IST 412

Engineering of Complex Software Systems Fall 2018 Joe Oakes

AWS ERP Team Vinyl

Unit Testing - AWS Lambda

Chan Morn

Ricky Zhao

Michael Morrell

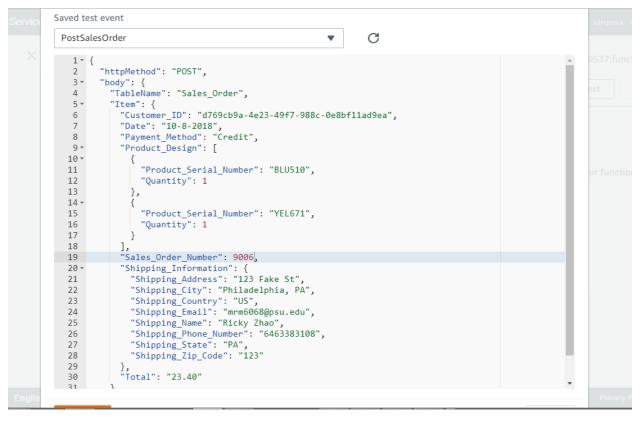
Zachary Sigamony

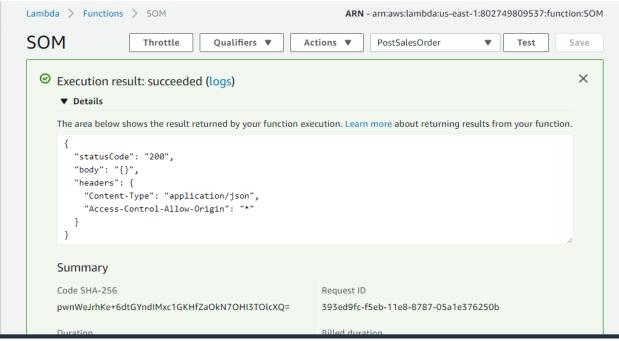
James O'Connell

Johnathan Seitz

Sales Order Module (SOM)

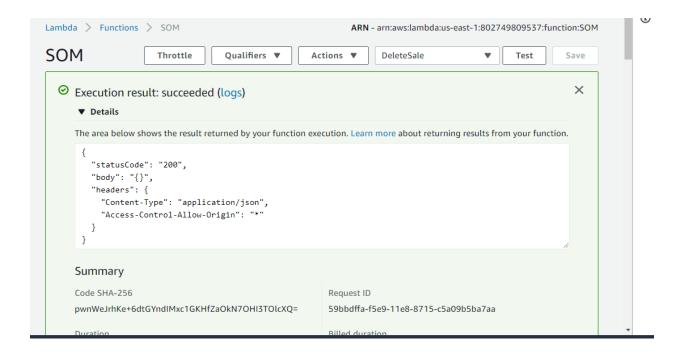
Post Sale





Delete Sale

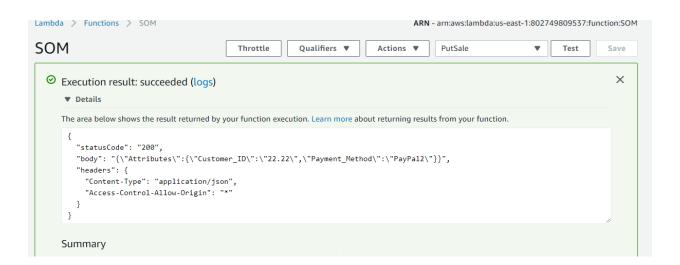
```
DeleteSale
  1 - {
  2
       "httpMethod": "DELETE",
       "queryStringParameters": {
  3 ₹
         "TableName": "Sales_Order"
  4
  5
  6 +
       "body": {
  7
         "TableName": "Sales_Order",
         "Key": {
  8 +
  9
         "Sales_Order_Number": 801
 10
 11
 12
```



