package main

```
"encoding/json"
"github.com/aws/aws-sdk-go/aws"
              'github.com/aws/aws-lambda-go/lambda"
'github.com/aws/aws-lambda-go/events"
              "github.com/aws/aws-sdk-go/service/dynamodb"
"github.com/aws/aws-sdk-go/service/dynamodb/dynamodbattribute"
                 ERROR_MISSING_ID_FIELD = 1
ERROR_INTERNAL_SERVERS_DATABAE = 2
ERROR_NO_ITEM_FOUNDED = 3
type SuccessResponse struct{
    Device types.Device
                                                                     `ison:"data"
type dynamoDBAPI struct{
DynamoDB dynamodbiface.DynamoDBAPI
var databseStruct *types.DatabseStruct
var dynamodbapi *dynamoDBAPI
func init(){
    databseStruct = new(types.DatabseStruct)
    region := os.Getenv("AWS_REGION")
    dynamodbapi = new(dynamoDBAPI) // crate a setter that can be used for inserting
    sess, err := session.NewSession(&aws.Config{Region: &region},)
    databseStruct SessionFrom = orr
             databseStruct.SessionError = err
            databasette.Sess.Delroi - eri
svc := dynamodb.New(sess)
dynamodbapi.DynamoDB = dynamodbiface.DynamoDBAPI(svc)
             if err != nil {
             // Get table name from OS's environment
fetchedTableName :=os.Getenv("DEVICES TABLE NAME")
                  len(fetchedTableName) == 0 {
    databseStruct.TableName = nil;
    fmt.Println("It is not possible to fetch device tabel name")
             var input = &dynamodb.GetItemInput{
                          TableName: databseStruct.TableName,
Key: map[string]*dynamodb.AttributeValue{
                                         "id": {
                                                     S: aws.String(id),
             result, err := ig.DynamoDB.GetItem(input)
return result, err
 / main AWS lambda function starting point.
/ It gets an id from client, parse it and tries to get corresponding device fromdynamodb.
unc GetDeviceById(request events.APIGatewayProxyRequest) (events.APIGatewayProxyResponse, error) {
             // there is some internal server error
if databseStruct.SessionError != nil || databseStruct.TableName == nil {
             // get requested id from APIGatewayProxyRequest
id := request.PathParameters["id"]
             // If no id provided, return HTTP error 404
if id == "" {
                         result, err := dynamodbapi.getFromDatabase(id)
validationResult := validateDatabaseResult(result, err)
return validationResult , nil
func validateDatabaseResult(result *dynamodb.GetItemOutput, err error)( events.APIGatewayProxyResponse) {
    // If an internal error occured in the database, return HTTP error 500
    // todo: log here
    if err != nil {
                          StatusCode: 500,
             // If no item founded, return error 404 if len(result.Item) == 0 {
```