Serverless | Cosmos DB

Running and publish Azure Functions project with Cosmos DB

avanade

Prerequisites and more!

An *Azure subscription* (free account).

Main supported languages:

Language	Extension
C#	C# for VS Code
Java	Debugger for Java * + Maven 3+
Typescript	.Net Core 2.2 (preview)
Javascript	Node 8.0+

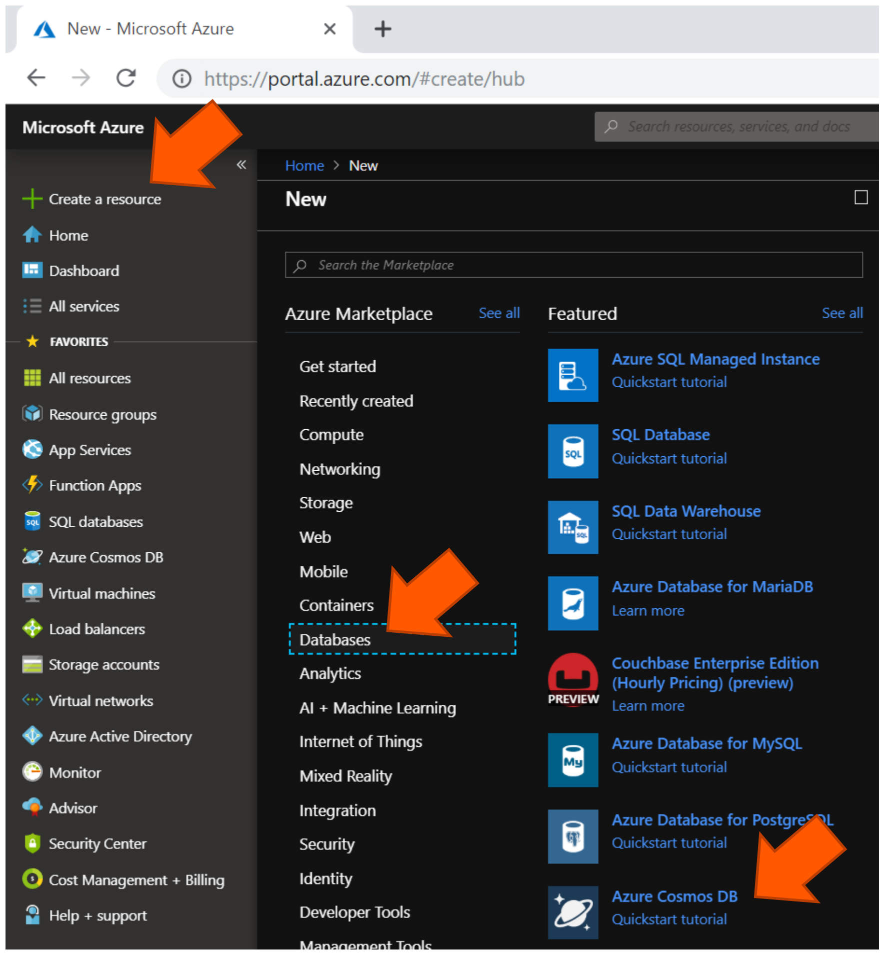
Create an Azure Cosmos DB

You must have an *Azure Cosmos DB* account that uses the *SQL API* before you create the output binding.

Sign in to the *Azure portal*.

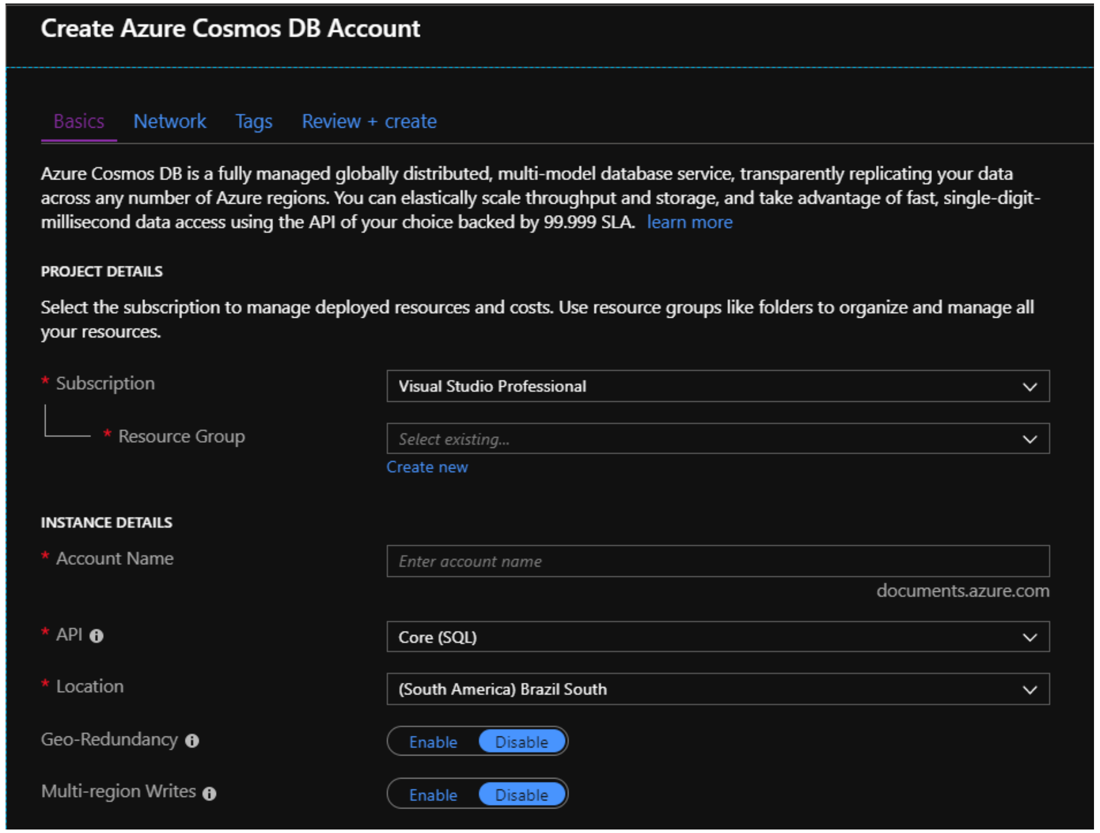
Select:

> *Create a resource*
>
> > *Databases*
> >
> > > *Azure Cosmos DB*



Create an Azure Cosmos DB

On the create *Azure Cosmos DB* page, enter the basic settings for the new *Azure Cosmos account*.



Create Azure Cosmos DB Account

Azure Cosmos DB is a fully managed globally distributed, multi-model database service, transparently replicating your data across any number of Azure regions. You can elastically scale throughput and storage, and take advantage of fast, single-digit-millisecond data access using the API of your choice backed by 99.999 SLA. learn more

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription: Visual Studio Professional
* Resource Group: Select existing...

Create new

INSTANCE DETAILS

* Account Name: Enter account name

documents.azure.com

* API: Core (SQL)
* Location: (South America) Brazil South

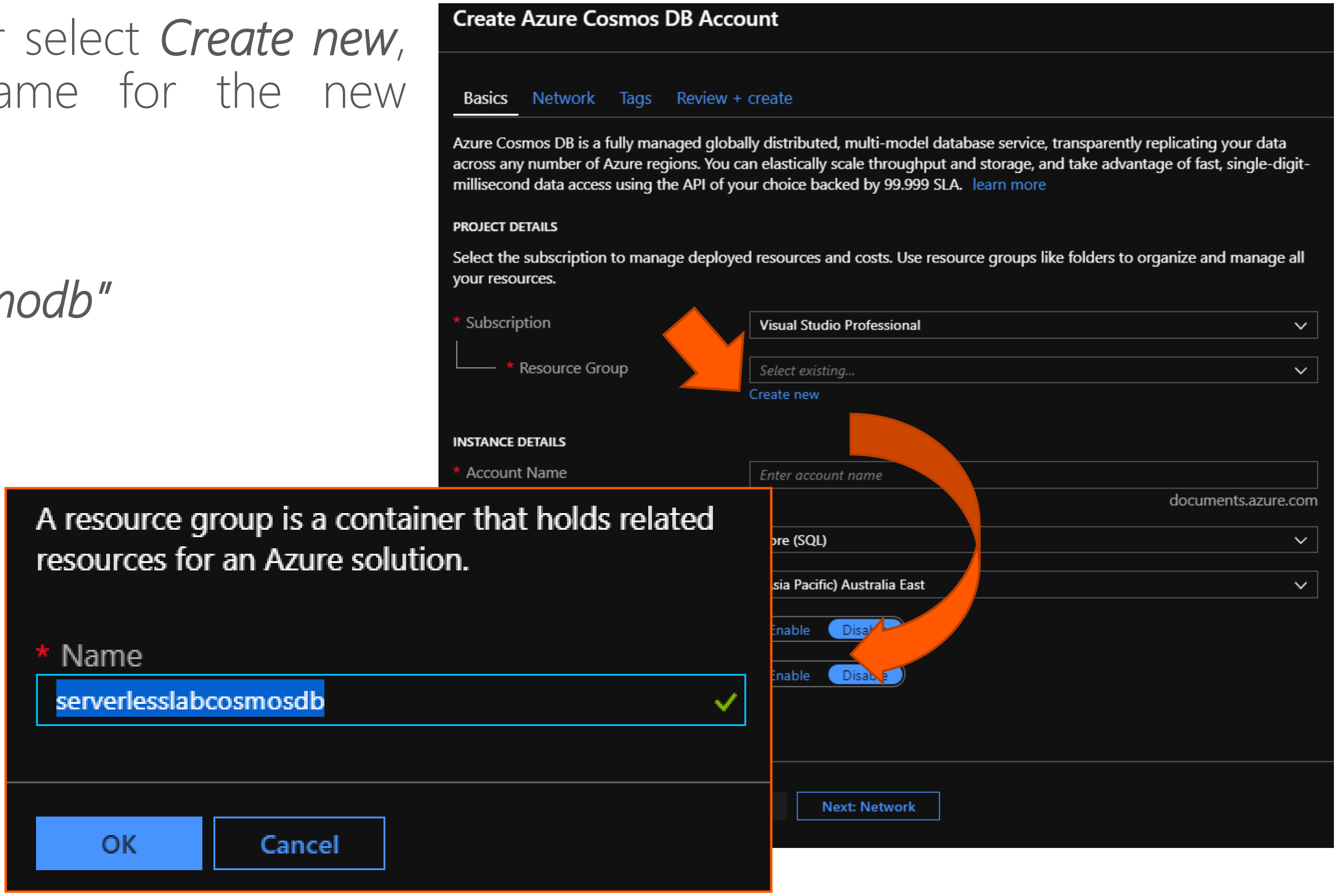
Geo-Redundancy: Enable [✓ Disable]

Multi-region Writes: Enable [✓ Disable]

Create an Azure Cosmos DB

Select a *resource group*, or select *Create new*, then enter a unique name for the new resource group.

Fill using: *"serverlesslabcosmodb"*



Create Azure Cosmos DB Account

Basics | Network | Tags | Review + create

Azure Cosmos DB is a fully managed globally distributed, multi-model database service, transparently replicating your data across any number of Azure regions. You can elastically scale throughput and storage, and take advantage of fast, single-digit-millisecond data access using the API of your choice backed by 99.999 SLA. learn more

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription	Visual Studio Professional
* Resource Group	Select existing...
	Create new

INSTANCE DETAILS

* Account Name	Enter account name
	documents.azure.com
	ore (SQL)
	sia Pacific) Australia East
	Enable Disable
	Enable Disable

Next: Network

A resource group is a container that holds related resources for an Azure solution.

* Name

serverlesslabcosmosdb ✓

OK Cancel

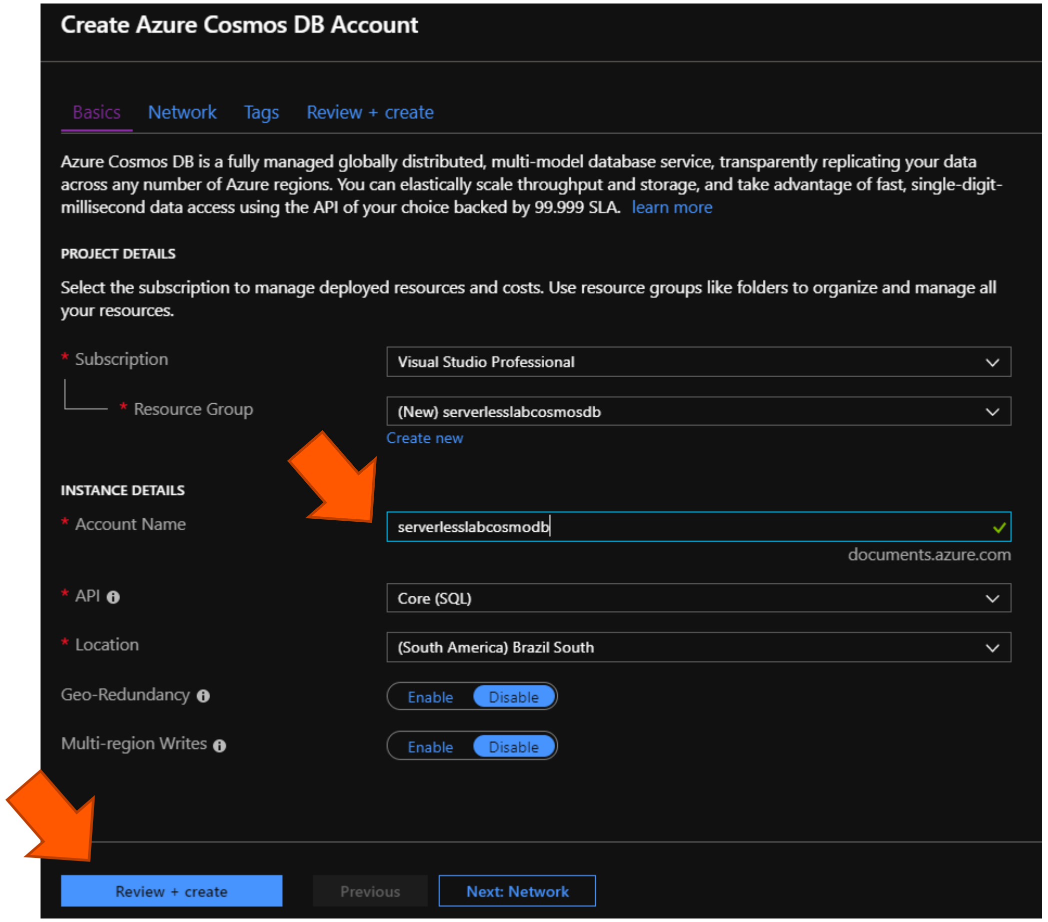
Create an Azure Cosmos DB

Enter a ***name*** to identify your ***Azure Cosmos*** account. (*Because documents.azure.com is appended to the ID that you provide to create your URI, use a unique ID.*)

Fill using: *"serverlesslabcosmodb"*

Select *Review + create*.

