Contents

[***HTTP Trigger function*** 2](#_Toc129073558)

[***Blob Storage Trigger*** 2](#_Toc129073559)

[***Queue Storage Trigger*** 4](#_Toc129073560)

[***Service Bus Trigger Function*** 5](#_Toc129073561)

[***Service Bus Topic Trigger Function*** 6](#_Toc129073562)

[***Time Trigger*** 7](#_Toc129073563)

[***Event Trigger Function*** 8](#_Toc129073564)

[***Event Hub Function*** 10](#_Toc129073565)

[***Azure Cosmos DB Trigger Function*** 11](#_Toc129073566)

# ***HTTP Trigger function***

What is a http trigger function in azure portal

An HTTP trigger function in Azure Portal is a type of Azure Function that responds to HTTP requests. It's a serverless compute service that lets you create functions that are automatically triggered by HTTP requests.

When an HTTP request is received, the function is invoked and can perform some logic or operation based on the request. The function can then return a response back to the client that made the request.

HTTP trigger functions are commonly used for building APIs or webhooks that respond to events or requests from other applications. You can create an HTTP trigger function in Azure Portal using a variety of programming languages such as C#, Java, JavaScript, or Python.

Where can you use this function and how

You can use HTTP trigger functions in a variety of scenarios where you need to respond to HTTP requests, such as:

1. Building APIs: You can create HTTP trigger functions to build APIs that other applications can call to retrieve or update data.
2. Webhooks: You can use HTTP trigger functions as webhooks that are called by external services when a specific event occurs. For example, you can create a webhook that is triggered when a new file is uploaded to a storage account, and then perform some custom logic such as sending an email or updating a database.
3. Serverless websites: You can use HTTP trigger functions to build serverless websites that respond to HTTP requests. For example, you can create a function that serves an HTML page when a GET request is made to a specific URL.

To use HTTP trigger functions in Azure Portal, you'll first need to create a Function App. This is a container for your functions that provides the underlying infrastructure and resources needed to run your code. Once you've created a Function App, you can create an HTTP trigger function within it using the Azure Portal or other development tools such as Visual Studio or Visual Studio Code. You can then deploy and configure your function to respond to HTTP requests using Azure Portal or other deployment tools such as Azure CLI or Azure PowerShell.

# ***Blob Storage Trigger***

What is blob storage function in azure portal

Azure Blob Storage is a cloud-based object storage solution offered by Microsoft Azure that allows you to store and manage large amounts of unstructured data in the cloud. Blob Storage provides a simple and scalable way to store text and binary data, including documents, images, videos, and logs.

Azure Blob Storage consists of three types of blobs:

1. Block blobs: These are optimized for streaming and storing large amounts of unstructured data, such as text and binary data. They are commonly used for backup and restore, media and video streaming, and big data processing scenarios.
2. Append blobs: These are designed for scenarios where you need to append data to an existing blob, such as log files or data from IoT devices.
3. Page blobs: These are optimized for random read and write operations and are typically used for storing virtual hard disks (VHDs) and other large files that require frequent updates.

Azure Blob Storage provides features such as geo-replication, which automatically replicates your data across multiple regions for increased availability and durability. You can also use Azure Blob Storage with other Azure services, such as Azure Functions and Azure Stream Analytics, to create powerful cloud-based solutions.

In summary, Azure Blob Storage is a cloud-based storage solution that provides a simple, scalable, and cost-effective way to store and manage unstructured data in the cloud.

Where can you use this function and how

Azure Blob Storage can be used in a variety of scenarios where you need to store and manage unstructured data in the cloud. Some common scenarios include:

1. Backup and restore: You can use Blob Storage to store backups of your data, applications, and virtual machines. Blob Storage provides durability, reliability, and the ability to store large amounts of data at a low cost.
2. Media and video streaming: Blob Storage can be used to store media files such as videos, music, and images, which can be streamed to clients on-demand. This is useful for applications that require high-quality video and audio playback, such as online video platforms or video conferencing tools.
3. Big data processing: Blob Storage can be used as a data lake for storing large amounts of unstructured data such as logs, sensor data, and social media feeds. You can use services like Azure HDInsight and Azure Databricks to process and analyze this data.
4. IoT data storage: You can use Blob Storage to store data from IoT devices, such as telemetry data and sensor readings. This data can be processed and analyzed using services like Azure Stream Analytics and Azure IoT Hub.
5. Web application data storage: Blob Storage can be used to store static assets such as images, CSS files, and JavaScript files, which can be served to clients by a web application. This can improve the performance of your web application by offloading the static content to a dedicated storage service.