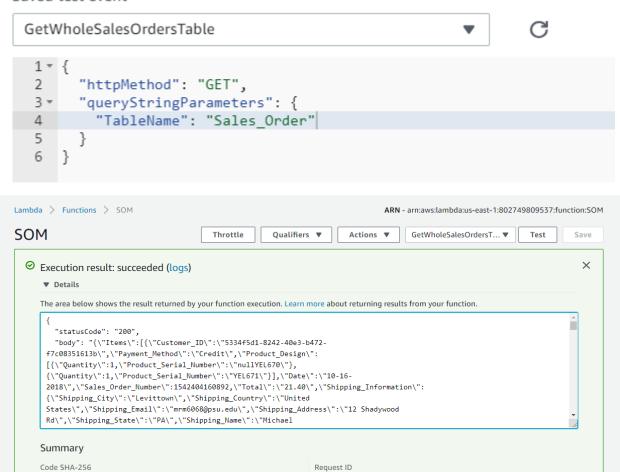
Update Sale

Edit saved test events

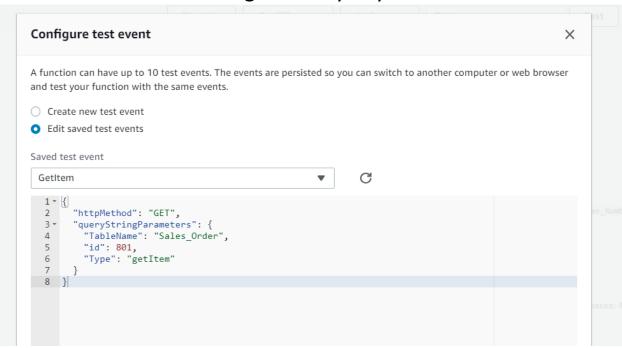
```
C
PutSale
  1 * {
       "httpMethod": "PUT",
  2
  3 ₹
       "queryStringParameters": {
        "TableName": "Sales_Order"
  4
       },
  5
  6 +
       "body": {
  7
        "TableName": "Sales_Order",
         "Key": {
 8 +
        "Sales_Order_Number": 801
 9
 10
 11
         "UpdateExpression": "set Payment_Method = :a, Customer_ID = :b",
        "ExpressionAttributeValues": {
 12 -
        ":a": "PayPal",
 13
         ":b": "23.40"
 14
 15
         "ReturnValues": "UPDATED_NEW"
 16
 17
       }
 18 }
```