You can skip the *Network* and *Tags* sections.



Create Azure Cosmos DB Account

Basics | Network | Tags | Review + create

Azure Cosmos DB is a fully managed globally distributed, multi-model database service, transparently replicating your data across any number of Azure regions. You can elastically scale throughput and storage, and take advantage of fast, single-digit-millisecond data access using the API of your choice backed by 99.999 SLA. learn more

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

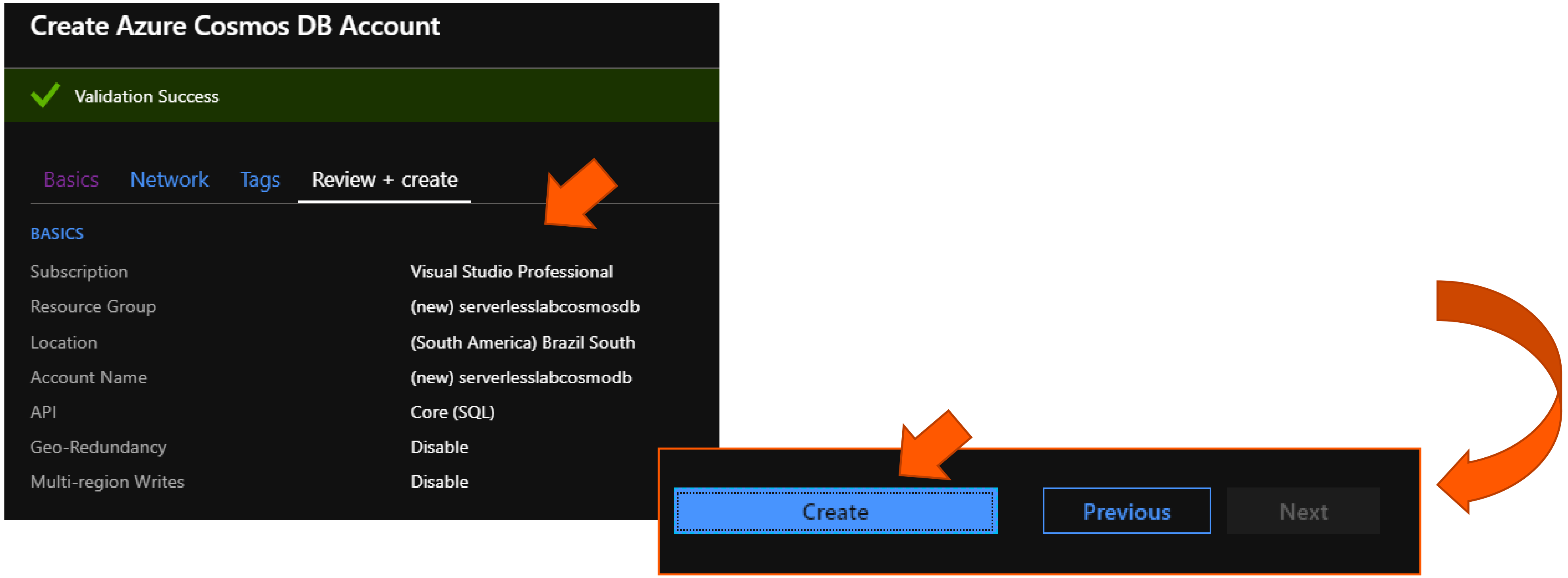
* Subscription	Visual Studio Professional
* Resource Group	(New) serverlesslabcosmosdb
	Create new

INSTANCE DETAILS

* Account Name	serverlesslabcosmodb ✓
	documents.azure.com
* API	Core (SQL)
* Location	(South America) Brazil South
Geo-Redundancy	Enable [✓ Disable]
Multi-region Writes	Enable [✓ Disable]

Create an Azure Cosmos DB

Review the account settings, and then click ***Create*** button. It takes a few minutes to create the account.



Create Azure Cosmos DB Account

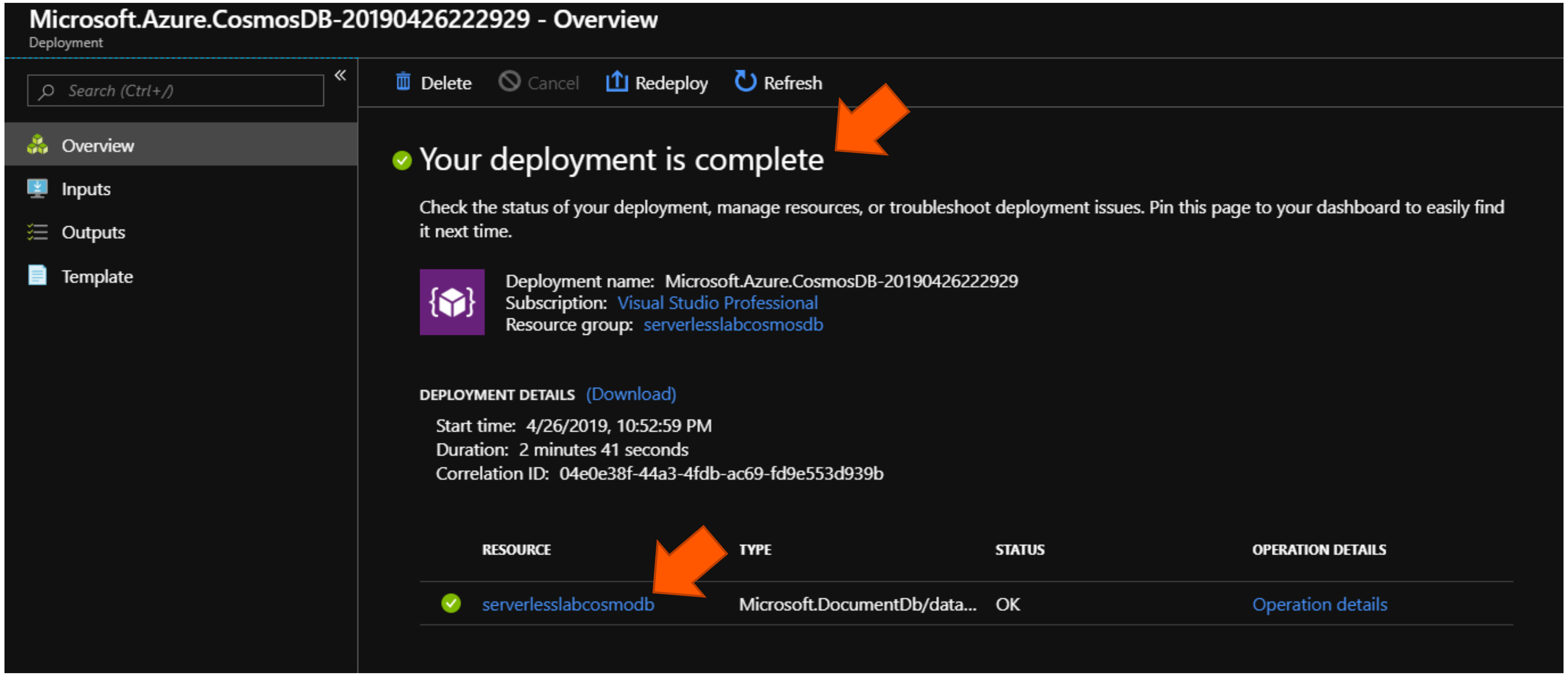
Validation Success

BASICS

Subscription	Visual Studio Professional
Resource Group	(new) serverlesslabcosmosdb
Location	(South America) Brazil South
Account Name	(new) serverlesslabcosmodb
API	Core (SQL)
Geo-Redundancy	Disable
Multi-region Writes	Disable

Create an Azure Cosmos DB

Wait for the portal page to display *your deployment is complete*. Check the resource's name.



Microsoft.Azure.CosmosDB-20190426222929 - Overview
Deployment

Your deployment is complete

Check the status of your deployment, manage resources, or troubleshoot deployment issues. Pin this page to your dashboard to easily find it next time.

Deployment name: Microsoft.Azure.CosmosDB-20190426222929
Subscription: Visual Studio Professional
Resource group: serverlesslabcosmosdb

DEPLOYMENT DETAILS (Download)

Start time: 4/26/2019, 10:52:59 PM
Duration: 2 minutes 41 seconds
Correlation ID: 04e0e38f-44a3-4fdb-ac69-fd9e553d939b

RESOURCE	TYPE	STATUS	OPERATION DETAILS
✓ serverlesslabcosmodb	Microsoft.DocumentDb/data...	OK	Operation details

avanade

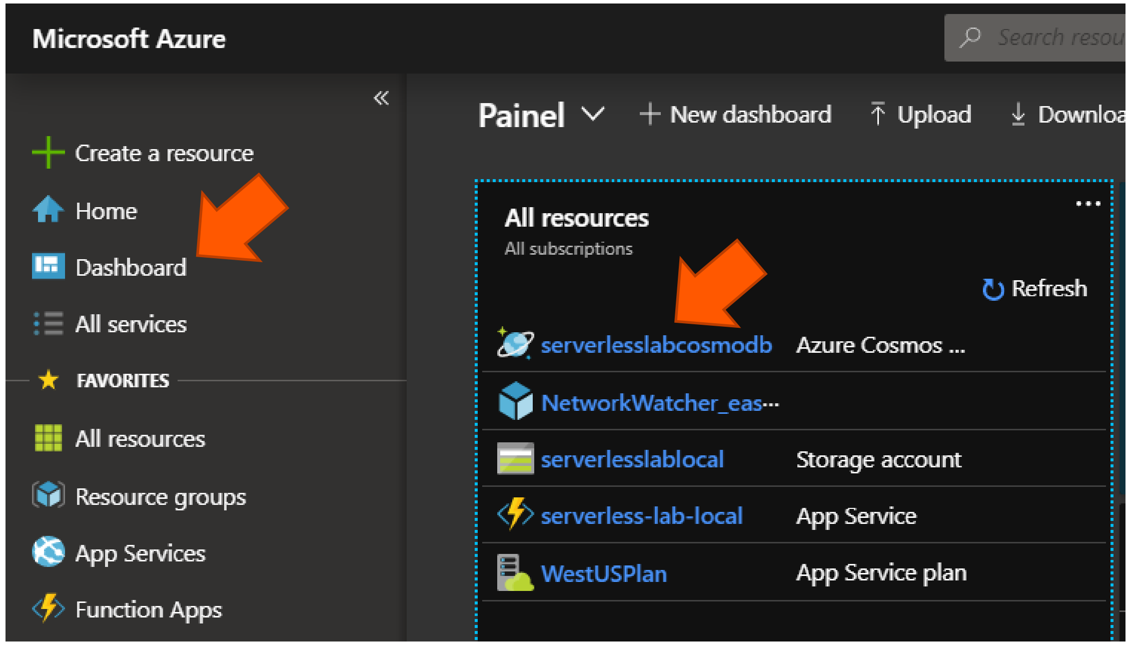
Access an Azure Cosmos DB resource

You must have to access *Dashboard* and find your *Azure Cosmos DB* resource.

Select:

> *Dashboard*

> > *serverlesslabcosmodb*

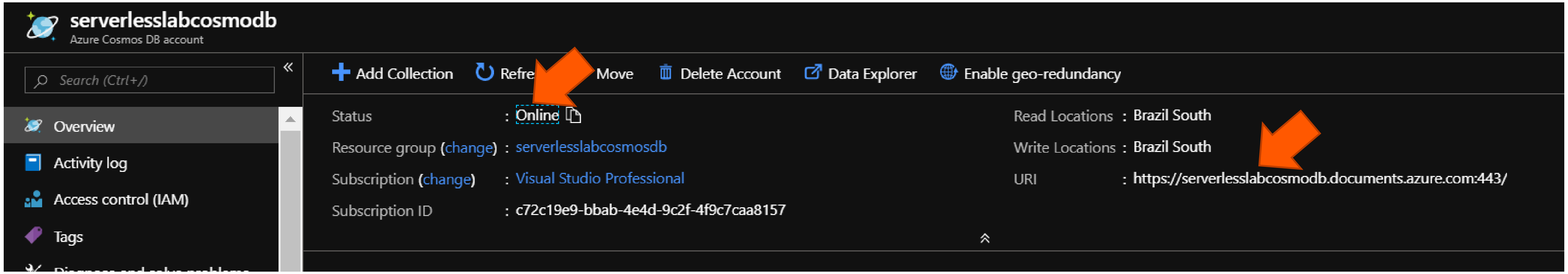


All resources
All subscriptions

serverlesslabcosmodb	Azure Cosmos ...
NetworkWatcher_eas...	
serverlesslablocal	Storage account
serverless-lab-local	App Service
WestUSPlan	App Service plan

Access an Azure Cosmos DB resource

Some important informations to have external access (*application, AFs and etc.*) to your *Azure Cosmos DB*.



