

cosmos-trigger-cosmodb-output-binding (Python)

Sample	Description	Trigger	In Bindings	Out Bindings
cosmos-trigger-cosmodb-output-binding	Azure Functions Cosmos DB Trigger Python Sample. The function gets document data from Azure Cosmos DB Trigger, ROT13 encode obtained clear text, and store encoded data into Azure Cosmos DB by using Cosmos DB output binding	CosmosDB	NONE	CosmosDB

Configurations

As specified in `functions.json`, you need Azure Cosmos DB account for triggering functions and storing data using Cosmos DB output binding

```
{
  "scriptFile": "__init__.py",
  "bindings": [
    {
      "type": "cosmosDBTrigger",
      "name": "docs",
      "direction": "in",
      "leaseCollectionName": "leases",
      "connectionStringSetting": "MyCosmosDBConnectionString",
      "databaseName": "testdb",
      "collectionName": "testcol01",
      "createLeaseCollectionIfNotExists": true
    },
    {
      "direction": "out",
      "type": "cosmosDB",
      "name": "outdoc",
      "databaseName": "testdb",
      "collectionName": "testcol02",
      "leaseCollectionName": "leases",
      "createLeaseCollectionIfNotExists": true,
      "connectionStringSetting": "MyCosmosDBConnectionString",
      "createIfNotExists": true
    }
  ]
}
```

Create Cosmos DB Account and DB & Collection for testing

Create a Cosmos DB Account

```
COSMOSDB_ACCOUNT_NAME="azfuncv2db"  
RESOURCE_GROUP="RG-azfuncv2"  
az cosmosdb create \  
  --name $COSMOSDB_ACCOUNT_NAME \  
  --kind GlobalDocumentDB \  
  --resource-group $RESOURCE_GROUP
```

Create Database and Collection in the Cosmos DB that you've created

```
# Get Key
COSMOSDB_KEY=$(az cosmosdb list-keys --name $COSMOSDB_ACCOUNT_NAME --resource-group $RES

# Create Database
az cosmosdb database create \
  --name $COSMOSDB_ACCOUNT_NAME \
  --db-name $DATABASE_NAME \
  --key $COSMOSDB_KEY \
  --resource-group $RESOURCE_GROUP

# Create a container with a partition key and provision 400 RU/s throughput.
COLLECTION_NAME="testcol01"
az cosmosdb collection create \
  --resource-group $RESOURCE_GROUP \
  --collection-name $COLLECTION_NAME \
  --name $COSMOSDB_ACCOUNT_NAME \
  --db-name $DATABASE_NAME \
  --partition-key-path /name \
  --throughput 400

COLLECTION_NAME="testcol02"
az cosmosdb collection create \
  --resource-group $RESOURCE_GROUP \
  --collection-name $COLLECTION_NAME \
  --name $COSMOSDB_ACCOUNT_NAME \
  --db-name $DATABASE_NAME \
  --partition-key-path /name \
  --throughput 400

# Create a container for leaves
# 'leaves' need to be a single collection partition
# Please see also: https://github.com/Azure/azure-functions-core-tools/issues/930
LEASES_COLLECTION_NAME="leases"
az cosmosdb collection create \
  --resource-group $RESOURCE_GROUP \
  --collection-name $LEASES_COLLECTION_NAME \
  --name $COSMOSDB_ACCOUNT_NAME \
  --db-name $DATABASE_NAME \
  --throughput 400
```

How to develop and publish the function

Local development

```
func host start
```

Publish the function to the cloud

Publish the function to the cloud

```
FUNCTION_APP_NAME="MyFunctionApp"
func azure functionapp publish $FUNCTION_APP_NAME --build-native-deps --no-bundler
```

Add Functions App Settings

```
COSMOS_DB_CONNECTION="*****"
az webapp config appsettings set \
  -n $FUNCTION_APP_NAME \
  -g $RESOURCE_GROUP \
  --settings \
    MyCosmosDBConnectionString=$COSMOS_DB_CONNECTION
```