To use Azure Blob Storage, you first need to create a storage account in the Azure portal. Once you have a storage account, you can use a variety of tools and libraries to interact with Blob Storage, such as the Azure portal, Azure CLI, Azure PowerShell, and various programming languages such as .NET, Java, Python, and Node.js. You can also integrate Blob Storage with other Azure services to create powerful cloud-based solutions.

# ***Queue Storage Trigger***

What is Service Bus Queue function in azure portal

Azure Service Bus Queue is a messaging service in the Azure Portal that enables you to send and receive messages between different components or applications. It provides a reliable way to exchange messages across different systems or components, even when those systems or components are not available at the same time.

The basic concept of Service Bus Queue is that it enables one application or component to send a message to another application or component through a message queue. The receiving application can then retrieve the message from the queue when it is available, process it, and then acknowledge that the message has been received.

Service Bus Queue in Azure Portal provides several features, including:

1. Message Queues: A message queue is a buffer that holds messages until the receiving application is available to process them. Service Bus Queue ensures that messages are stored safely and reliably, so they are not lost if the receiving application is offline or unavailable.
2. Message Ordering: Service Bus Queue guarantees that messages are processed in the order they are received. This ensures that critical messages are processed first.
3. Message Filtering: Service Bus Queue enables you to filter messages based on specific criteria, so you only receive the messages that are relevant to your application.
4. Dead-Letter Queues: Service Bus Queue provides a Dead-Letter Queue (DLQ) that holds messages that cannot be processed. This enables you to investigate why messages could not be processed and take corrective actions.

Overall, Service Bus Queue in Azure Portal provides a reliable and scalable messaging solution that enables you to integrate different components or applications in your solution.

Where can you use this function and how

You can use Azure Service Bus Queue function in a variety of scenarios where you need to communicate and exchange messages between different components or applications in a reliable ***and scalable manner. Here are some examples of where you can use this function:***

1. Microservices Architecture: In a microservices architecture, each service is designed to perform a specific function and communicates with other services using messaging. Service Bus Queue can be used to exchange messages between microservices in a scalable and reliable manner.
2. Asynchronous Processing: Service Bus Queue can be used to perform asynchronous processing, where messages are processed in the background without affecting the main application or user interface. This can be useful for long-running tasks such as batch processing, data analysis, or report generation.
3. Workflow Orchestration: Service Bus Queue can be used to coordinate workflows between different components or applications. For example, you can use Service Bus Queue to send messages to trigger different steps in a workflow and ensure that the steps are executed in the correct order.
4. Event-Driven Architecture: Service Bus Queue can be used in an event-driven architecture to exchange messages between different components or applications in response to events. For example, you can use Service Bus Queue to send messages when a new customer is added to a database, triggering other components or applications to perform specific actions.

To use Azure Service Bus Queue function, you need to create a Service Bus namespace in the Azure

# ***Service Bus Trigger Function***

what is Service Bus Queue trigger function in azure portal

Service Bus Queue trigger function in Azure Portal is a serverless computing feature provided by Azure Functions. It enables developers to create an event-driven architecture for their applications by automatically invoking a function in response to a message arriving in a Service Bus Queue.

When a new message is added to a Service Bus Queue, the Service Bus Queue trigger function automatically executes the specified function. The message is passed as a parameter to the function, allowing developers to process the message and perform any required actions.

This feature is particularly useful in scenarios where there is a need for real-time processing of messages, such as processing incoming orders or messages from IoT devices.

To create a Service Bus Queue trigger function in Azure Portal, you can use the Azure Functions runtime, which is a serverless platform that allows you to develop, run, and scale your code without managing infrastructure. You can create a new Azure Function and select the Service Bus Queue trigger option as the trigger type. You then specify the connection string for the Service Bus Queue, the queue name, and the function to execute when a new message arrives.

When can you use this function and how

Service Bus Queue trigger function can be used when there is a need to process messages in an asynchronous and scalable way. This can be useful in a variety of scenarios, including:

1. Decoupled and scalable architecture: If you have a decoupled architecture where different parts of your application need to communicate with each other asynchronously, you can use Service Bus Queue trigger function to process messages that are sent between them.
2. Real-time processing: If you need to process messages in real-time as they arrive, such as processing orders or processing messages from IoT devices, you can use Service Bus Queue trigger function to process the messages as they are received.
3. Error handling: If you have a process that fails occasionally due to network connectivity issues or other reasons, you can use Service Bus Queue trigger function to handle failed messages and retry processing them later.

