**Go Pointers**

**type API struct {**

**mux \*http.ServeMux**

**muxInitialized bool**

**}**

**// binding struct input function parameters**

**func (api \*API) AddResource(resource interface{}, paths ...string) {**

**for \_, path := range paths {**

**api.Mux().HandleFunc(path, api.requestHandler(resource))**

**}**

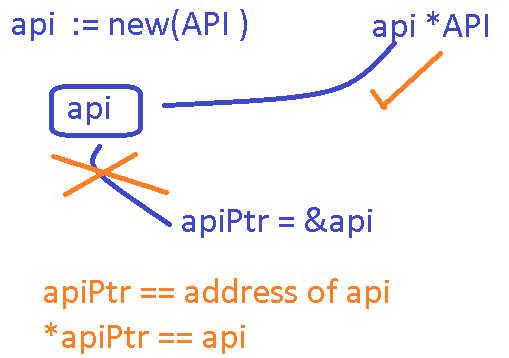
**}**

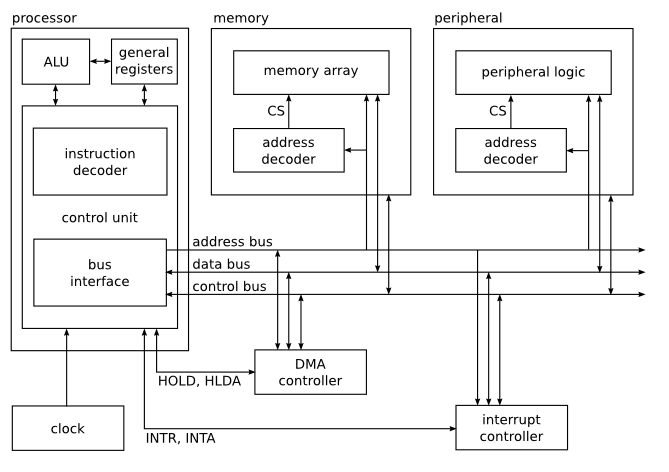
**item:= new(Item)**

**api := new(API )**

**api.AddResource(item, "/items")**

Pointers allow you to refer to the same space in memory from multiple locations. This means that you can update memory in one location and the change can be seen from another location in your program. You will also save space by being able to share components in your data structures.





**Pass-by-reference**

**type API struct {**

**a int**

**b int**

**}**

**func (api MyAPI ) passByReference() {**

**api.a = 5**

**api.b = 7**

**}**

**var api= API{0, 0}**

**api.passByReference()**

**fmt.Println(api)**

**{0 0}**

**Pass-by-value.**

**type API struct {**

**a int**

**b int**

**}**

**func (api \*API ) passByValue() {**

**api.a = 5**

**api.b = 7**

**}**

**var api= API{0, 0}**

**api.passByValue()**

**fmt.Println(api)**

**{5 7}**