serverlesslabcosmodb
Azure Cosmos DB account

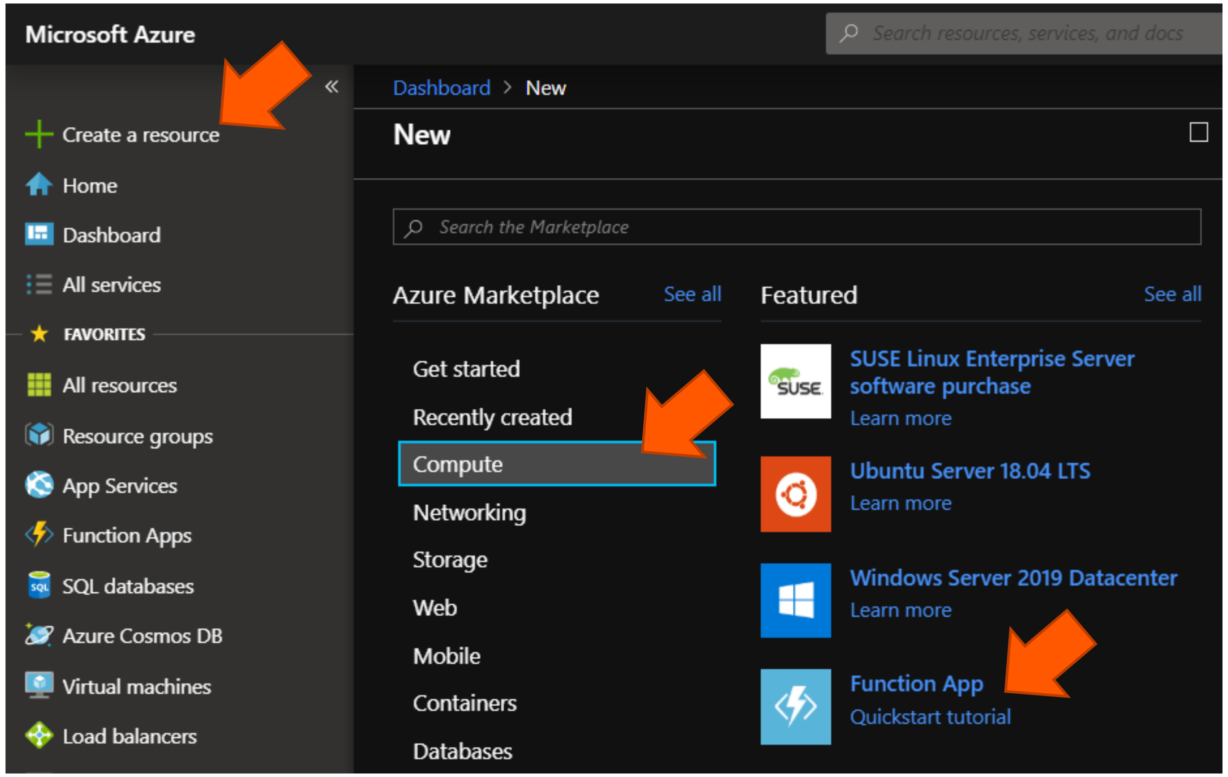
Status	: Online	Read Locations	: Brazil South
Resource group (change)	: serverlesslabcosmosdb	Write Locations	: Brazil South
Subscription (change)	: Visual Studio Professional	URI	: https://serverlesslabcosmodb.documents.azure.com:443/
Subscription ID	: c72c19e9-bbab-4e4d-9c2f-4f9c7caa8157		

Create an Azure Function | App

Select the *Create* a resource button found on the upper left-hand corner of the *Azure portal*.

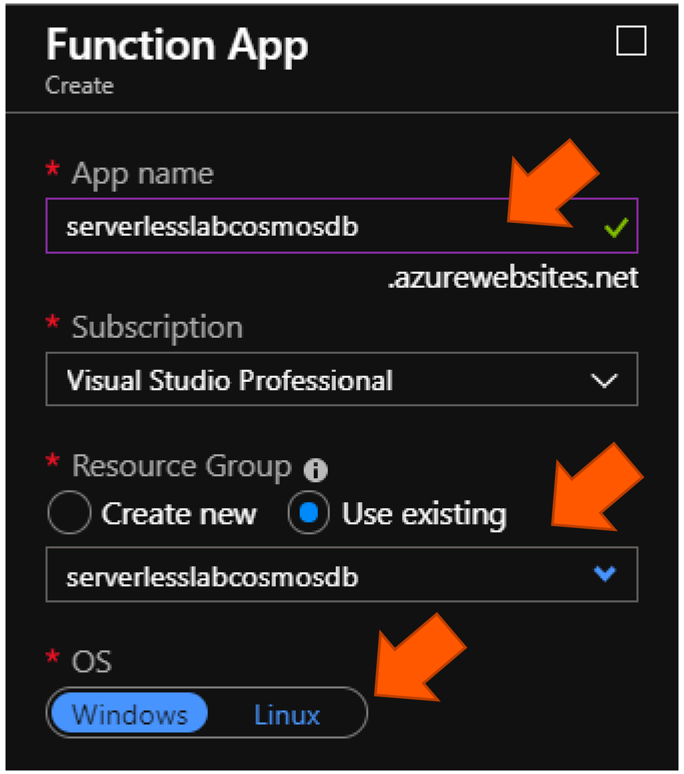
Select:

> *Create a resource*
>
> > *Compute*
> >
> > > *Function App*



Create an Azure Function | App

Fill and review the *Azure Function* settings, and then click *Create* button.



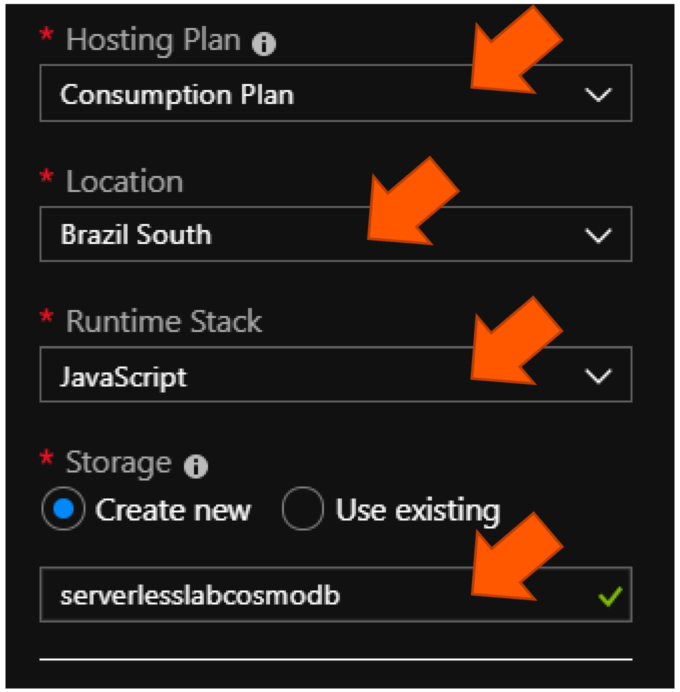
Function App
Create

* App name
serverlesslabcosmosdb ✓
.azurewebsites.net

* Subscription
Visual Studio Professional

* Resource Group
Create new [✓ Use existing]
serverlesslabcosmosdb

* OS
[✓ Windows] Linux

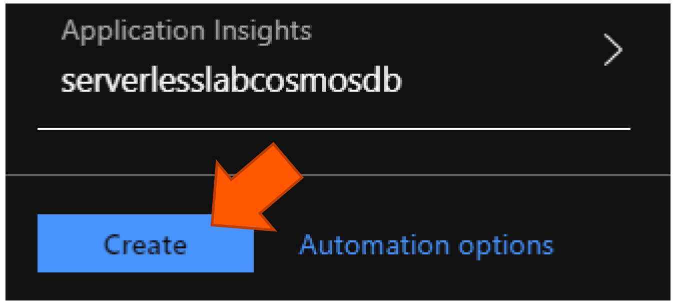


* Hosting Plan
Consumption Plan

* Location
Brazil South

* Runtime Stack
JavaScript

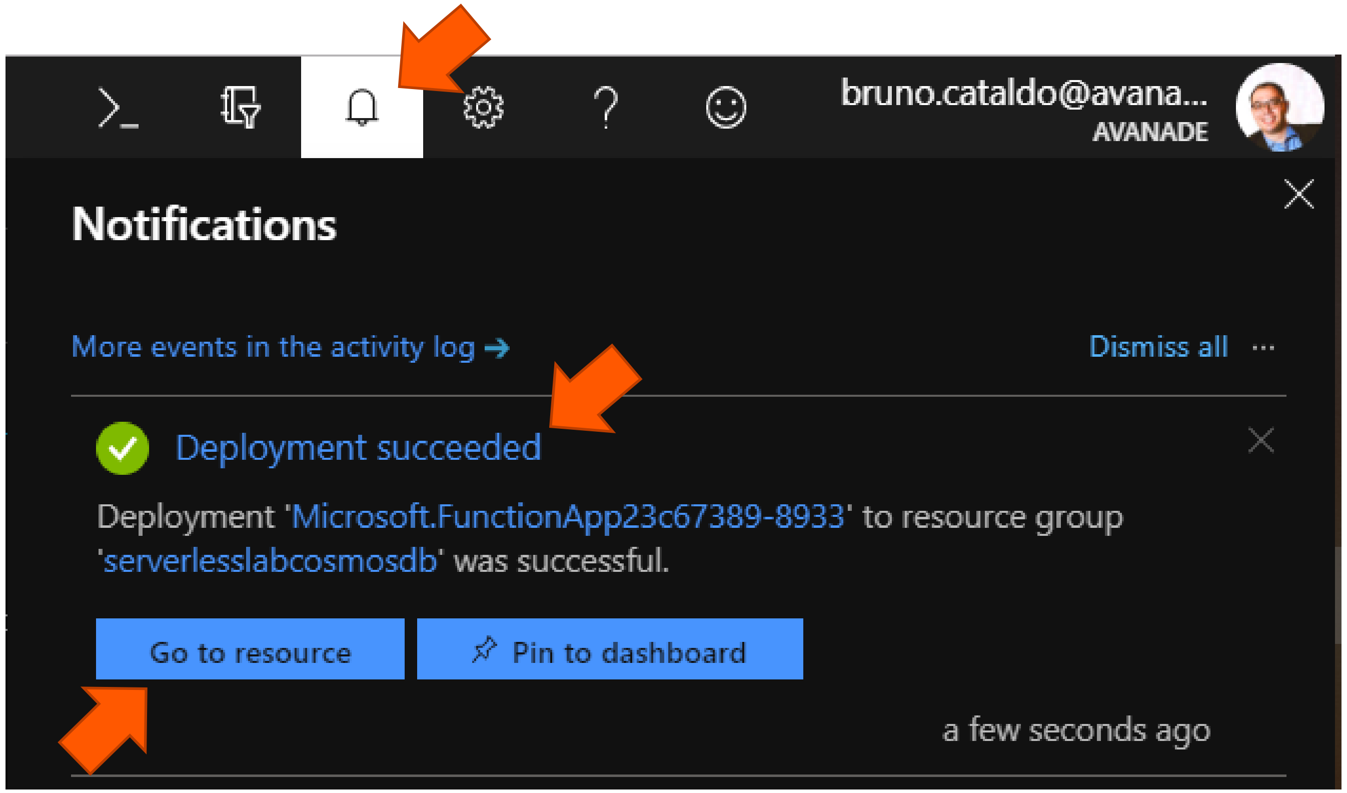
* Storage
[✓ Create new] Use existing
serverlesslabcosmodb ✓



Application Insights
serverlesslabcosmosdb

Create an Azure Function | App

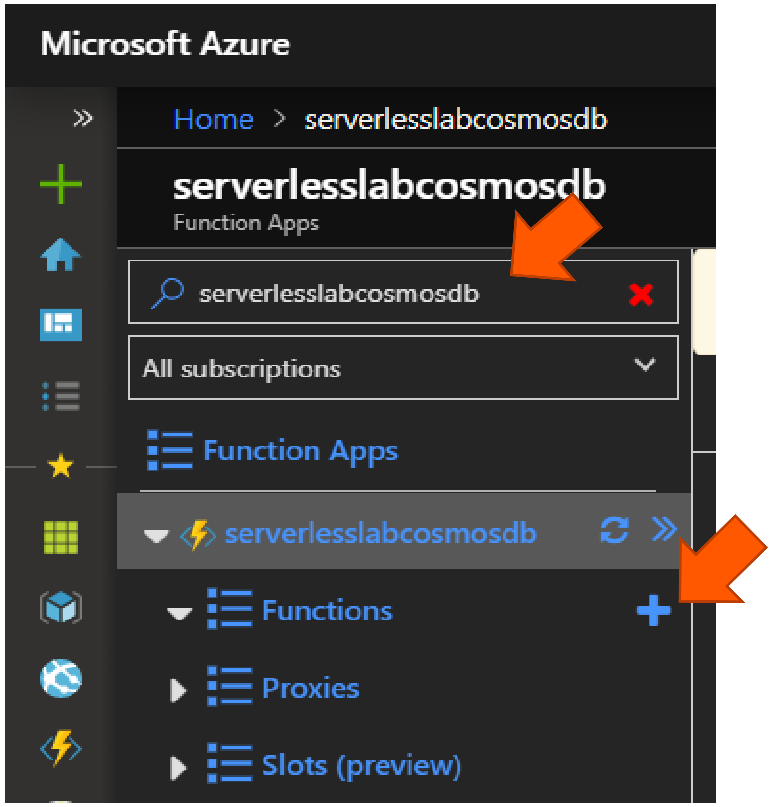
Wait for the portal page notification to display *Deployment succeeded*. Click the *Go to resource* button.



Deployment 'Microsoft.FunctionApp23c67389-8933' to resource group 'serverlesslabcosmosdb' was successful.

