

作为公共组件的 **apiserver**

Kubernetes的一些基础组件

- `klog`: 日志
- `utils`: 辅助代码
- `apiserver`: kubernetes的apiserver库
- `client-go`: 客户端
- `component-base`: 共享代码
- `apimachinery`: kubernetes api对象处理

日志

`klog` fork 自 `glog`, 优势

1. 轻量级
2. 方便测试
3. 扩展性强
4. 与 `glog` 兼容性强

Metrics

apiserver中预定义的 metrics

```
// apiserver/pkg/endpoints/metrics/metrics.go`  
  
* apiserver_request_total  
* apiserver_longrunning_gauge  
* apiserver_request_duration_seconds  
* apiserver_response_sizes  
* apiserver_dropped_requests_total  
* apiserver_registered_watchers
```

Example

```
func InstrumentHandlerFunc(
    verb, resource...string,
    handler http.HandlerFunc) http.HandlerFunc {
    return func(w http.ResponseWriter,
        req *http.Request) {
        now := time.Now()
        d := &ResponseWriterDelegator{
            ResponseWriter: w
        }
        handler(w, req)
        MonitorRequest(req, verb, resource,
            d.Header().Get("Content-Type"),
            d.Status(),
            d.ContentLength(),
            time.Since(now))}
}
```

Feature Gates

用于功能开关，类似我们的 Weblab。插件化架构

```
--feature-gates mapStringBool    A set of key=value pairs
    APIListChunking=true|false (BETA - default=true)
    DryRun=true|false (BETA - default=true)
    DynamicAuditing=true|false (ALPHA - default=false)
    WatchBookmark=true|false (ALPHA - default=false)
    ...
```

Interface

```
// compoent-base/featuregate/feature_date.go
type FeatureGate interface {
    Enabled(key Feature) bool
    KnownFeatures() []string
    DeepCopy() MutableFeatureGate
}

type MutableFeatureGate interface {
    FeatureGate
    AddFlag(fs *pflag.FlagSet)
    Set(value string) error
    SetFromMap(m map[string]bool) error
    Add(features map[Feature]FeatureSpec) error
}
```

Audit

从 Request 中提取出先关信息并记录，类似我们 Jakiro 中的事件

- 从 Request/Reponse 中提取出结构化信息
- 输出到不同的存储中

RequestInfo

```
// apiserver/pkg/endpoints/request/requestinfo.go

type RequestInfoResolver interface {
    NewRequestInfo(req *http.Request) (*RequestInfo, error)
}

type RequestInfo struct {
    IsResourceRequest bool
    Path string
    Verb string
    APIPrefix string
    APIGroup string
    APIVersion string
    Namespace string
    Resource string
    Subresource string
    Name string
}
```

Attributes

```
// apierver/pkg/authorization/authorizer/interfaces.go
```

```
type Attributes interface {  
    GetUser() user.Info  
    GetVerb() string  
    IsReadOnly() bool  
    GetNamespace() string  
    GetResource() string  
    GetSubresource() string  
    GetName() string  
    GetAPIGroup() string  
    GetAPIVersion() string  
    IsResourceRequest() bool  
    GetPath() string  
}
```

Event

```
// apiserver/pkg/audit/request.go  
  
func NewEventFromRequest(req *http.Request,  
                        level auditinternal.Level,  
                        attrs authorizer.Attributes)  
    (*auditinternal.Event, error)  
  
// apiserver/pkg/endpoints/filters/audit.go  
func WithAudit(handler http.Handler, sink audit.Sink,  
              policy policy.Checker, longRunningCheck  
              request.LongRunningRequestCheck)  
    http.Handler
```

Sink

```
// apiserver/pkg/audit/types.go  
type Sink interface {  
    ProcessEvents(events ...*auditinternal.Event) bool  
}  
  
type Backend interface {  
    Sink  
    Run(stopCh <-chan struct{}) error  
    Shutdown()  
    String() string  
}
```

存储

- 资源下沉
- DB -> ETCD
- Watch

Interface

```
//apiserver/pkg/storage/interfaces.go
type Interface interface {
    Versioner() Versioner
    Create(ctx context.Context, key string,
          obj, out runtime.Object, ttl uint64) error
    Delete(ctx context.Context, key string,
          out runtime.Object,
          preconditions *Preconditions) error
    Watch(ctx context.Context, key string,
          resourceVersion string,
          p SelectionPredicate) (watch.Interface, error)
    Get(ctx context.Context, key string,
        resourceVersion string,
        objPtr runtime.Object,
        ignoreNotFound bool) error
    ...
}
```

Admission

修改，校验 Request 数据

- 内置 admission controller
- Initializers
- MutatingAdmissionWebhook/ValidatingAdmissionWebhook

Usage

```
--enable-admission-plugins=Initializers,  
                             NamespaceLifecycle,  
                             LimitRanger,  
                             ServiceAccount,  
                             DefaultStorageClass,  
                             DefaultTolerationSeconds,  
                             ResourceQuota
```

```
// example of initializers  
metadata:  
  creationTimestamp: 2019-01-09T08:56:36Z  
  generateName: echo-7cfbbd7d49-  
  initializers:  
    pending:  
      - name: podimage.example.com
```


Interfaces

```
//apiserver/pkg/admission/interfaces.go
```

```
type Interface interface {  
    // create/update...  
    Handles(operation Operation) bool  
}
```

```
type ValidationInterface interface {  
    Interface  
    Validate(a Attributes, o ObjectInterfaces) error  
}
```

```
type MutationInterface interface {  
    Interface  
    Admit(a Attributes, o ObjectInterfaces) error  
}
```

Admission webhooks

```
// apiserver/pkg/admission/plugin/webhook/generic/interfac
type Dispatcher interface {
    // Dispatch a request to the webhooks using the gi
    Dispatch(ctx context.Context,
             a *VersionedAttributes,
             o admission.ObjectInterfaces,
             hooks []*v1beta1.Webhook) error
}

// Source can list dynamic webhook plugins.
type Source interface {
    Webhooks() []*v1beta1.Webhook
    HasSynced() bool
}
```

其他

各个库都有很多可以复用的代码

Leader Election

- package:

```
client-go/tools/leaderelection/leaderelection.go
```

- 用于kubernetes ControllerManager / Scheduler选主
- 使用 ConfigMap / Endpoint 作为 ResourceLock

client-go/utils

- `retry`: 重试操作
- `workqueue`: `queue/reate_limit_queue`
- `flowcontrol`: 流量控制,