# Azure Functions

**What is it for?** Processing data, integrating systems, working with the internet-of-things (IoT), and building simple APIs and microservices. Consider Functions for tasks like image or order processing, file maintenance, or for any tasks that you want to run on a schedule.

**Supported languages:** C#, F#, JavaScript, TypeScript, Java, PowerShell, Python, Bash, Batch (.cmd, .bat), PHP

Can be run on a server (App Service Plan) 🡪 but also can be run SERVERLESS (its called Consumption Plan [with this, it can scale out automatically if there’s too many triggers coming in parallel]) 🡪 this can be selected when creating it, under „Hosting Plan”

**Triggers and bindings** help avoid hardcoding access to other services.  
Function receives data in function parameters.  
Data gets sent by using the return value of the function.

**Bindings:** Ways to simplify coding for input/output data, a way of declaratively connecting another resource to the function (input/output or both type of bindings) Can have more than one

Except for HTTP and timer triggers, bindings are implemented as extension packages (nuget)

In a C# class library project, the bindings are defined as binding attributes on the function method. The *function.json* file is then auto-generated based on these attributes.

Append before Run():

[FunctionName("HttpTrigger")]

public static async Task<IActionResult> Run(

[HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,

[Queue("outqueue"),StorageAccount("AzureWebJobsStorage")] ICollector<string> msg, ILogger log)

// Add a message to the output collection.

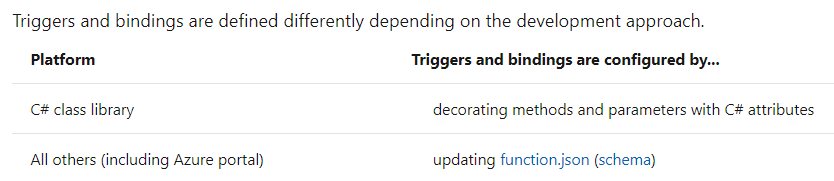
msg.Add(string.Format("Name passed to the function: {0}", name));

The msg parameter is an ICollector<T> type, which represents a collection of messages that are written to an output binding when the function completes.

In this case, the output is a storage queue named outqueue. The connection string for the Storage account is set by the StorageAccountAttribute. This attribute indicates the setting that contains the Storage account connection string and can be applied at the class, method, or parameter level.

**Triggers:** Ways to start the execution of code. Must have only one  
Triggers have associated data, which is often provided as the payload of the function.  
HTTPTrigger

* TimerTrigger
* CosmosDbTrigger: docs are added/updated in collections of a NoSql DB
* BlobTrigger: when added to blob (eg. Image resizing)
* QueueTrigger: respond to messages when they arrive to **Azure Storage queueu**
* EventGridTrigger: respond to events delivered to a subscription in **Azure Event Grid** (filtering, etc)
* EventHubTrigger: respond to events delivered to **Azure Event Hub** (great for IoT, app instrumentation, UX/workflow processing)
* ServiceBusQueueTrigger: connect to other Azure/**onPremise** services by listening to queues
* ServiceBusTopicTrigger: connect to other Azure/**onPremise** services by listening to topics



Cannot be triggered from table storage

**Can a Function App bind to a Notification Hub to send mobile notifications as an output binding? yes**

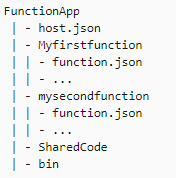
Functions uses a general-purpose account in Azure Storage to maintain state and other information about your functions.

az storage account create \  
--name <storage\_name> \  
--location westeurope \  
--resource-group myResourceGroup \  
--sku Standard\_LRS

az functionapp create \  
--resource-group myResourceGroup \  
--consumption-plan-location westeurope \

--name <APP\_NAME> \  
--storage-account <STORAGE\_NAME> \  
--runtime <language>  
  
Setting the ***consumption-plan-location*** parameter means that the function app is hosted in a Consumption hosting plan.

func new --name MyHttpTrigger --template "HttpTrigger"  
func start –build  
func azure functionapp publish <APP\_NAME>

All functions in a function app must be authored in the same language. In [previous versions](https://docs.microsoft.com/en-us/azure/azure-functions/functions-versions) of the Azure Functions runtime, this wasn't required.

Folder structure:

The [host.json](https://docs.microsoft.com/en-us/azure/azure-functions/functions-host-json) file contains runtime-specific configurations and is in the root folder of the function app. A *bin* folder contains packages and other library files that the function app requires.

When multiple triggering events occur faster than a single-threaded function runtime can process them, the runtime may invoke the function multiple times in parallel.

Each instance of the function app, whether the app runs on the Consumption hosting plan or a regular [App Service hosting plan](https://docs.microsoft.com/en-us/azure/app-service/overview-hosting-plans), might process concurrent function invocations in parallel using multiple threads. The maximum number of concurrent function invocations in each function app instance varies based on the type of trigger being used as well as the resources used by other functions within the function app.

Can be developed and debugged on local workstation, which is a big plus to increase developer productivity.

When dealing with synchronous request/response calls, that execute more complex logic, Azure function is preferred option over Azure Logic Apps.

Azure Functions is a serverless compute service: code being triggered by an event  
Azure Logic Apps provides serverless workflows: work flow triggered by an event

Both can create complex orchestrations.

* An orchestration is a collection of functions or steps
  + Actions in Azure Logic Apps
  + Durable Functions extensions in Azure Functions
    - Starter -> Orchestrator -> calls workers