To use Service Bus Queue trigger function, you need to first create a Service Bus Queue and configure it to receive messages from your application. You can then create an Azure Function that is triggered by new messages arriving in the queue. The function can be written in a variety of programming languages, including C#, JavaScript, Python, and more.

Once the function is set up, any messages sent to the Service Bus Queue will automatically trigger the function, which can then process the message as required. This provides a scalable and efficient way to process messages in real-time and helps to decouple different parts of your application.

# ***Service Bus Topic Trigger Function***

What is service bus topic trigger function in azure portal

Azure Service Bus is a message broker that provides a way for different applications and services to communicate with each other using messages. It supports multiple messaging patterns, including publish/subscribe.

A Service Bus topic is a virtual entity that acts as a message broker between multiple publishers and subscribers. Publishers can send messages to the topic, and subscribers can receive those messages from the topic.

Azure Functions is a serverless compute service that allows you to run code in response to events and triggers. You can use Azure Functions to process messages from a Service Bus topic.

A Service Bus topic trigger function is an Azure Function that gets triggered when a new message is added to a Service Bus topic. This function can be written in different languages, such as C#, JavaScript, or Python.

When you create a Service Bus topic trigger function in the Azure portal, you need to specify the Service Bus connection string, the topic name, and the subscription name (if applicable). You can also configure other settings, such as the batch size and the maximum number of concurrent function invocations.

Once the function is deployed and running, it will automatically listen for new messages in the specified Service Bus topic. When a new message is added to the topic, the function will be triggered, and you can write code to process the message. This can include parsing the message content, performing some business logic, and then sending a response back to the topic or to another service.

Overall, using a Service Bus topic trigger function in Azure portal can provide a powerful way to build reactive and scalable applications that can respond to events and messages in real-time.

Where can you use this function and how

You can use a Service Bus topic trigger function in Azure portal in a variety of scenarios where you need to process messages in a scalable and reactive way. Some common use cases include:

1. Microservices communication: If you are using a microservices architecture, you can use Service Bus topics to enable communication between different services. Each service can publish messages to the topic, and other services can subscribe to those messages and process them as needed.
2. Asynchronous processing: If you have a process that takes a long time to complete, such as a batch job or data processing task, you can use a Service Bus topic trigger function to handle the processing asynchronously. You can send messages to the topic that represent different steps in the process, and the function can handle each message as it arrives, without blocking the main application.
3. Real-time data processing: If you need to process data in real-time, such as processing sensor data from IoT devices, you can use a Service Bus topic trigger function to handle the incoming data streams. The function can receive messages as they arrive and perform real-time analysis, triggering alerts or taking other actions as needed.

To use a Service Bus topic trigger function, you need to first create a Service Bus topic in Azure portal and configure it with the appropriate settings, such as the access policies and subscription rules. Then, you can create an Azure Function app and add a new function with the Service Bus trigger. You can write your code in the function to process the messages from the topic, using the appropriate programming language and libraries.

Once the function is deployed, it will automatically listen for new messages in the Service Bus topic and trigger the function when a new message arrives. You can scale the function up or down as needed, depending on the volume of messages and the processing requirements.

# ***Time Trigger***

What is timer trigger function in azure portal

Azure Functions is a serverless compute service that allows you to run code in response to events and triggers. One of the triggers available in Azure Functions is the Timer trigger.

A Timer trigger function is an Azure Function that runs on a schedule, based on a cron expression. The cron expression specifies the frequency and interval at which the function should be triggered. The Timer trigger can be used for a variety of tasks, such as data synchronization, backups, and maintenance tasks that need to run on a regular schedule.

When you create a Timer trigger function in Azure portal, you can specify the cron expression for the schedule, which is made up of six fields that represent different parts of the schedule, such as the minute, hour, and day of the week. You can also set other properties, such as the time zone and the function name.

Once the Timer trigger function is deployed and running, it will be triggered at the specified schedule, and you can write code to perform the desired task. For example, you can use the function to retrieve data from an external API and store it in a database, or to perform regular maintenance tasks on your application.

Overall, the Timer trigger function in Azure portal can provide a convenient and reliable way to schedule and automate routine tasks, without requiring you to manage the infrastructure or resources. It can help you reduce operational overhead and increase efficiency in your application.

