AKSO rest API

1, Endpoint

The syntax of URI of a endpoint will be "/food/<API version>/<Method>". Such as   
"/food/v1/material"  
“/food/v1/material/<materialName>”

2, Request

Client submit request -> Router -> manage API

3， http listener

http.ListenAndServe(rest.DefaultStoreEndPoint, routers)

4, Router

route{

"CreateMaterial",

"POST",

FoodPath + "material",

manager.createMaterial,

},

route{

"GetMaterial",

"GET",

FoodPath + "material/{materialName}",

manager.getMaterial,

},

Codes in manager

type foodManager struct {

foodStore store.FoodStore

}

func (mgr \*foodManager) createMaterial(w http.ResponseWriter, r \*http.Request) {

var material meta.Material

err := decodeRequestBody(r, &material)

if err != nil {

sendBadRequestResponse(w, err.Error())

return

}

if \_, err := mgr.foodStore.GetMaterial(material.Name); err == nil {

// material already exists.

sendBadRequestResponse(w, "material already exists.")

return

}

if err := mgr.foodStore.CreateMaterial(&material); err != nil {

sendBadRequestResponse(w, err.Error())

return

}

sendCreatedResponse(w, &material)

}

5, manage API

1. Struct :

type Material struct {

Name string `json:"name"`

Type string `json:"type"`

Description string `json:"description,omitempty"`

Tags string `json:"tags,omitempty"`

}

type MaterialStore struct {

Materials []\*Material `json:"materials,omitempty"`

}

1. Interface:

type storeManager interface {

ReadStore() (\*meta.MaterialStore, error)

WriteStore(\*meta.MaterialStore) error

}

type materialManager interface {

GetMaterial(materialName string) (\*meta.Material, error)

CreateMaterial(meterial \*meta.Material) error

}

type FoodStore interface {

storeManager

materialManager

}