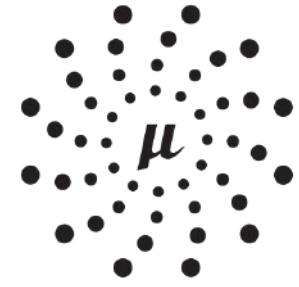
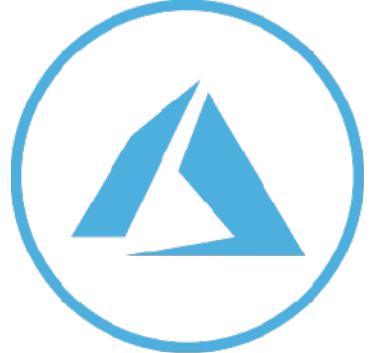




OBJECT
COMPUTING



MICRONAUT



Micronaut + Azure Functions

Speakers

Julien Dubois (Microsoft)

Java developer Advocacy team manager at Microsoft.

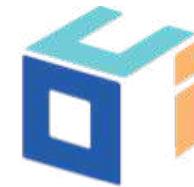
He is the creator and lead developer of the [JHipster](#) project and Java Champion.



Sergio del Amo (Object Computing)

Senior Software Engineer of the Micronaut and Grails teams at Object Computing.

He writes the newsletter [Groovy Calamari](#) and organizes the conference [GreachConf](#).

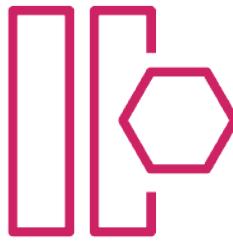


OBJECT COMPUTING
HOME TO GRAILS & MICRONAUT

What is Serverless?



Abstraction of servers



Event-driven / instant scale



Micro-billing

Azure Function Core Tools

<https://docs.microsoft.com/en-us/cli/azure>

Azure Function Core Tools includes a version of the same runtime that powers Azure Functions runtime that you can run on your local development computer. It also provides commands to create functions, connect to Azure, and deploy function projects.

Prerequisites

Azure CLI

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-run-local>

Azure CLI is a set of commands used to create and manage Azure resources. The Azure CLI is available across Azure services and is designed to get you working quickly with Azure, with an emphasis on automation.

Prerequisites



Creating a function

Creating a function app



A screenshot of the Microsoft Azure portal homepage. At the top, there's a blue header bar with the "Microsoft Azure" logo, a search bar containing "Search resources, services, and docs (G+)", and several icons for account management. Below the header is a section titled "Azure services" with a grid of service tiles. The tiles include: "Create a resource" (with a plus sign icon), "Virtual machines" (with a monitor icon), "App Services" (with a globe icon), "Storage accounts" (with a bar chart icon), "SQL databases" (with a cylinder icon), "Azure Database for PostgreSQL" (with a database icon), "Azure Cosmos DB" (with a planet icon), "Kubernetes services" (with a cluster icon), and "Function App" (with a lightning bolt icon). A red box highlights the "Function App" tile.

Creating a function app



A screenshot of the Microsoft Azure portal's "Function App" blade. The top navigation bar includes the Microsoft Azure logo, an "Upgrade" button, a search bar, and various account and service icons. Below the navigation is a breadcrumb trail: "Home > Function App". The main title is "Function App" with a gear icon. A red box highlights the "Add" button in the toolbar, which has a plus sign and a lightning bolt icon. The toolbar also includes "Manage view", "Refresh", "Export to CSV", "Open query", "Assign tags", "Start", "Restart", "Stop", "Delete", and "Feedback" buttons. Below the toolbar are filter options: "Filter by name...", "Subscription == all", "Resource group == all", "Location == all", and "Add filter". The message "Showing 0 to 0 of 0 records." is displayed. A table header row shows columns for "Name", "Status", "Location", "Pricing Tier", "App Service Plan", and "Subscription". The main content area features a large lightning bolt icon and the text "No function apps to display". A note below states: "The resources are currently filtered and not all resources may be displayed, such as hidden resources." It also says "Try changing your filters if you don't see what you're looking for." and provides a link to "Learn more about App Service". At the bottom are two buttons: a large blue "Create Function App" button with a red border, and a smaller white "Clear filters / Show hidden" button.

