api

Schema of the external API types that are served by the Kubernetes API server.

Purpose

This library is the canonical location of the Kubernetes API definition. Most likely interaction with this repository is as a dependency of client-go.

It is published separately to avoid diamond dependency problems for users who depend on more than one of `k8s.io/client-go`, `k8s.io/apimachinery`, `k8s.io/apiserver` …

Recommended Use

We recommend using the go types in this repo. You may serialize them directly to JSON.

If you want to store or interact with proto-formatted Kubernetes API objects, we recommend using the "official" serialization stack in `k8s.io/apimachinery`. Directly serializing these types to proto will not result in data that matches the wire format or is compatible with other kubernetes ecosystem tools. The reason is that the wire format includes a magic prefix and an envelope proto. Please see: https://kubernetes.io/docs/reference/using-api/api-concepts/#protobuf-encoding

For the same reason, we do not recommend embedding these proto objects within your own proto definitions. It is better to store Kubernetes objects as byte arrays, in the wire format, which is self-describing. This permits you to use either JSON or binary (proto) wire formats without code changes. It will be difficult for you to operate on both Custom Resources and built-in types otherwise.

Compatibility

Branches track Kubernetes branches and are compatible with that repo.

Where does it come from?

`api` is synced from https://github.com/kubernetes/kubernetes/blob/master/staging/src/k8s.io/api. Code changes are made in that location, merged into `k8s.io/kubernetes` and later synced here.

Things you should *NOT* do

1. https://github.com/kubernetes/kubernetes/blob/master/staging/src/k8s.io/api is synced to k8s.io/api. All changes must be made in the former. The latter is read-only.