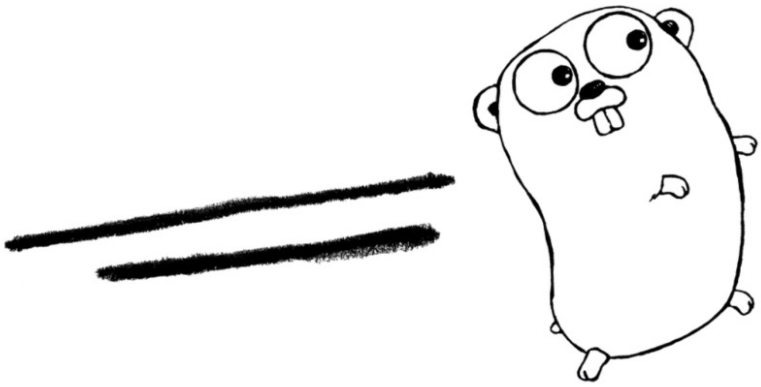
# Go Restful With MongoDB

linkedin.com/pulse/go-restful-mongodb-moshe-beeri



I am just starting to learn the wonderful Go language, somehow I decided my Hello World! program will be implementing a restful micro-service that is delegating to MongoDB, while implementing I was thinking that the idea of using MongoDB is a bit not Google way, I more of expected go to naturally connect with Firestore running as cloud function, but learning Go Restful and MongoDB in one struck seams large enough bit to take, so I left it for the next time.

The result is a this "one pager", you may also find it on my GitHub https://github.com/moshebeeri/gorest

```
package main
import (
  "context"
  "encoding/json"
  "fmt"
  "io/ioutil"
  "log"
  "net/http"


  "go.mongodb.org/mongo-driver/bson"
  "go.mongodb.org/mongo-driver/bson/primitive"
  "go.mongodb.org/mongo-driver/mongo"
  "go.mongodb.org/mongo-driver/mongo/options"
)


// Feature information
type Feature struct {
  Name        string `json:"name"`
  Description string `json:"description"`
```

```
}


var collection *mongo.Collection


func httpServerFunc(w http.ResponseWriter, r *http.Request) {
  if r.URL.Path != "/" {
    http.Error(w, "404 not found.", http.StatusNotFound); return
  }
  switch r.Method {
  case "GET":
    id := r.URL.Query().Get("id")
    fmt.Println("collection.FindOne: ", id)


    var feature Feature
    objectID, _ := primitive.ObjectIDFromHex(id)
    filter := bson.M{"_id": objectID}
    err := collection.FindOne(context.TODO(), filter).Decode(&feature)
    if err != nil {
      log.Fatal(err)
    }
    fmt.Printf("Found a single document: %+v\n", feature)
    json.NewEncoder(w).Encode(feature)


  case "POST":
    reqBody, err := ioutil.ReadAll(r.Body)
    if err != nil {
      log.Fatal(err)
    }
    var feature Feature
    if err := json.Unmarshal([]byte(reqBody), &feature); err != nil {
      panic(err)
    }
    insertResult, err := collection.InsertOne(context.TODO(), feature)
    if err != nil {
      log.Fatal(err)
      fmt.Fprintf(w, "ParseForm() err: %v", err)
      return
    }
    fmt.Println("Inserted a single document: ", insertResult.InsertedID)
  default:
    fmt.Fprintf(w, "Sorry, only GET and POST methods are supported.")
  }
}


func main() {
  // Set client options
  clientOptions := options.Client().ApplyURI("mongodb://localhost:27017/go")
  // Connect to MongoDB
  client, err := mongo.Connect(context.TODO(), clientOptions)
```

```
  if err != nil {
    log.Fatal(err)
  }
  // Check the connection
  err = client.Ping(context.TODO(), nil)
  if err != nil {
    log.Fatal(err)
  }
  fmt.Println("Connected to MongoDB!")
  collection = client.Database("go").Collection("features")
  http.HandleFunc("/", httpServerFunc)
  err = http.ListenAndServe("localhost:8765", nil)
  if err != nil {
    log.Fatal("ListenAndServe", err)
  }
}
```