Project Details

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



Subscription * ⓘ

Free Trial ✓

Resource Group * ⓘ

examplemicronaut ✓

[Create new](#)

Instance Details

Function App name *

testmicronaut ✓

.azurewebsites.net

Publish *

Code Docker Container

Runtime stack *

Java ✓

Version *

11.0 (Preview) ✓

Region *

Central US ✓

Storage

When creating a function app, you must create or link to a general-purpose Azure Storage account that supports Blobs, Queue, and Table storage.

Storage account *

(New) storageaccountexampleblob9



[Create new](#)

Operating system

The Operating System has been recommended for you based on your selection of runtime stack.

Operating System *

Linux Windows

Plan

The plan you choose dictates how your app scales, what features are enabled, and how it is priced. [Learn more](#)

Plan type *

Consumption (Serverless)



Creating a function app



Basics Hosting **Monitoring** Tags Review + create

Application Insights is a code-less attach to provide detailed observability in to your application. [Learn more](#)

Application Insights

Enable Application Insights *

No Yes

Application Insights *

(New) testmicronaut202009301652 (Central US)



[Create new](#)

Region

Central US

Creating a function app



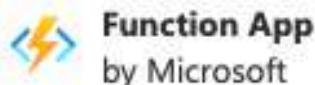
Basics Hosting Monitoring **Tags** Review + create

Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups.

Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.

Name ⓘ	Value ⓘ	Resource
project	: micronaut-azure-webinar	4 selected ▾ ...
	:	4 selected ▾

Summary



Details

Subscription	9825e0b9-244a-4eeb-9194-d3e8123f
Resource Group	examplemicronaut
Name	testmicronaut
Runtime stack	Java 11.0 (Preview)
Tags	project: micronaut-azure-webinar

Plan (New)

Plan type	Consumption (Serverless)
Name	ASP-examplemicronaut-9315
Operating System	Linux
Region	Central US
SKU	Dynamic
Tags	project: micronaut-azure-webinar

Hosting

Storage (New)	
Storage account	storageaccountexamplepbdb9
Tags	project: micronaut-azure-webinar

Monitoring (New)

Application Insights	Enabled
Name	testmicronaut202009301652
Region	Central US
Tags	project: micronaut-azure-webinar

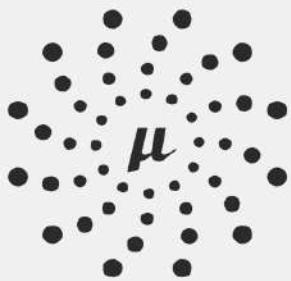
Micronaut Azure Modules



Dependency	Description
Simple Azure Functions	<code>io.micronaut.azure:micronaut-azure-function</code>
Azure HTTP Functions	<code>io.micronaut.azure:micronaut-azure-function-http</code>



Creating an Azure function application



MICRONAUT™

LAUNCH



Application Type

Application

Java Version

11

Base Package

com.example

Name

demo

Micronaut Version

2.0.3

2.1.0.BUILD-SNAPSHOT

Language

Java

Kotlin

Groovy

Build

Gradle

Maven

Test Framework

JUnit

Spock

KoTest



FEATURES



DIFF



PREVIEW



GENERATE PROJECT

Included Features (1)

azure-function X

Micronaut Azure dependencies



```
dependencies {  
    ...  
    implementation("com.microsoft.azure.functions:azure-functions-java-library")  
    implementation("io.micronaut.azure:micronaut-azure-function-http")  
}
```

