

Home (<https://erwinvaneyk.nl/>) / **Converting Kubernetes unstructured to typed objects**

# Converting Kubernetes unstructured to typed objects

📅 January 2, 2020 · ⌚ 1 minute read

To interact with the Kubernetes API using the [client-go](https://github.com/kubernetes/client-go) (<https://github.com/kubernetes/client-go>) library there are two primary APIs: the typed `kubernetes.Interface` API and the unstructured `dynamic.Interface` API.

Although using the core kubernetes API is (for Kubernetes) [well-documented](https://github.com/kubernetes/client-go/tree/master/examples/create-update-delete-deployment) (<https://github.com/kubernetes/client-go/tree/master/examples/create-update-delete-deployment>), the dynamic API has fewer examples. There are a couple of examples, these did not show how more advanced examples how to work with the unstructured responses.

Because of the lack of documentation or examples, it took me some time to find out the specific package/function to convert unstructured to a typed object—which is why this post aims to document it for others (or at least my future self).

The short answer to converting `unstructured.Unstructured` to a typed resource is to use the `runtime.UnstructuredConverter` interface. Generally, the `runtime.DefaultUnstructuredConverter` implementation suffices for almost all use cases.

A full example which interacts with the Cluster CRD from [Cluster API](https://cluster-api.sigs.k8s.io) (<https://cluster-api.sigs.k8s.io>):

```

package main

import (
    "fmt"

    metav1 "k8s.io/apimachinery/pkg/apis/meta/v1"
    "k8s.io/apimachinery/pkg/runtime"
    "k8s.io/client-go/dynamic"
    "k8s.io/client-go/rest"
    "sigs.k8s.io/cluster-api/api/v1alpha2"
)

func main() {
    // Create a new dynamic client.
    restConfig, err := clientcmd.BuildConfigFromFlags("", "")
    assertNoError(err)
    kubeClient, err := dynamic.NewForConfig(restConfig)
    assertNoError(err)


    // Get a resource (returns an unstructured object).
    resourceScheme := v1alpha2.SchemeBuilder.
        GroupVersion.WithResource("cluster")
    resp, err := kubeClient.Resource(resourceScheme).
        Namespace(namespace).
        Get(name, metav1.GetOptions{})
    assertNoError(err)

    // Convert the unstructured object to cluster.
    unstructured := resp.UnstructuredContent()
    var cluster v1alpha2.Cluster
    err = runtime.DefaultUnstructuredConverter.
        FromUnstructured(unstructured, &cluster)
    assertNoError(err)

    // Use the typed object.
    fmt.Println(cluster.Status.Phase)
}

func assertNoError(err error) {
    if err != nil {
        panic(err)
    }
}

```


 **Tags:**

Golang

Kubernetes

short

tooling

 **Updated:** January 2, 2020