Where can you use this function and how

You can use a Timer trigger function in Azure portal in a variety of scenarios where you need to perform regular or periodic tasks. Some common use cases include:

1. Data synchronization: If you have data stored in different systems or databases that need to be synchronized regularly, you can use a Timer trigger function to retrieve and update the data on a schedule. For example, you can use the function to fetch data from an external API and store it in a database, or to export data from a database to a file.
2. Scheduled backups: If you need to backup data or resources on a regular schedule, you can use a Timer trigger function to perform the backups automatically. For example, you can use the function to create regular backups of your database, file system, or virtual machines.
3. Maintenance tasks: If you have routine maintenance tasks that need to be performed on your application, such as clearing caches, purging old data, or restarting services, you can use a Timer trigger function to perform those tasks on a regular schedule.

To use a Timer trigger function, you need to first create an Azure Function app and add a new function with the Timer trigger. You can then write your code in the function to perform the desired task, using the appropriate programming language and libraries.

When you configure the Timer trigger, you can specify the cron expression for the schedule, which determines when the function should be triggered. The cron expression consists of six fields that specify the minute, hour, day of the month, month, day of the week, and time zone.

Once the Timer trigger function is deployed, it will be triggered automatically on the schedule specified by the cron expression. You can monitor the function's execution and logs in Azure portal, and you can scale the function up or down as needed, depending on the processing requirements.

# ***Event Trigger Function***

What is event grid function in azure portal

An Azure Event Grid is a messaging service that allows you to react to events happening across various Azure services and third-party applications. You can use Event Grid to build reactive, event-driven applications in a serverless manner.

An Event Grid function in Azure portal is an Azure Function that is triggered by events published to an Event Grid topic. The Event Grid topic can receive events from various sources, including Azure services such as Storage, Cosmos DB, and Resource Manager, as well as custom events from third-party applications.

When you create an Event Grid function in Azure portal, you can specify the Event Grid topic that the function should listen to and the subscription details. You can also define the input and output bindings for the function, which allow you to receive and send data to other Azure services.

Once the Event Grid function is deployed and running, it will be triggered automatically when an event is published to the specified Event Grid topic. You can write your code in the function to perform the desired task in response to the event. For example, you can use the function to send an email notification when a new file is added to a storage account, or to trigger an Azure Data Factory pipeline when a database record is updated.

Overall, Event Grid functions in Azure portal can provide a powerful and flexible way to build reactive, event-driven applications that can respond to changes and events in near-real-time. They can help you automate workflows, integrate services, and improve the overall efficiency and responsiveness of your application.

Where can you use this function and how

You can use an Event Grid function in Azure portal in a variety of scenarios where you need to react to events happening across various Azure services and third-party applications. Some common use cases include:

1. Workflow automation: If you have a series of tasks or workflows that need to be automated based on specific events, you can use an Event Grid function to trigger those workflows automatically. For example, you can use the function to trigger a data pipeline in response to a new file being uploaded to Azure Storage, or to send a notification when a database record is updated.
2. Service integration: If you have multiple Azure services or third-party applications that need to be integrated together, you can use an Event Grid function to receive and process events from those services. For example, you can use the function to receive events from Azure Resource Manager when resources are created or deleted, and then trigger a series of actions in response.
3. Real-time monitoring: If you need to monitor changes or events happening in real-time across multiple services, you can use an Event Grid function to receive and process those events in real-time. For example, you can use the function to monitor changes in Azure AD, and trigger an alert or action when a user is added or removed from a group.

To use an Event Grid function, you need to first create an Azure Function app and add a new function with the Event Grid trigger. You can then write your code in the function to perform the desired task, using the appropriate programming language and libraries.

When you configure the Event Grid trigger, you can specify the Event Grid topic that the function should listen to and the subscription details. You can also define the input and output bindings for the function, which allow you to receive and send data to other Azure services.

Once the Event Grid function is deployed, it will be triggered automatically when an event is published to the specified Event Grid topic. You can monitor the function's execution and logs in Azure portal, and you can scale the function up or down as needed, depending on the processing requirements.

# ***Event Hub Function***

What is event hub trigger function in azure portal

An Event Hub trigger function in Azure portal is an Azure Function that is triggered by incoming messages or events in an Azure Event Hub.

Azure Event Hub is a streaming platform and event ingestion service that can process and store millions of events per second. You can use Event Hub to ingest, buffer, and process large-scale streaming data from various sources, such as devices, applications, and services.

When you create an Event Hub trigger function in Azure portal, you can specify the Event Hub namespace and name, the consumer group, and the connection string for the Event Hub. You can also define the input and output bindings for the function, which allow you to receive and send data to other Azure services.

Once the Event Hub trigger function is deployed and running, it will be triggered automatically when a new event or message is received in the specified Event Hub. You can write your code in the function to perform the desired task in response to the event. For example, you can use the function to process and analyze sensor data from IoT devices, or to aggregate and store log data from web applications.