Creating a function app



```
public class Function extends AzureHttpFunction {  
    @FunctionName("testmicronaut")  
    public HttpResponseMessage invoke(  
        @HttpTrigger(  
            name = "req",  
            methods = {HttpMethod.POST, HttpMethod.GET},  
            route = "{*route}",  
            authLevel = AuthorizationLevel.ANONYMOUS)  
        HttpRequestMessage<Optional<String>> request,  
        final ExecutionContext context) {  
        return super.route(request, context);  
    }  
}
```





```
import edu.umd.cs.findbugs.annotations.NonNull;  
  
import javax.validation.constraints.Pattern;  
  
public interface NameTransformer {  
  
    @NonNull  
    String transform(@NonNull @Pattern(regexp = "sergio") String name);  
}
```



```
import edu.umd.cs.findbugs.annotations.NonNull;  
import javax.inject.Singleton;  
  
@Singleton  
public class UppercaseNameTransformer implements NameTransformer {  
  
    @NonNull  
    @Override  
    public String transform(@NonNull String name) {  
        return name.toUpperCase();  
    }  
}
```





```
import io.micronaut.http.annotation.Body;
import io.micronaut.http.annotation.Controller;
import io.micronaut.http.annotation.Post;

@Controller("/person")
public class PersonController {

    private final NameTransformer nameTransformer;

    public PersonController(NameTransformer nameTransformer) {
        this.nameTransformer = nameTransformer;
    }

    @Post
    public Person index(@Body Person person){
        return new Person(nameTransformer.transform(person.getName()));
    }
}
```



```
import io.micronaut.core.annotation.Introspected;  
  
@Introspected  
public class Person {  
  
    private String name;  
  
    public Person() {}  
  
    public Person(String name) {  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
}
```





Run the function locally



[https://plugins.gradle.org/plugin/
com.microsoft.azure.azurefunctions](https://plugins.gradle.org/plugin/com.microsoft.azure.azurefunctions)



[https://docs.microsoft.com/en-us/java/api/overview/
azure/maven/azure-functions-maven-plugin/readme](https://docs.microsoft.com/en-us/java/api/overview/azure/maven/azure-functions-maven-plugin/readme)

RUN THE FUNCTION LOCALLY



```
% ./gradlew azureFunctionRun
```

```
...
```

Functions:

```
testmicronaut: [POST,GET] http://localhost:7071/api/{*route}
```

```
...
```

```
> :azureFunctionsRun
```

```
% curl -X POST -d "{\"name\": \"sergio\"}" http://localhost:7071/api/person  
{"name":"SERGIO"}
```



Deploy the function

DEPLOY THE FUNCTION



```
azurefunctions {
    resourceGroup = 'examplemicronaut'
    appName = 'micronautest'
    pricingTier = 'Consumption'
    region = 'westus'
    runtime {
        os = 'linux'
    }
    localDebug = "transport=dt_socket,server=y,suspend=n,address=5005"
}
```

DEPLOY THE FUNCTION



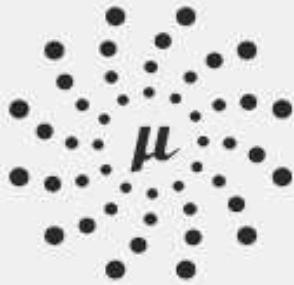
```
% ./gradlew azureFunctionsDeploy  
Successfully deployed the artifact to https://testmicronaut.azurewebsites.net  
Successfully deployed the function app at https://testmicronaut.azurewebsites.net  
  
% curl -X POST -d "{\"name\": \"sergio\"}" https://testmicronaut.azurewebsites.net/api/person  
{"name":"SERGIO"}
```



Creating an Azure serverless function

Dependency	Description
Timer	A set interval
HTTP	An HTTP request is received
Blob	file is uploaded or updated in Azure blob storage
Queue	A message is added to an Azure Storage queue
Cosmos DB	A document changes in a collection
Event Grid	An event hub receives a new event

