

# CRD介绍

# 如何使用

在K8S系统扩展点中，开发者可以通过CRD（CustomResourceDefinition）来扩展K8S API，其功能主要由APIExtensionServer负责。使用CRD扩展资源分为三步：

- **注册自定义资源：**开发者需要通过K8S提供的方式注册自定义资源，即通过CRD进行注册，注册之后，K8S就知道我们自定义资源的存在了，然后我们就可以像使用K8S内置资源一样使用自定义资源（CR）
- **使用自定义资源：**像内置资源比如Pod一样声明资源，使用CR声明我们的资源信息
- **删除自定义资源：**当我们不再需要时，可以删除自定义资源

# 注册自定义资源

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  # 名字必需与下面的 spec 字段匹配, 并且格式为 '<名称的复数形式>.<组名>'
  name: demos.example.com
spec:
  # 组名称, 用于 REST API: /apis/<组>/<版本>
  group: example.com
  names:
    # 名称的复数形式, 用于 URL: /apis/<组>/<版本>/<名称的复数形式>
    plural: demos
    # 名称的单数形式, 作为命令行使用时和显示时的别名
    singular: demo
    # kind 通常是单数形式的帕斯卡编码 (PascalCased) 形式。你的资源清单会使用这一形式。
    kind: Demo
    # shortNames 允许你在命令行使用较短的字符串来匹配资源
    shortNames:
      - dm
  # 可以是 Namespaced 或 Cluster
  scope: Namespaced
  # 列举此 CustomResourceDefinition 所支持的版本
  versions:
    - name: v1
      # 每个版本都可以通过 served 标志来独立启用或禁止
      served: true
      # 其中一个且只有一个版本必需被标记为存储版本
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                name:
```

# 使用自定义资源

待CRD创建完成之后，我们就可以使用它来创建我们的自定义资源了，其创建方式跟内置的资源如Pod这些是一样的，只是需要将 **kind**、**apiVersion** 指定为我们CRD中声明的值，比如使用上面例子中的CRD定义资源：

```
apiVersion: "demos.example.com/v1"
kind: Demo
metadata:
  name: crd-demo
spec:
  name: test
```

# Finalizers

Finalizer 能够让控制器实现异步的删除前（Pre-delete）回调。与内置对象类似，定制对象也支持 Finalizer

```
apiVersion: "example.com/v1"  
kind: Demo  
metadata:  
  finalizers:  
    - example.com/finalizer
```

# 合法性验证

```
properties:  
  name:  
    type: string  
    pattern: '^test$'
```

# 附加字段

```
....  
  schema:  
    ...  
  additionalPrinterColumns:  
  - name: Name  
    type: string  
    description: The name of resource  
    jsonPath: .spec.name  
    ...
```

# 子资源

CRD仅支持status和scale子资源

```
...
  schema:
    ...
    subresources:
      # status 启用 status 子资源
      status: {}
      # scale 启用 scale 子资源
      scale:
        # specReplicasPath 定义定制资源中对应 scale.spec.replicas 的 JSON 路径
        specReplicasPath: .spec.replicas
        ...
```



# 设置默认值

```
properties:  
  name:  
    type: string  
    default: "demo"
```

# 多版本

```
...
versions:
  ...
  conversion:
    strategy: Webhook
    webhook:
      conversionReviewVersions: ["v1", "v1beta1"]
      clientConfig:
        service:
          namespace: default
          name: example-conversion-webhook-server
          path: /crdconvert
          caBundle: "Ci0tLS0tQk...<base64-encoded PEM bundle>...tLS0K"
```

# 删除自定义资源

当我们不再需要时，可以像删除其他资源一样，删除我们的自定义资源。

[更多信息，请阅读官方文档](#)

谢谢