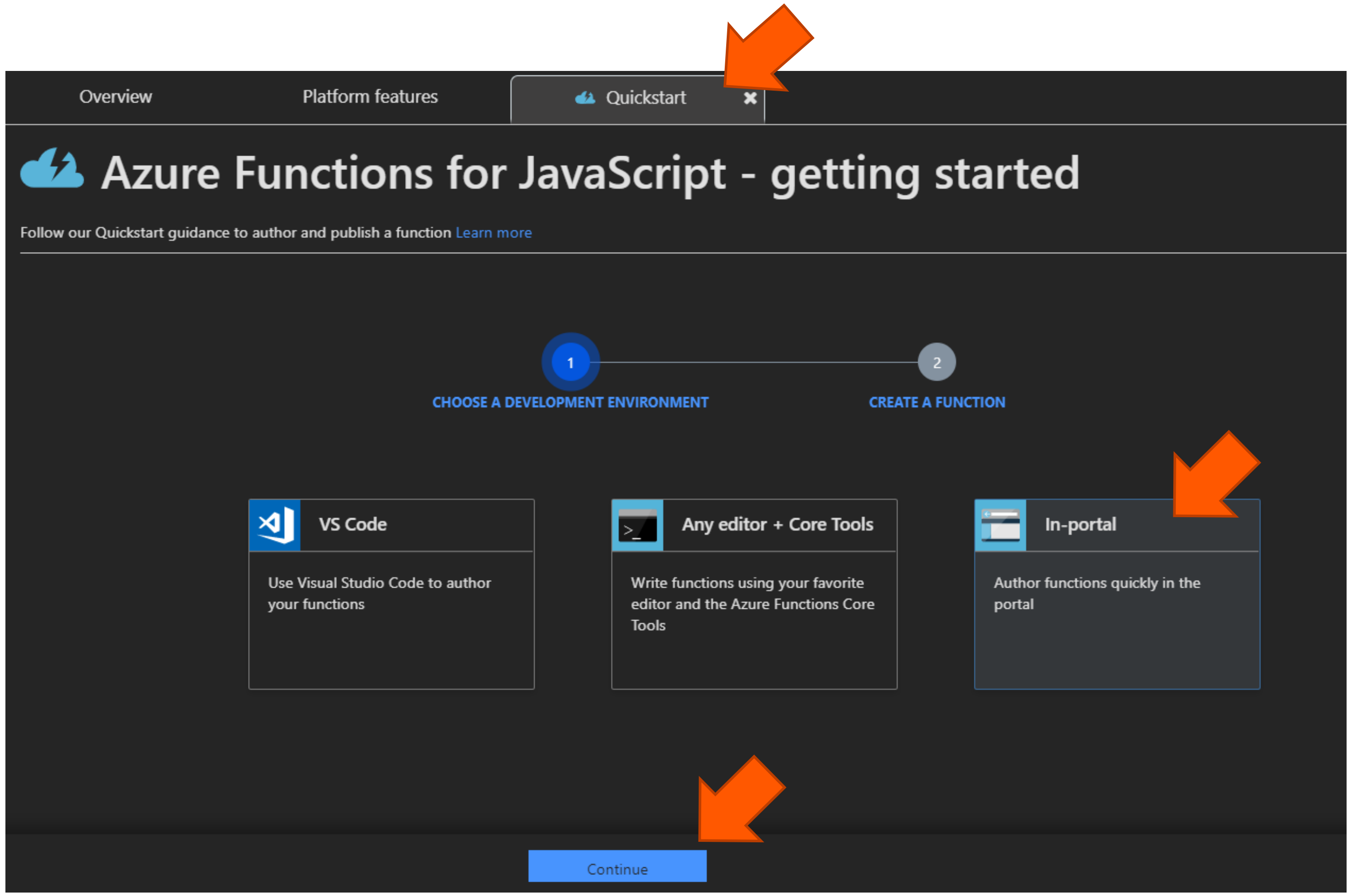
Create an Azure Function | Integrate

In the portal, navigate to the function app you created previously and expand both your function app and your function. After that click on the "+" button.



Create an Azure Function | Integrate

Click the *Quickstart* button, select the *In-portal* option and click *Continue* button.



Azure Functions for JavaScript - getting started

Follow our Quickstart guidance to author and publish a function Learn more

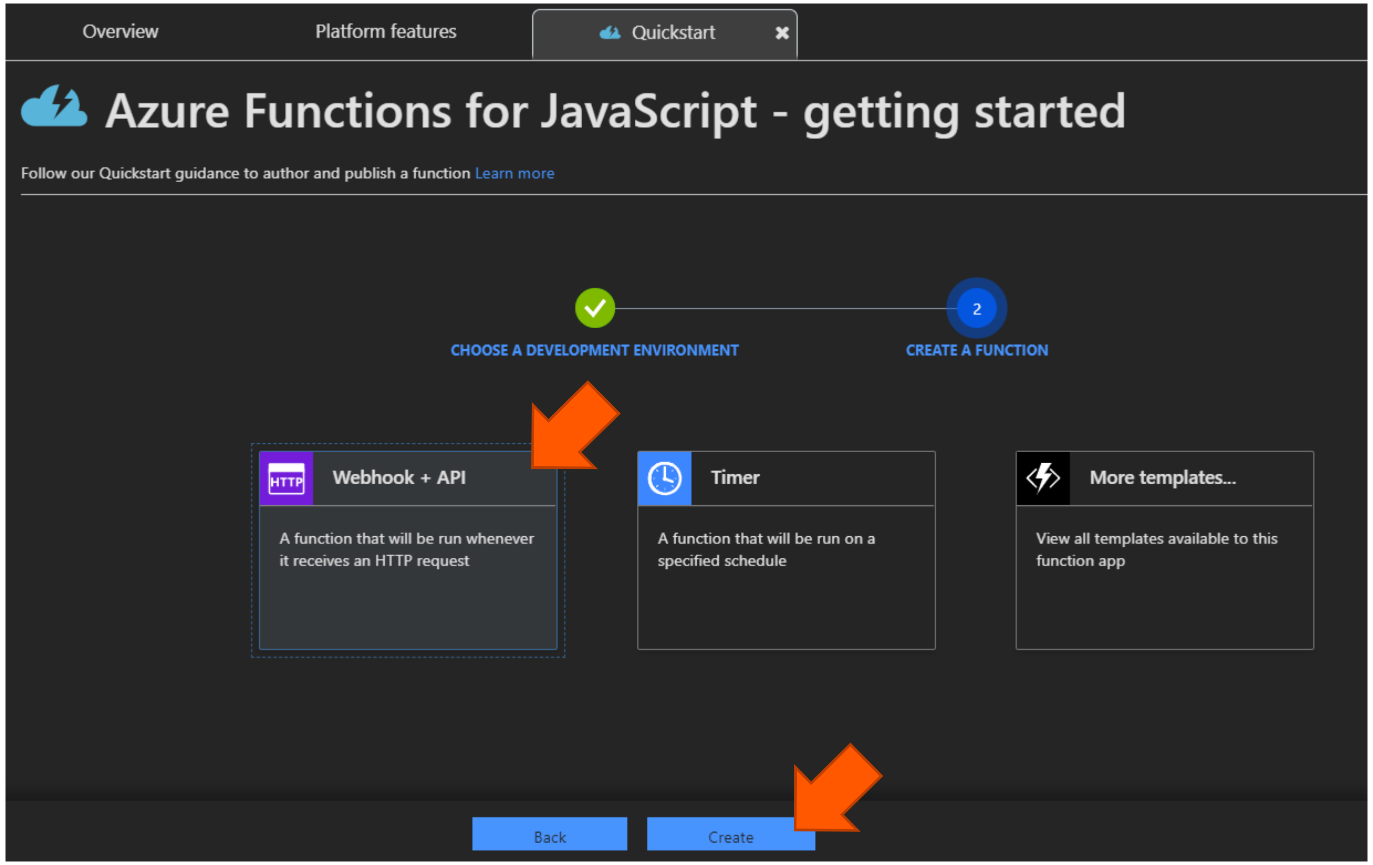
Use Visual Studio Code to author your functions

Write functions using your favorite editor and the Azure Functions Core Tools

Author functions quickly in the portal

Create an Azure Function | Integrate

Select the *Webhook + API* option, click *Create* button and refresh the page.



Azure Functions for JavaScript - getting started

Follow our Quickstart guidance to author and publish a function Learn more

Webhook + API

A function that will be run whenever it receives an HTTP request

Timer

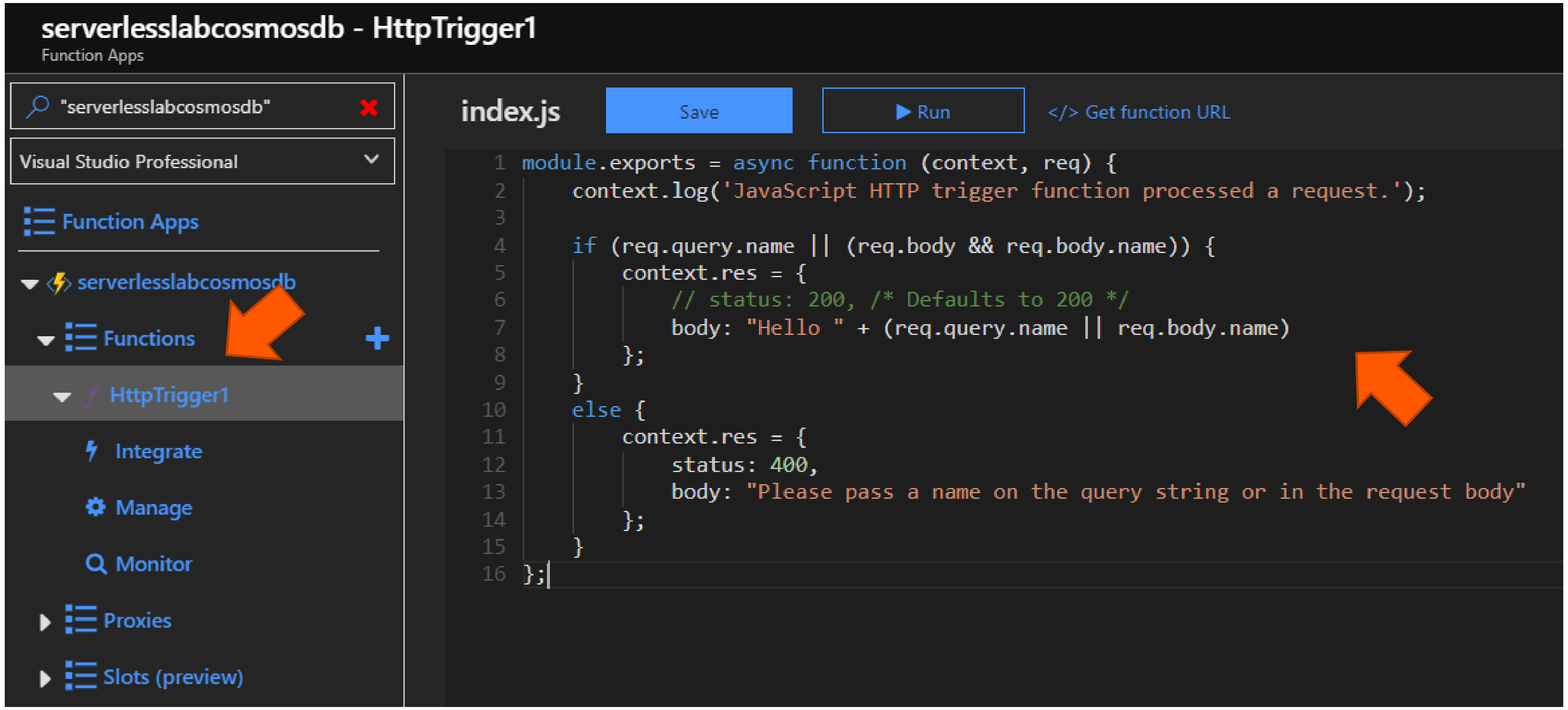
A function that will be run on a specified schedule

More templates...

View all templates available to this function app

Create an Azure Function | Integrate

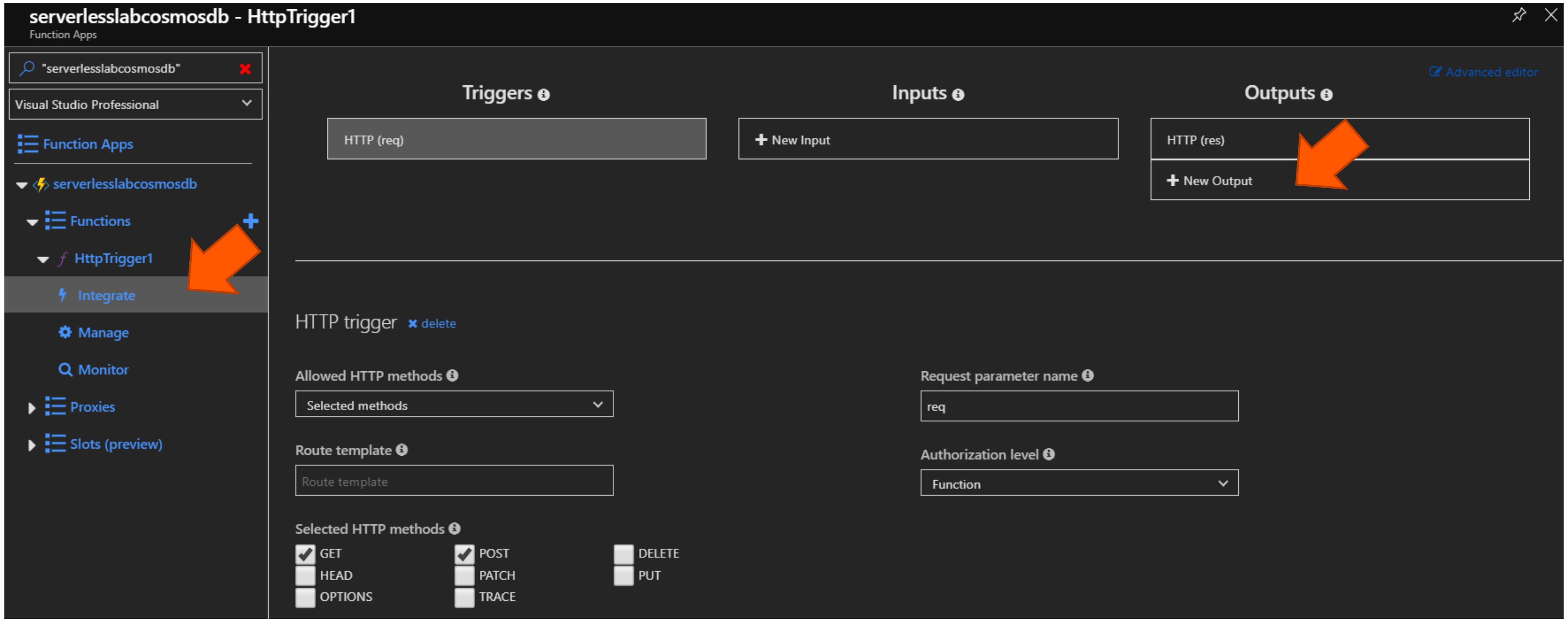
After page refresh select the *Function* option in the list and *HttpTrigger1* to view your *Azure Function* script.



```
module.exports = async function (context, req) {
    context.log('JavaScript HTTP trigger function processed a request.');

    if (req.query.name || (req.body && req.body.name)) {
        context.res = {
            // status: 200, /* Defaults to 200 */
            body: "Hello " + (req.query.name || req.body.name)
        };
    }
    else {
        context.res = {
            status: 400,
            body: "Please pass a name on the query string or in the request body"
        };
    }
};
```