Check classes inside package
`com.microsoft.azure.functions.annotation`



MICRONAUT™ LAUNCH



Application Type	Java Version	Base Package	Name
Serverless Function	11	com.example	demo
Micronaut Version	Language	Build	Test Framework
<input checked="" type="radio"/> 2.0.3	<input checked="" type="radio"/> Java	<input checked="" type="radio"/> Gradle	<input checked="" type="radio"/> JUnit
<input type="radio"/> 2.1.0.BUILD-SNAPSHOT	<input type="radio"/> Kotlin	<input type="radio"/> Maven	<input type="radio"/> Spock
	<input type="radio"/> Groovy		<input type="radio"/> KoTest

FEATURES **DIFF** **PREVIEW** **GENERATE PROJECT**

Included Features (1)

azure-function X

Micronaut Azure dependencies



```
dependencies {  
    ...  
    implementation("com.microsoft.azure.functions:azure-functions-java-library")  
    implementation("io.micronaut.azure:micronaut-azure-function")  
}
```



TimeTrigger



```
import com.microsoft.azure.functions.ExecutionContext;
import com.microsoft.azure.functions.annotation.FunctionName;
import com.microsoft.azure.functions.annotation.TimerTrigger;
import io.micronaut.azure.function.AzureFunction;

public class Function extends AzureFunction {

    @FunctionName("triggertestmicronaut")
    public void echo(@TimerTrigger(name = "req",
        dataType = "string",
        schedule = "0 */5 * * *") String req, // trigger once every 5m
                     ExecutionContext context) {
        if (context != null) {
            context.getLogger().info("Executing Function: " + getClass().getName() + " req: " + req);
        }
    }
}
```



Azure Gradle Plugin Configuration



The screenshot shows the Azure Monitor for Functions interface for a function named "triggertestmicronaut". The left sidebar includes links for Overview, Developer, Code + Test, Integration, Monitor (which is selected), and Function Keys. The main area displays "Invocation Details" with a table of logs. A specific log entry from 2020-09-29 at 13:40:00.043 is highlighted with a red box, showing the execution of a function with a complex JSON payload. The table has columns for Timestamp, Message, and Type.

Timestamp	Message	Type
2020-09-29 13:40:00.032	Executing 'Functions.triggertestmicronaut' (Reason='Timer fired at 2020-09-29T13:40:00.0315290+00:00', Id=18df3c25-78d2-4eee-9d95-a44283cadda4)	Information
2020-09-29 13:40:00.043	Executing Function: com.example.Function req: {"Schedule": {"AdjustForDST": true}, "ScheduleStatus": {"Last": "2020-09-29T13:35:00.0190683+00:00", "Next": "2020-09-29T13:40:00+00:00", "LastUpdated": "2020-09-29T13:35:00.0190683+00:00"}, "IsPastDue": false}	Information
2020-09-29 13:40:00.043	Function "triggertestmicronaut" (Id: 18df3c25-78d2-4eee-9d95-a44283cadda4) invoked by Java Worker	Information
2020-09-29 13:40:00.044	Executed 'Functions.triggertestmicronaut' (Succeeded, Id=18df3c25-78d2-4eee-9d95-a44283cadda4, Duration=12ms)	Information

Micronaut Azure Resources

- <https://micronaut-projects.github.io/micronaut-azure/snapshot/guide/index.html>
- <https://github.com/Azure/azure-functions-java-librarymicronaut.io/faq.html>
- <https://docs.microsoft.com/en-us/azure/azure-functions/functions-reference-java>

Micronaut Resources

- gitter.im/micronautfw
- docs.micronaut.io
- guides.micronaut.io
- micronaut.io/faq.html
- github.com/micronaut-projects/micronaut-core
- github.com/micronaut-projects/micronaut-examples
- objectcomputing.com/products/micronaut
- info@micronaut.io

Questions?



CONNECT WITH US



1+ (314) 579-0066



@objectcomputing



objectcomputing.com