

Sampling New OpenAPI Data

This document describes how to fetch OpenAPI data from a live kubernetes API server. The scripts used will create a clean [kind](#) instance for this purpose.

Replacing the default openapi schema version

Delete all currently built-in schema

This will remove both the Kustomization and Kubernetes schemas:

```
make nuke
```

Choose the new version to use

The compiled-in schema version should maximize API availability with respect to all actively supported Kubernetes versions. For example, while 1.20, 1.21 and 1.22 are the actively supported versions, 1.21 is the best choice. This is because 1.21 introduces at least one new API and does not remove any, while 1.22 removes a large set of long-deprecated APIs that are still supported in 1.20/1.21.

Update the built-in schema to a new version

In the Makefile in this directory, update the `API_VERSION` to your desired version.

You may need to update the version of Kind these scripts use by changing `KIND_VERSION` in the Makefile in this directory. You can find compatibility information in the [kind release notes](#).

In this directory, fetch the openapi schema and generate the corresponding swagger.go for the kubernetes api:

```
make all
```

The above command will update the [OpenAPI schema](#) and the [Kustomization schema](#). It will create a directory `kubernetesapi/v1212` and store the resulting `swagger.json` and `swagger.go` files there.

Precomputations

To avoid expensive schema lookups, some functions have precomputed results based on the schema. Unit tests ensure these are kept in sync with the schema; if these tests fail you will need to follow the suggested diff to update the precomputed results.

Run all tests

At the top of the repository, run the tests.

```
make prow-presubmit-check >& /tmp/k.txt; echo $?
```

The exit code should be zero; if not, examine `/tmp/k.txt`.

Generating additional schemas

Instead of replacing the default version, you can specify a desired version as part of the make invocation:

```
rm kubernetesapi/swagger.go
make kubernetesapi/swagger.go API_VERSION=v1.21.2
```

While the above commands generate the `swagger.go` files, they do not make them available for use nor do they update the `info` field reported by `kustomize openapi info`. To make the newly fetched schema and `swagger.go` available:

```
rm kubernetesapi/openapiinfo.go
make kubernetesapi/openapiinfo.go
```

Partial regeneration

You can also regenerate the kubernetes api schemas specifically with:

```
rm kubernetesapi/swagger.go
make kubernetesapi/swagger.go
```

To fetch the schema without generating the `swagger.go`, you can run:

```
rm kubernetesapi/swagger.json
make kubernetesapi/swagger.json
```

Note that generating the `swagger.go` will re-fetch the schema.