Create an Azure Function | Integrate

Select the ***Integrate*** item in the list to navigate to the next page. Click ***+ New Output*** button.



serverlesslabcosmosdb - HttpTrigger1
Function Apps

HTTP trigger ✖ delete

Allowed HTTP methods: Selected methods

Request parameter name: req

Route template: Route template

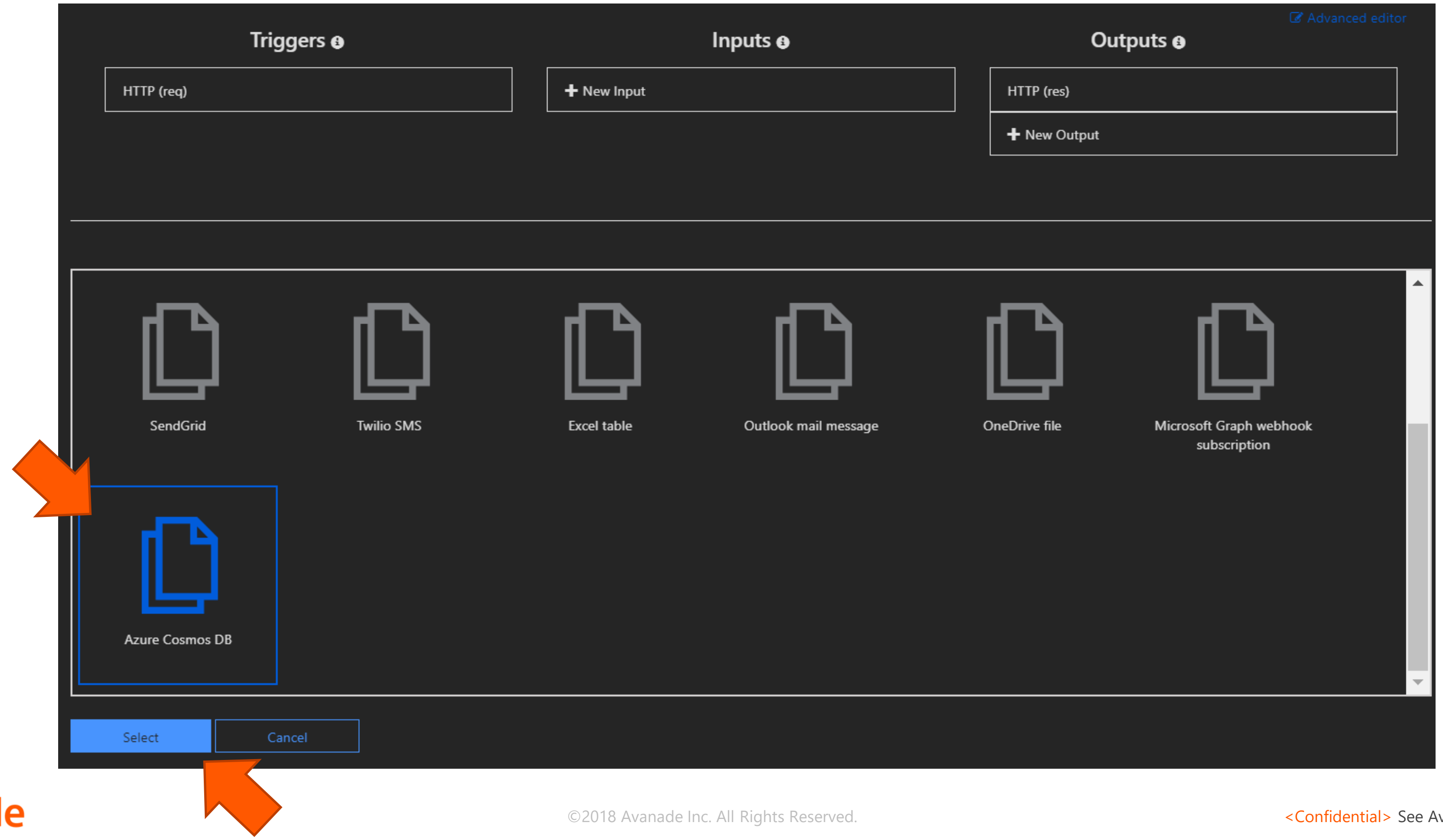
Authorization level: Function

Selected HTTP methods

✓ GET	✓ POST	DELETE
HEAD	PATCH	PUT
OPTIONS	TRACE	

Create an Azure Function | Integrate

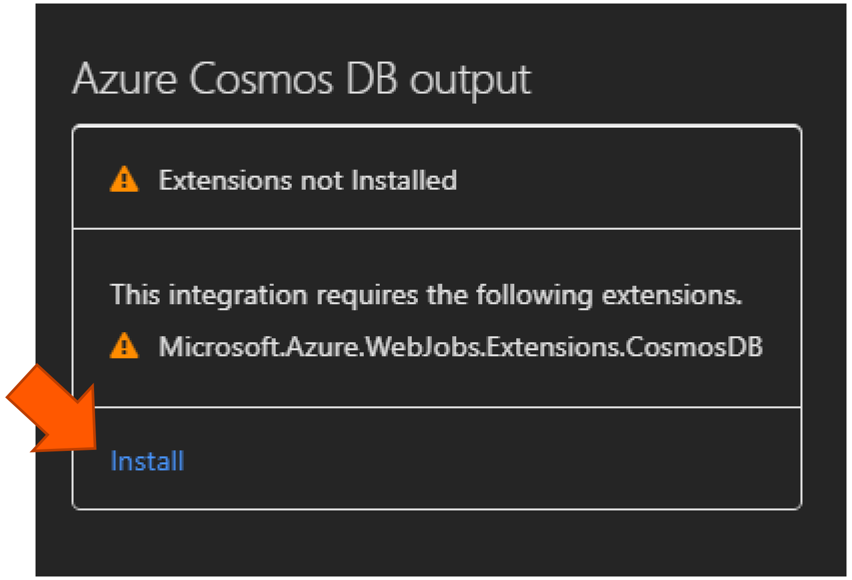
Select the *Azure Cosmo DB* item in the list and click *Select* button.



Create an Azure Function | Integrate

After reload page, scroll down page.

If you get an *Extensions* not installed message, choose Install to install the *Azure Cosmos DB* bindings extension in the function app. Installation may take a minute or two.

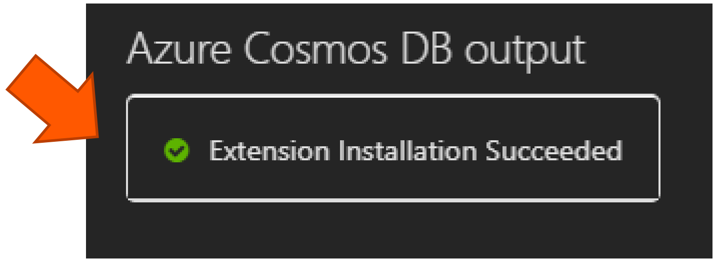


Azure Cosmos DB output

Extensions not Installed

This integration requires the following extensions.

Microsoft.Azure.WebJobs.Extensions.CosmosDB

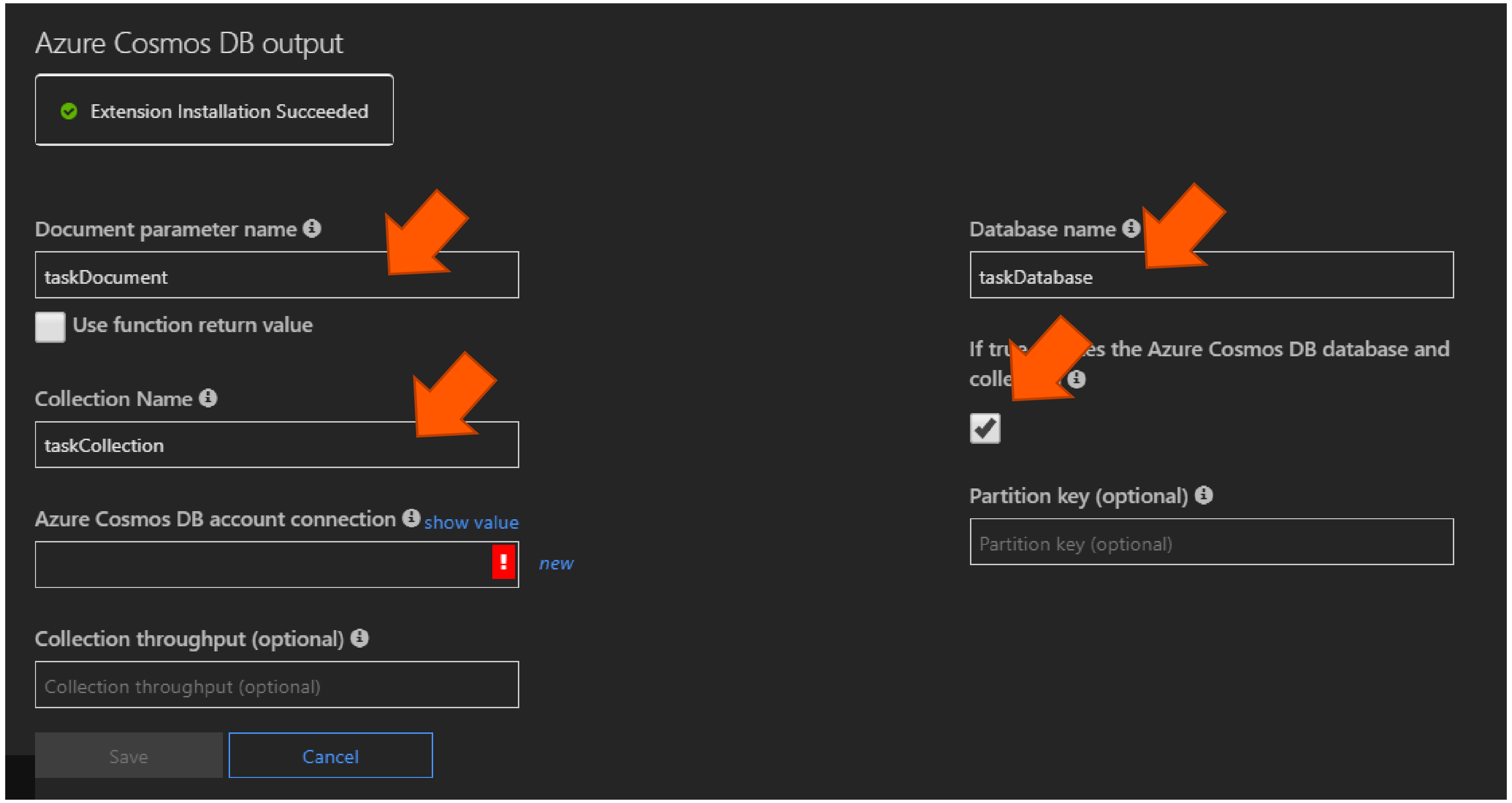


Azure Cosmos DB output

Extension Installation Succeeded

Create an Azure Function | Integrate

Use the *Azure Cosmos DB output settings* as specified in the table



Azure Cosmos DB output

Extension Installation Succeeded

Document parameter name: taskDocument

[ ] Use function return value

Collection Name: taskCollection

Azure Cosmos DB account connection

Collection throughput (optional): Collection throughput (optional)

