

clientset,lister,informer相关标记

```
// +genclient
// +genclient:noStatus
//cluster级别的
// +genclient:nonNamespaced
// +genclient:noVerbs
// +genclient:onlyVerbs=create,delete
// +genclient:skipVerbs=get,list,create,update,patch,delete,deleteCollection,watch
// +genclient:method=Create,verb=create,result=k8s.io/apimachinery/pkg/apis/meta/v1.Status
```

常用code-generator标记(续)

包级别标记,定义在doc.go

```
// +k8s:deepcopy-gen=package  
// +groupName=foo.example.com  
package v1
```


demo

<https://github.com/kubernetes/code-generator>

谢谢