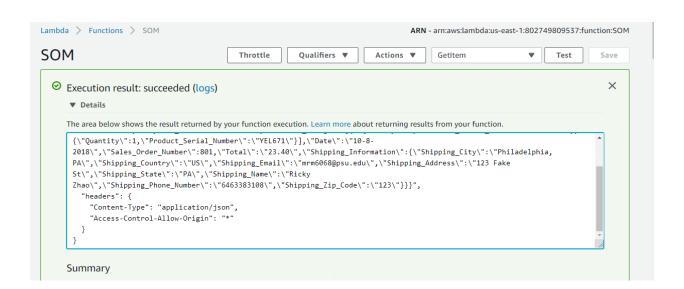


Get All Sales

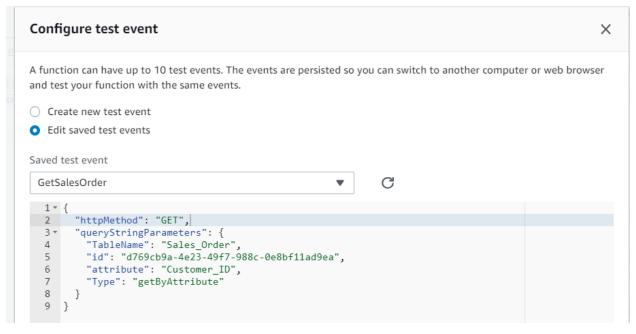


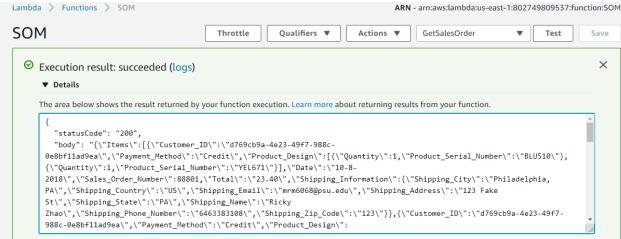
Get Single Sale by Key Value





Get Sale(s) by Any Attribute

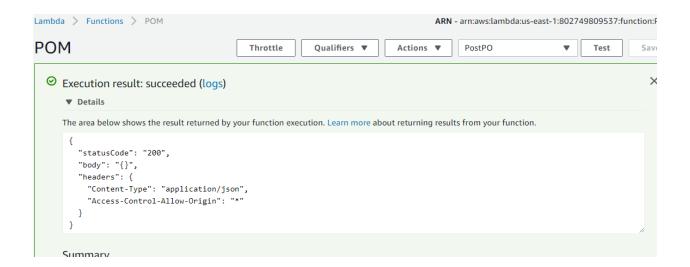




Purchase Order Module (POM)

Post PO

```
PostPO
  1 * {
  2
       "httpMethod": "POST",
  3 ₹
       "body": {
         "TableName": "Purchase_Order",
  4
  5 +
          "Item": {
  6
           "Arrival_Date": "6-3-1",
  7
           "Date_Issued": "1-1-1",
  8
           "Material ID": "1",
  9
           "Order_Placement_Date": "1-1-1",
           "Payment_Method": "1",
 10
           "Purchase_Order_Number": 4,
 11
           "Purchase_Order_Total": "$1",
 12
           "Supplier_ID": "4",
 13
           "Warehouse ID": "2"
 14
 15
 16
     }
 17
```



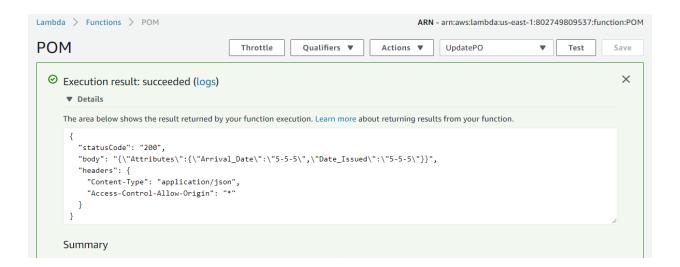
Delete PO

```
C
DeletePO
 1 * {
 2
       "httpMethod": "DELETE",
 3 *
      "queryStringParameters": {
        "TableName": "Purchase_Order"
 4
      },
"body": {
  5
 6 +
 7
        "TableName": "Purchase_Order",
       "Key": {
 8 +
 9
          "Purchase_Order_Number": 4
 10
 11
```



Update PO

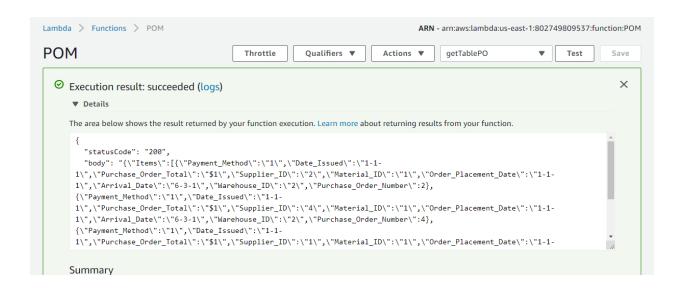
A function can have up to 10 test events. The events are persisted so you can switch to another computer or web brows and test your function with the same events. Create new test event Edit saved test events Saved test event UpdatePO C 1 * [{ 2 "httpMethod": "PUT", 3 = "queryStringParameters": { 4 "TableName": "Purchase_Order" 5 "body": { 6 * 7 "TableName": "Purchase_Order", "Key": { 8 + 9 "Purchase_Order_Number": 1 10 11 "UpdateExpression": "set Arrival_Date = :a, Date_Issued = :b", "ExpressionAttributeValues": { 12 -":a": "5-5-5", ":b": "5-5-5" 13 14 15 }, "ReturnValues": "UPDATED_NEW" 16 17 18 }