Database name: taskDatabase

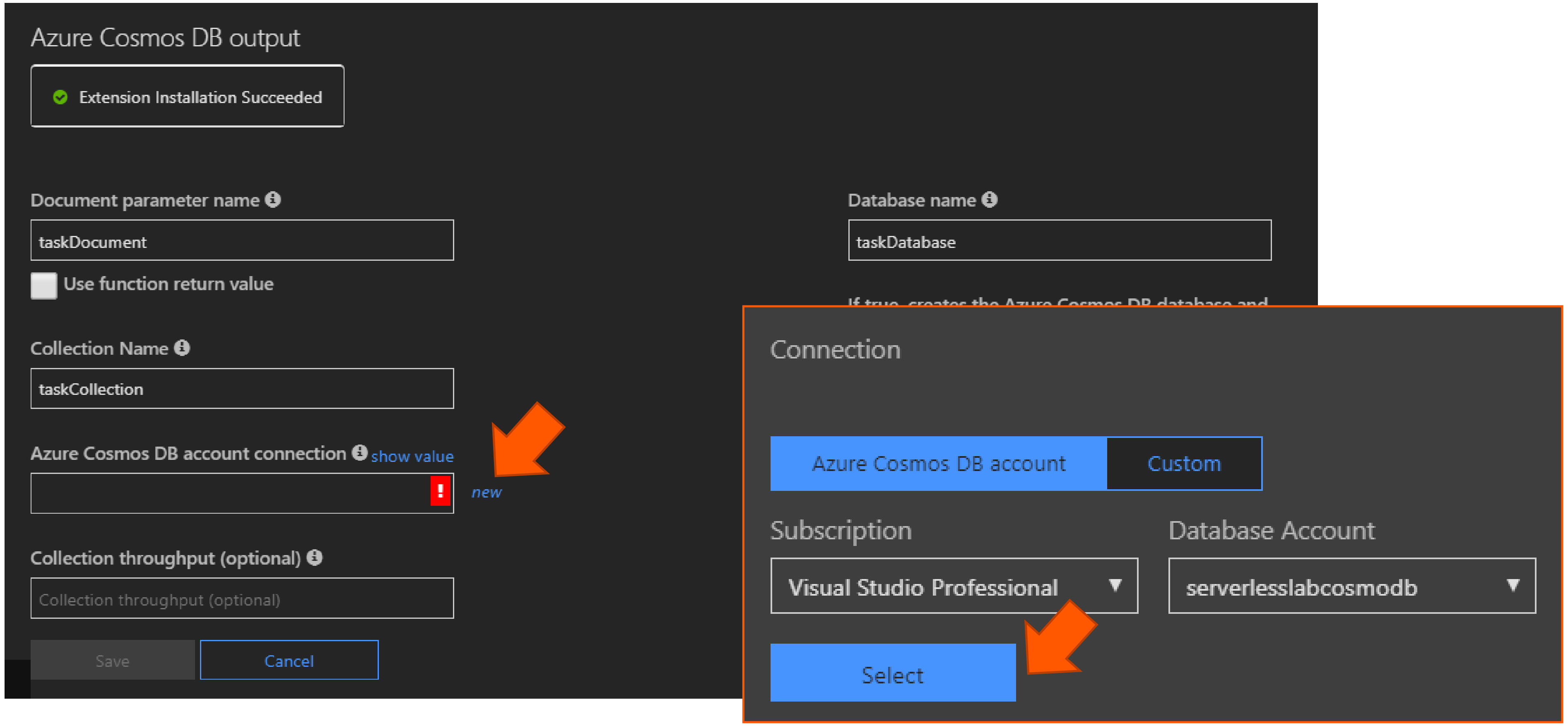
If tru[illegible]es the Azure Cosmos DB database and colle[illegible]

[✓]

Partition key (optional): Partition key (optional)

Create an Azure Function | Integrate

Select *New* to create an application setting for your account connection. Click *Select* button.



Azure Cosmos DB output

Extension Installation Succeeded

Document parameter name: taskDocument

Use function return value

Collection Name: taskCollection

Azure Cosmos DB account connection show value

new

Collection throughput (optional): Collection throughput (optional)

Database name: taskDatabase

If true, creates the Azure Cosmos DB database and

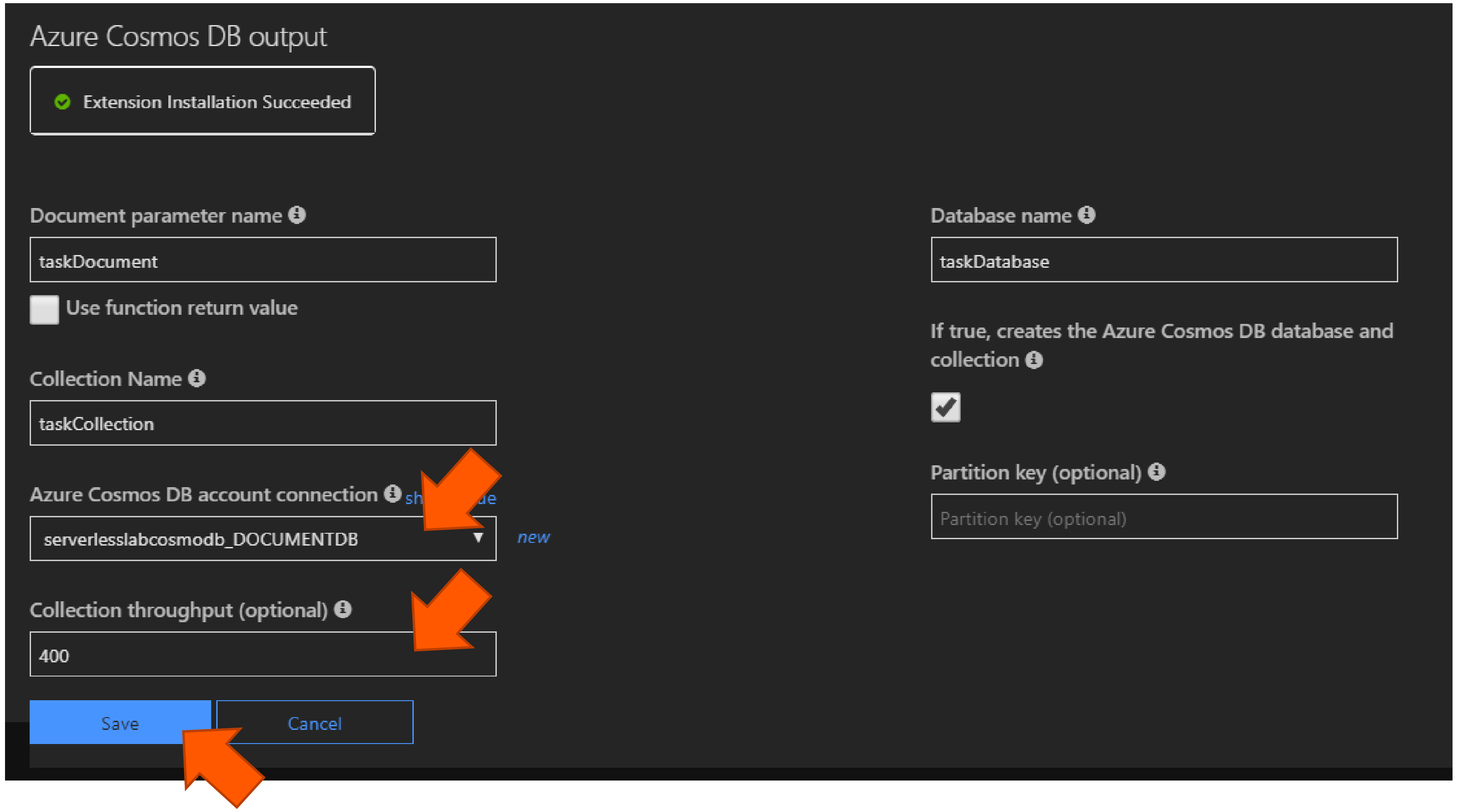
Connection

Subscription: Visual Studio Professional

Database Account: serverlesslabcosmodb

Create an Azure Function | Integrate

After all the configuration has been completed, click the *Save* button.



Azure Cosmos DB output

Extension Installation Succeeded

Document parameter name: taskDocument

☐ Use function return value

Collection Name: taskCollection

Azure Cosmos DB account connection: serverlesslabcosmodb_DOCUMENTDB

Collection throughput (optional): 400

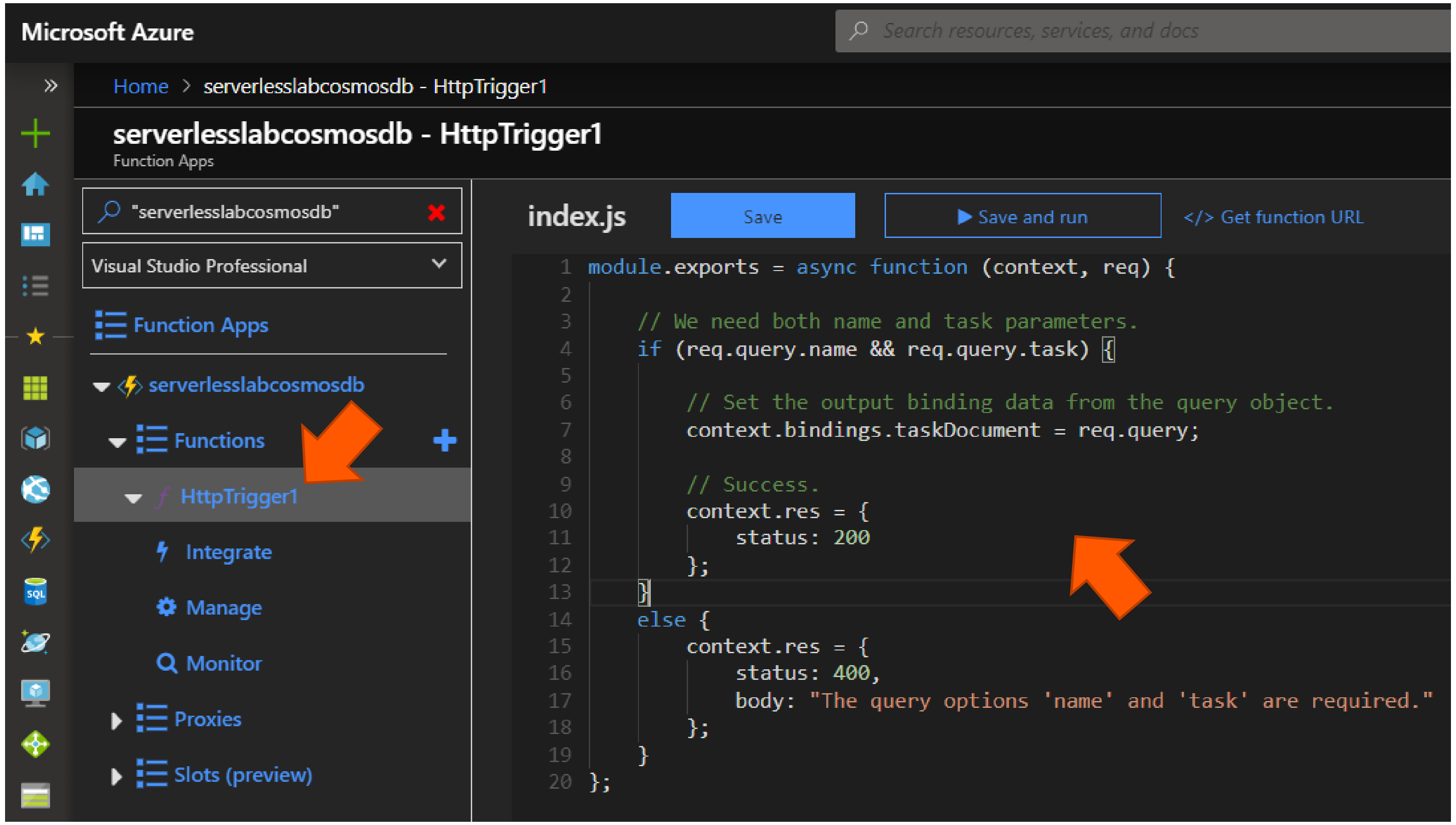
Database name: taskDatabase

If true, creates the Azure Cosmos DB database and collection: ✓

Partition key (optional): Partition key (optional)

Create an Azure Function | Integrate

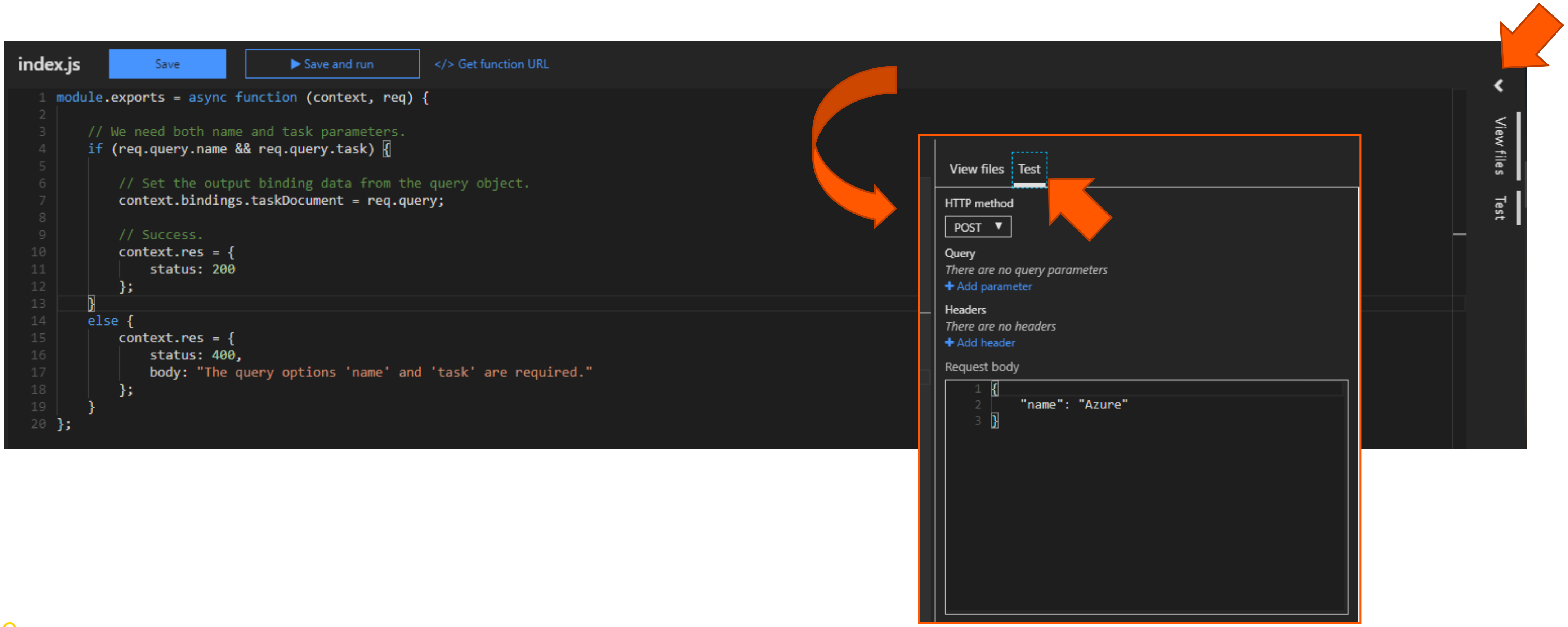
Click *HttpTrigger1* link and replace the existing *JavaScript* function with the following *code*:



```
module.exports = async function (context, req) {

    // We need both name and task parameters.
    if (req.query.name && req.query.task) {

        // Set the output binding data from the query object.
        context.bindings.taskDocument = req.query;

        // Success.
        context.res = {
            status: 200
        };
    }
    else {
        context.res = {
            status: 400,
            body: "The query options 'name' and 'task' are required."
        };
    }
};
```