Overall, Event Hub trigger functions in Azure portal can provide a powerful and flexible way to process and analyze large-scale streaming data in real-time. They can help you extract insights, detect anomalies, and trigger alerts or actions based on specific patterns or conditions in the data.

Where can you use this function and how

You can use an Event Hub trigger function in Azure portal in a variety of scenarios where you need to process and analyze large-scale streaming data from various sources. Some common use cases include:

1. Internet of Things (IoT) data processing: If you have a large number of IoT devices that are generating streaming data, you can use an Event Hub trigger function to process and analyze that data in real-time. For example, you can use the function to calculate average sensor values or to detect anomalies in sensor data.
2. Log data processing: If you have web applications or services that are generating large amounts of log data, you can use an Event Hub trigger function to process and analyze that data in real-time. For example, you can use the function to detect errors or security breaches in log data, or to generate alerts based on specific patterns or conditions.
3. Real-time analytics: If you need to perform real-time analytics on large-scale streaming data, you can use an Event Hub trigger function to process and analyze that data in real-time. For example, you can use the function to calculate rolling averages or to detect trends in streaming data.

To use an Event Hub trigger function, you need to first create an Azure Function app and add a new function with the Event Hub trigger. You can then write your code in the function to perform the desired task, using the appropriate programming language and libraries.

When you configure the Event Hub trigger, you can specify the Event Hub namespace and name, the consumer group, and the connection string for the Event Hub. You can also define the input and output bindings for the function, which allow you to receive and send data to other Azure services.

Once the Event Hub trigger function is deployed, it will be triggered automatically when a new event or message is received in the specified Event Hub. You can monitor the function's execution and logs in Azure portal, and you can scale the function up or down as needed, depending on the processing requirements.

# ***Azure Cosmos DB Trigger Function***

What is Azure Cosmos DB trigger function in azure portal

An Azure Cosmos DB trigger function in Azure portal is an Azure Function that is triggered by changes or updates to data in an Azure Cosmos DB database.

Azure Cosmos DB is a fully-managed NoSQL database service that provides global scale and low-latency data access. You can use Cosmos DB to store and query various types of data, including documents, key-value pairs, and graph data.

When you create a Cosmos DB trigger function in Azure portal, you can specify the Cosmos DB database and container that you want to monitor for changes, as well as the type of trigger that you want to use. There are three types of triggers available for Cosmos DB trigger functions:

1. Change feed trigger: This type of trigger is fired when a new document is added, updated, or deleted in the specified Cosmos DB container. The function can then process the document and perform the desired task, such as sending a notification or updating another database.
2. Lease-based trigger: This type of trigger is used for distributed scenarios where multiple instances of the function may be running concurrently. The trigger uses a lease mechanism to ensure that only one instance of the function is processing a given partition of the Cosmos DB container at a time.
3. Time-based trigger: This type of trigger is fired at regular intervals, regardless of whether there are any changes to the Cosmos DB container. This can be useful for scenarios where you need to perform periodic cleanup tasks or aggregate data over a specific time period.

Once the Cosmos DB trigger function is deployed and running, it will be triggered automatically when a new document or change is detected in the specified Cosmos DB container. You can write your code in the function to perform the desired task in response to the change. For example, you can use the function to update

Where can you use this azure cosmos db function and how

Azure Cosmos DB is a distributed database service that supports several APIs for data access, including SQL, MongoDB, Cassandra, and Azure Table storage. The functions in Azure Cosmos DB are designed to work with these APIs and provide developers with a way to execute serverless code within the database service.

Some examples of where you can use Azure Cosmos DB functions include:

1. Data processing and transformation: You can use Azure Cosmos DB functions to process and transform data within the database service. For example, you can create a function that takes in data from one collection and transforms it before inserting it into another collection.
2. Real-time updates: Azure Cosmos DB functions can be used to trigger real-time updates in response to database events. For example, you can create a function that sends a notification when a new document is inserted into a collection.
3. Custom business logic: Azure Cosmos DB functions can be used to implement custom business logic within the database service. For example, you can create a function that enforces specific business rules, such as ensuring that a new document meets certain criteria before it can be inserted into a collection.

To use Azure Cosmos DB functions, you can write your function code using one of the supported programming languages, such as JavaScript, C#, or Java. You can then deploy your function code to Azure and use the Azure portal or the Azure Functions runtime to manage and run your functions. You can also use Azure Cosmos DB triggers to automatically invoke your functions in response to specific database events.