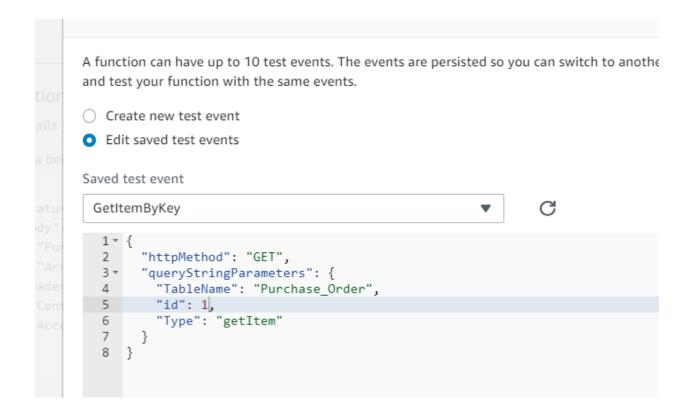
Get All POs

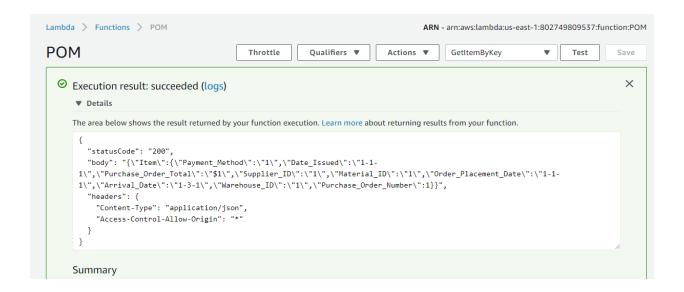
```
getTablePO ▼ C

1 * {
2    "httpMethod": "GET",
3 * "queryStringParameters": {
4    "TableName": "Purchase_Order"
5    }
6 }
```



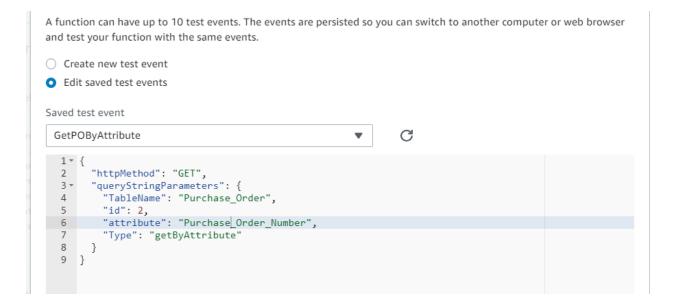
Get Single PO by Key





Get PO(s) by any Attribute

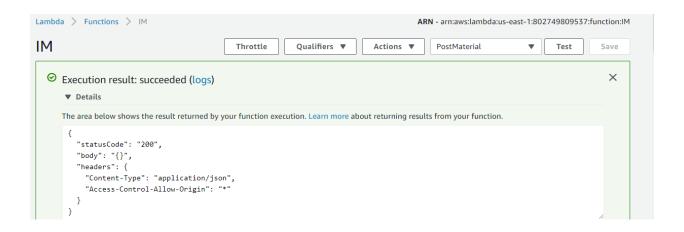
A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events. Oreate new test event Edit saved test events Saved test event GetPOByAttribute C "httpMethod": "GET", 2 3 ₹ "queryStringParameters": { "TableName": "Purchase_Order",
"id": 2, 4 6 "attribute": "Purchase_Order_Number", 7 8 } 9 } "Type": "getByAttribute"



Inventory Module (IM)

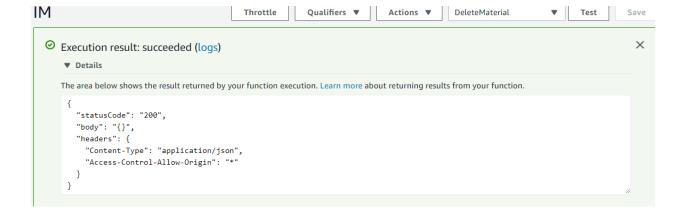
Post Material

```
PostMaterial
  1 * {
        "httpMethod": "POST",
  2
  3 ₹
       "body": {
  4
         "TableName": "Material",
  5 +
          "Item": {
  6
              "Material_Components": "5555r",
              "Material_Count": "555",
  7
              "Material_Description": "5555",
  8
              "Material_ID": "5555",
  9
              "Material_Level": "Medium",
 10
              "Product_Serial_Number": "CTX714",
 11
              "Warehouse_ID": "5555"
 12
 13
 14
 15
```