Test an Azure Function | Integrate

Click on the *right* button to open the panel and click the *Test* tab to open the test panel:



```
module.exports = async function (context, req) {

    // We need both name and task parameters.
    if (req.query.name && req.query.task) {

        // Set the output binding data from the query object.
        context.bindings.taskDocument = req.query;

        // Success.
        context.res = {
            status: 200
        };
    }
    else {
        context.res = {
            status: 400,
            body: "The query options 'name' and 'task' are required."
        };
    }
};
```

HTTP method
POST

Query
There are no query parameters
+ Add parameter

Headers
There are no headers
+ Add header

Request body

```
{
    "name": "Azure"
}
```

Test an Azure Function | Integrate

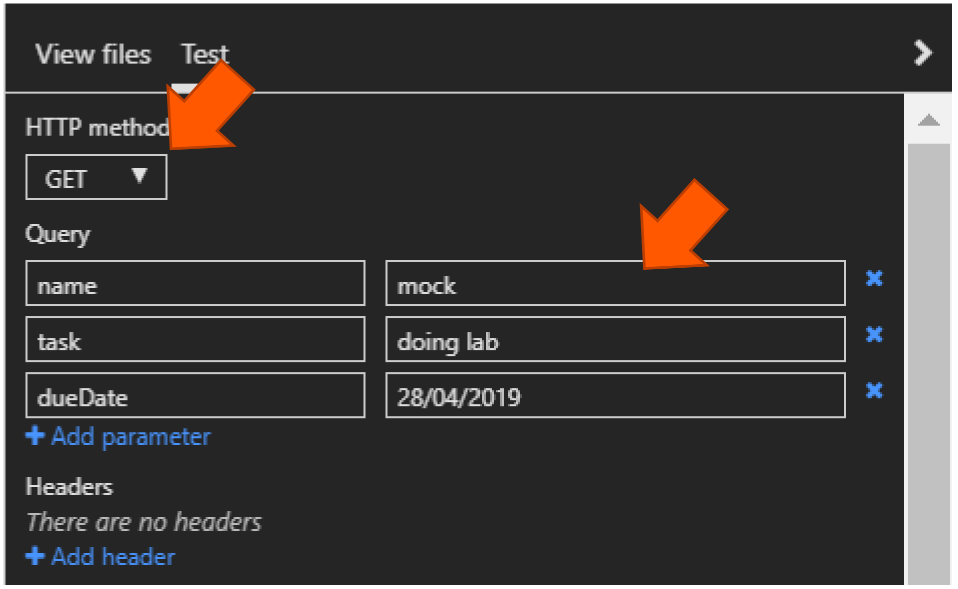
Select *HTTP method* with *GET* option.

Use the *Query fields* to fill the json request that going to be used to add an item and test.

Add 3 fields like image beside.

Click *Save and run* button.

* You can use a *Request body* to fill writing too.



HTTP method

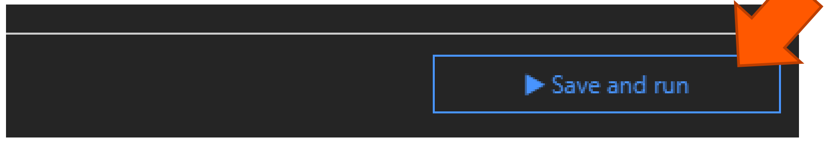
GET

Query

name	mock
task	doing lab
dueDate	28/04/2019

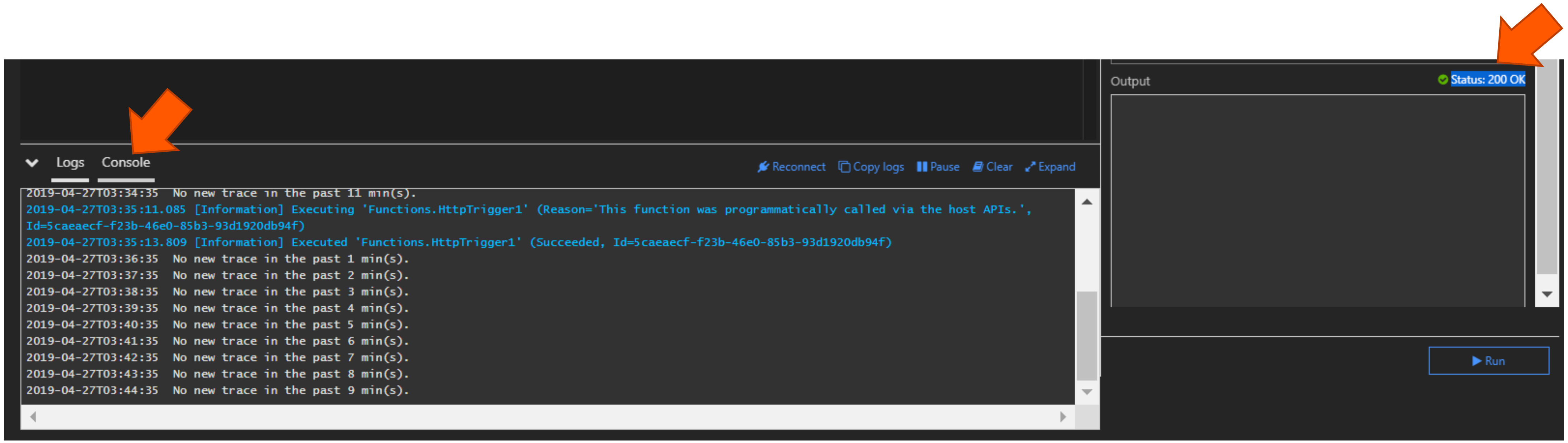
Headers

There are no headers



Test an Azure Function | Integrate

Wait for the test page to display ***Status: 200 OK*** in the output console (right side). For more details check the *Logs* and *Console* panel.



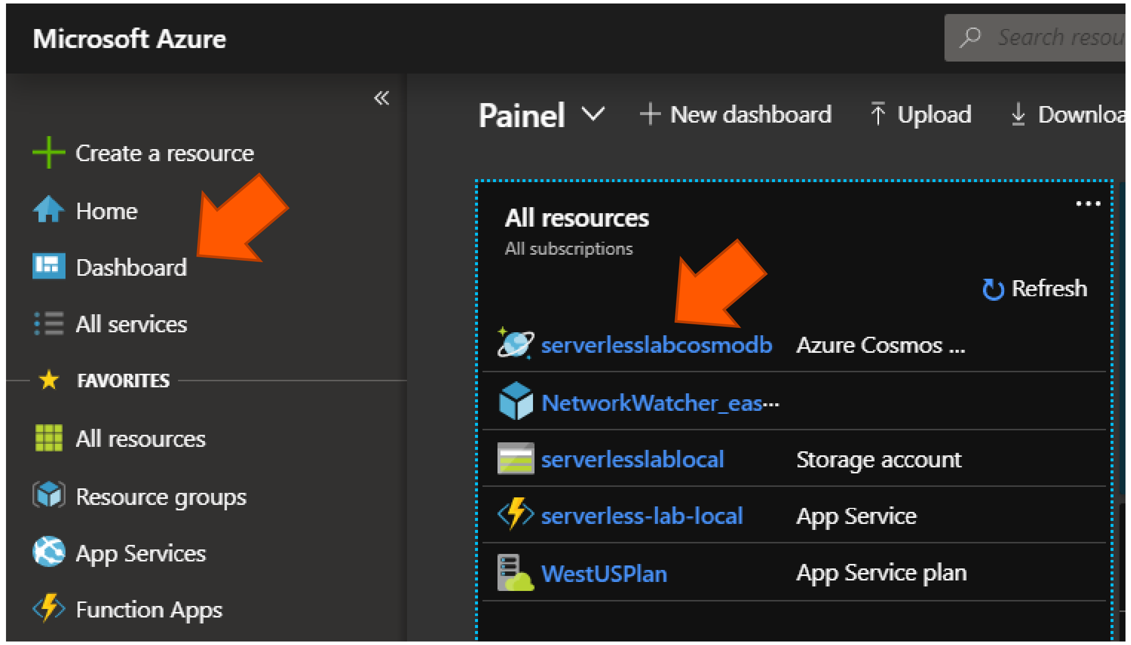
```
2019-04-27T03:34:35  No new trace in the past 11 min(s).
2019-04-27T03:35:11.085 [Information] Executing 'Functions.HttpTrigger1' (Reason='This function was programmatically called via the host APIs.',
Id=5caeaecf-f23b-46e0-85b3-93d1920db94f)
2019-04-27T03:35:13.809 [Information] Executed 'Functions.HttpTrigger1' (Succeeded, Id=5caeaecf-f23b-46e0-85b3-93d1920db94f)
2019-04-27T03:36:35  No new trace in the past 1 min(s).
2019-04-27T03:37:35  No new trace in the past 2 min(s).
2019-04-27T03:38:35  No new trace in the past 3 min(s).
2019-04-27T03:39:35  No new trace in the past 4 min(s).
2019-04-27T03:40:35  No new trace in the past 5 min(s).
2019-04-27T03:41:35  No new trace in the past 6 min(s).
2019-04-27T03:42:35  No new trace in the past 7 min(s).
2019-04-27T03:43:35  No new trace in the past 8 min(s).
2019-04-27T03:44:35  No new trace in the past 9 min(s).
```

Check data on Azure Cosmos DB | Integrate

You must have to access *Dashboard* and find your *Azure Cosmos DB* resource.

Select:

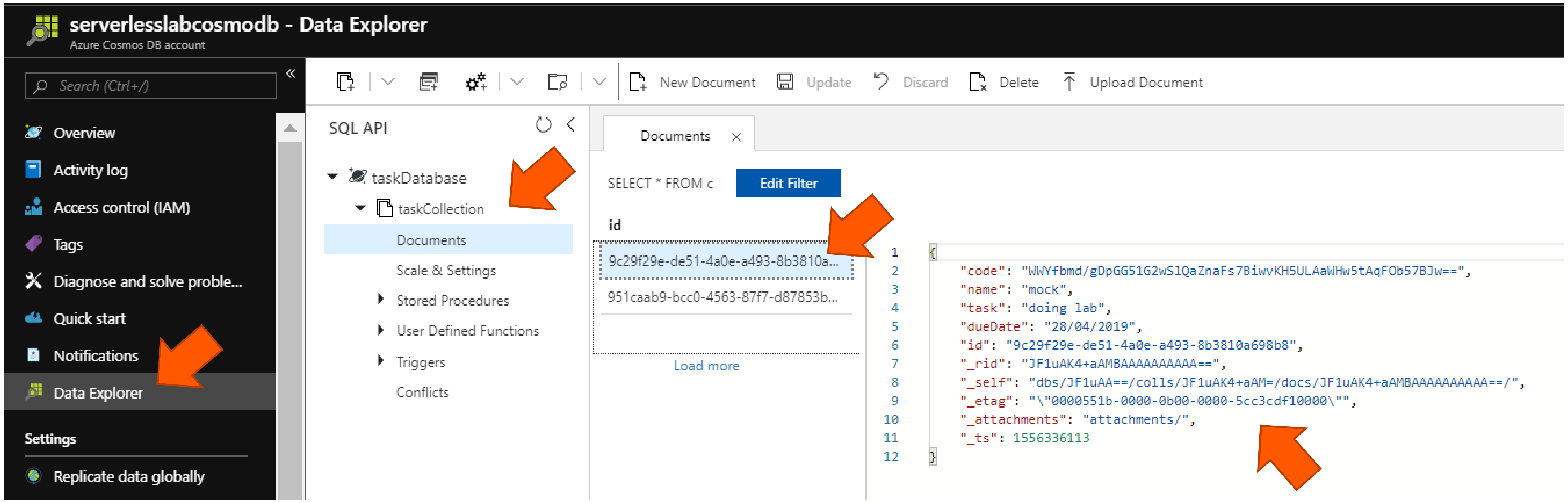
> *Dashboard*
>
> > *serverlesslabcosmodb*



serverlesslabcosmodb	Azure Cosmos ...
NetworkWatcher_eas...	
serverlesslablocal	Storage account
serverless-lab-local	App Service
WestUSPlan	App Service plan

Check data on Azure Cosmos DB | Integrate

Select *Data Explorer* item, expand *taskDatabase*, expand *taskCollection*, click on *Documents* and *hash indetificator* like image. Check the data.

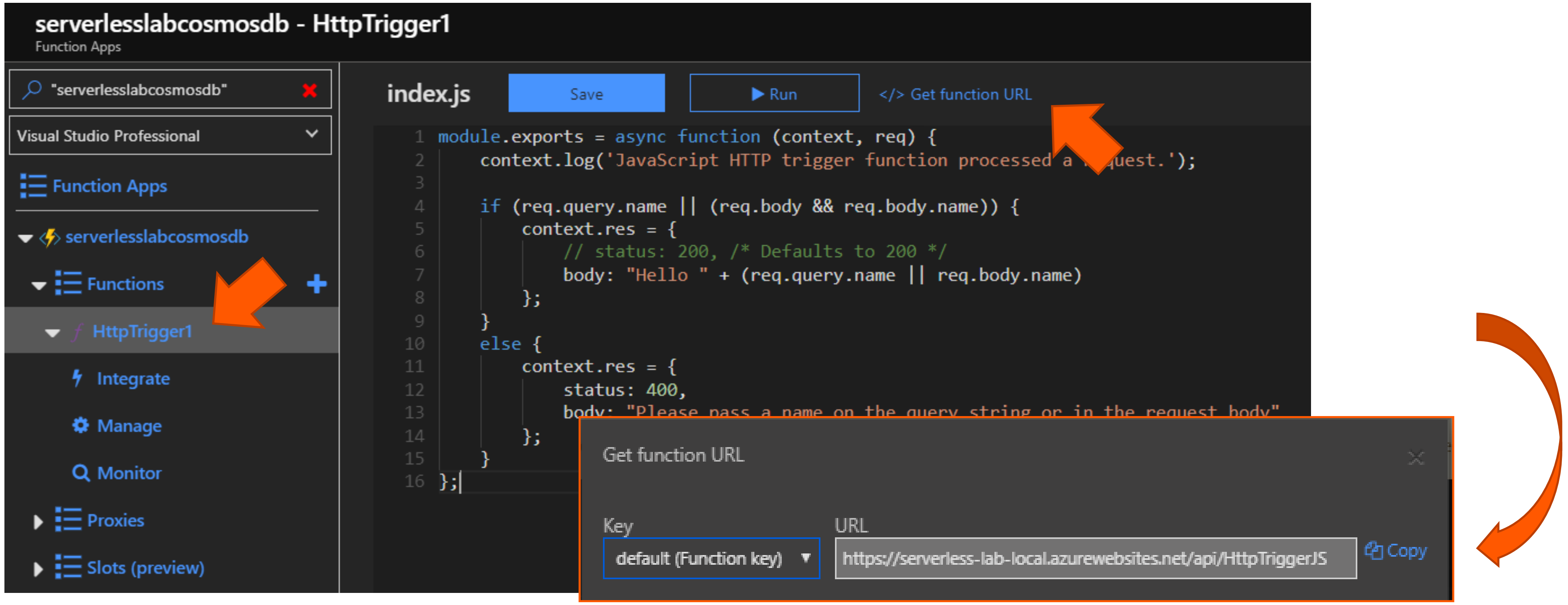


```
{
    "code": "WWYfbmd/gDpGG51G2wS1QaZnaFs7BiwvKH5ULAaWHw5tAqFOb57BJw==",
    "name": "mock",
    "task": "doing lab",
    "dueDate": "28/04/2019",
    "id": "9c29f29e-de51-4a0e-a493-8b3810a698b8",
    "_rid": "JF1uAK4+aAMBAAAAAAAAAA==",
    "_self": "dbs/JF1uAA==/colls/JF1uAK4+aAM=/docs/JF1uAK4+aAMBAAAAAAAAAA==/",
    "_etag": "\"0000551b-0000-0b00-0000-5cc3cdf10000\"",
    "_attachments": "attachments/",
    "_ts": 1556336113
}
```

Test an Azure Function | Integrate + Browser

Click the "*</> Get function URL*" link like the image below.

Using the button to *copy* the URL.



```
module.exports = async function (context, req) {
    context.log('JavaScript HTTP trigger function processed a request.');

    if (req.query.name || (req.body && req.body.name)) {
        context.res = {
            // status: 200, /* Defaults to 200 */
            body: "Hello " + (req.query.name || req.body.name)
        };
    }
    else {
        context.res = {
            status: 400,
            body: "Please pass a name on the query string or in the request body"
        };
    }
};
```

Key: default (Function key)

URL: https://serverless-lab-local.azurewebsites.net/api/HttpTriggerJS

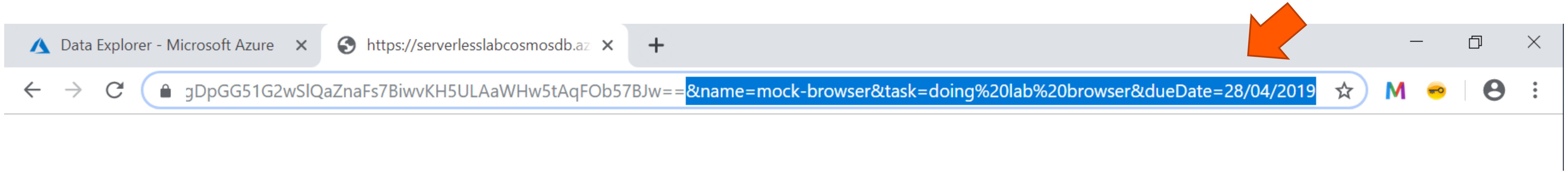
Test an Azure Function | Integrate + Browser

Paste the URL for the HTTP request into your browser's address bar. Append the query string in the end and execute the request.

&name=<yourname>

&task=<yourname>

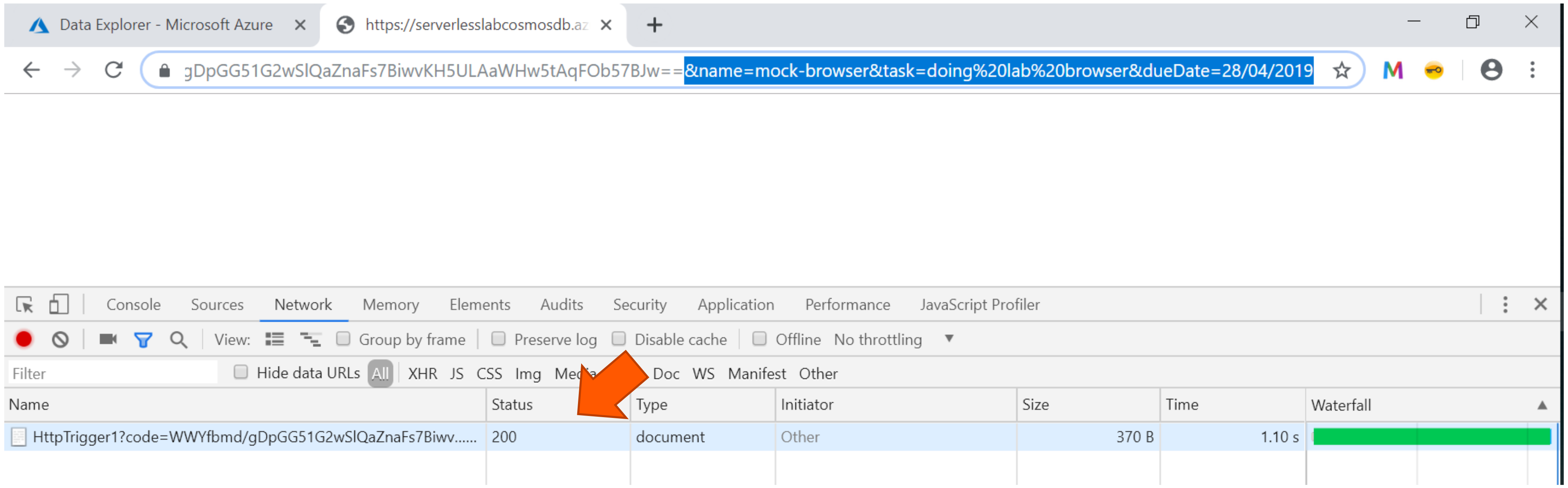
&dueDate=<yourname>


Data Explorer - Microsoft Azure
https://serverlesslabcosmosdb.az

```
gDpGG51G2wSlQaZnaFs7BiwvKH5ULAaWHw5tAqFOb57BJw==&name=mock-browser&task=doing%20lab%20browser&dueDate=28/04/2019
```

Test an Azure Function | Integrate + Browser

The following shows the response ***status 200*** in the browser to the ***GET*** request using ***F12*** or ***Network*** console of your browser.



Data Explorer - Microsoft Azure

https://serverlesslabcosmosdb.az

```
gDpGG51G2wSlQaZnaFs7BiwvKH5ULAaWHw5tAqFOb57BJw==&name=mock-browser&task=doing%20lab%20browser&dueDate=28/04/2019
```

Console Sources Network Memory Elements Audits Security Application Performance JavaScript Profiler
View: Group by frame Preserve log Disable cache Offline No throttling
Filter Hide data URLs All XHR JS CSS Img Media Doc WS Manifest Other

Name	Status	Type	Initiator	Size	Time	Waterfall
HttpTrigger1?code=WWYfbmd/gDpGG51G2wSlQaZnaFs7Biwv......	200	document	Other	370 B	1.10 s	🟩

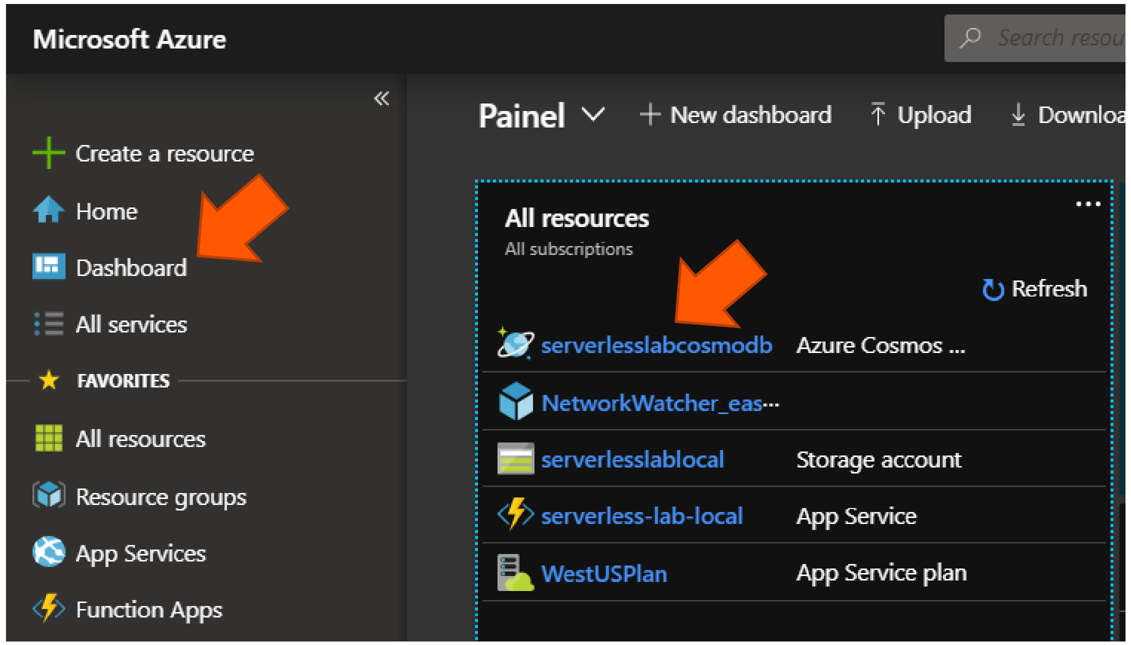
Check data on Azure Cosmos DB | Integrate

You must have to access *Dashboard* and find your *Azure Cosmos DB* resource.

Select:

> *Dashboard*

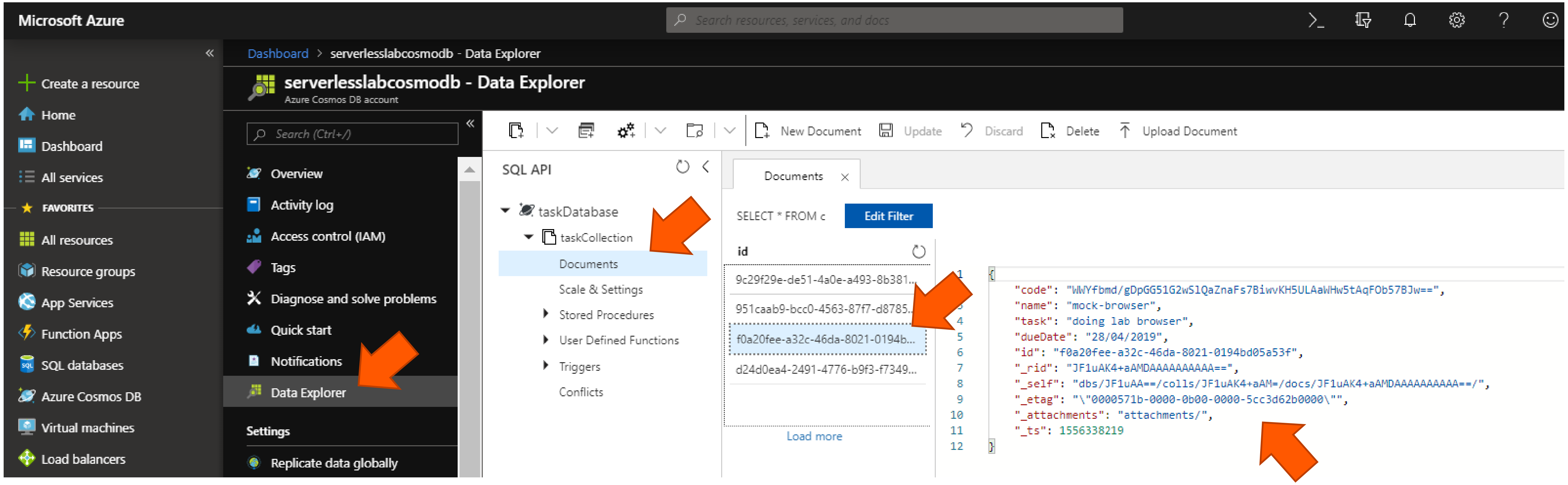
> > *serverlesslabcosmodb*



serverlesslabcosmodb	Azure Cosmos ...
NetworkWatcher_eas...	
serverlesslablocal	Storage account
serverless-lab-local	App Service
WestUSPlan	App Service plan

Check data on Azure Cosmos DB | Integrate

Select *Data Explorer* item, expand *taskDatabase*, expand *taskCollection*, click on *Documents* and *hash indetificator* like image. Check the data.



SELECT * FROM c

id
9c29f29e-de51-4a0e-a493-8b381...
951caab9-bcc0-4563-87f7-d8785...
f0a20fee-a32c-46da-8021-0194b...
d24d0ea4-2491-4776-b9f3-f7349...

```
{
    "code": "WWYfbmd/gDpGG51G2wS1QaZnaFs7BiwvKH5ULAaWHw5tAqFOb57BJw==",
    "name": "mock-browser",
    "task": "doing lab browser",
    "dueDate": "28/04/2019",
    "id": "f0a20fee-a32c-46da-8021-0194bd05a53f",
    "_rid": "JF1uAK4+aAMDAAAAAAAAAA==",
    "_self": "dbs/JF1uAA==/colls/JF1uAK4+aAM=/docs/JF1uAK4+aAMDAAAAAAAAAA==/",
    "_etag": "\"0000571b-0000-0b00-0000-5cc3d62b0000\"",
    "_attachments": "attachments/",
    "_ts": 1556338219
}
```