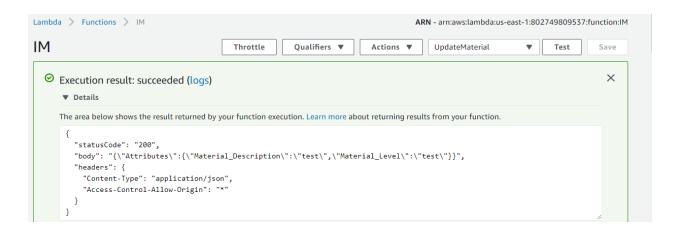
Delete Material

```
C
DeleteMaterial
  1 - {
        "httpMethod": "DELETE",
"queryStringParameters": {
  2
  3 ₹
          "TableName": "Material"
  4
  5
       "body": {
  6 +
  7
         "TableName": "Material",
  8 +
          "Key": {
        "Material_ID": "5555"
}
  9
 10
       }
 11
```



Update Material

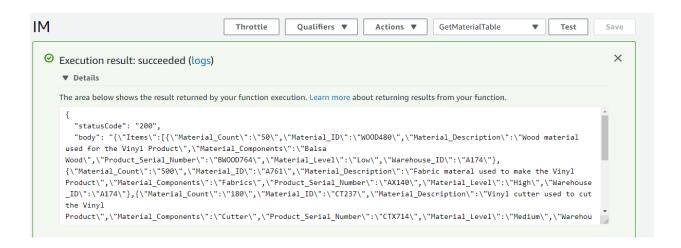
```
UpdateMaterial
                                                              C
  1 * {
       "httpMethod": "PUT",
  2
        "queryStringParameters": {
  3 +
  4
         "TableName": "Material"
  5
       "body": {
  6 +
  7
         "TableName": "Material",
  8 +
         "Key": {
           "Material_ID": "5555"
  9
 10
 11
         "UpdateExpression": "set Material Description = :a, Material Level = :b",
         "ExpressionAttributeValues": {
 12 -
         ":a": "test",
":b": "test"
 13
 14
 15
         },
"ReturnValues": "UPDATED_NEW"
 16
 17
 18 }
```



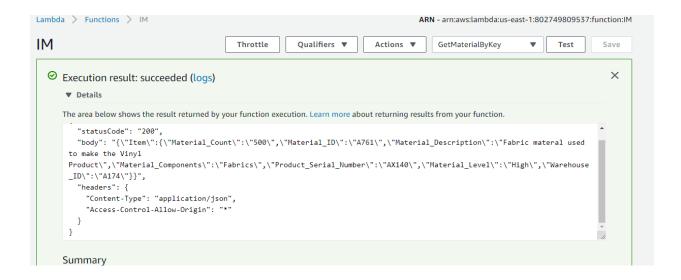
Get Material Table

```
GetMaterialTable

1 ▼ {
2    "httpMethod": "GET",
3 ▼    "queryStringParameters": {
4     "TableName": "Material"|
5    }
6 }
```



Get Single Material by Key



Get Material(s) by any Attribute

```
GetMaterialByAttribute

1 → {
2     "httpMethod": "GET",
3 → "queryStringParameters": {
4     "TableName": "Material",
5     "id": "test2",
6     "attribute": "Material_Level",
7     "Type": "getByAttribute"
8     }
0     N
```

```
Throttle Qualifiers ▼ Actions ▼ GetMaterialByAttribute ▼ Test Save

The area below shows the result returned by your function execution. Learn more about returning results from your function.

{
    "statusCode": "200",
    "body": "{\"Items\":
    [{\"Material_Description\":\"test2\",\"Material_ID\":\"5555\",\"Material_Level\":\"test2\"}],\"Count\":1,\"ScannedCount\":6}",
    "headers": {
    "Content-Type": "application/json",
    "Access-Control-Allow-Origin": "*"
    }

Summary
```