When we using queue sender and receiver will not communicate directly. Instead of exchanging the messages via queue

Sender sends the message to queue and receiver pulls the messages from queue then processing it

The sender does not have to wait for a reply from the receiver in order to continue to process and send further messages

Queues offer *First In, First out* (FIFO) message delivery to one or more competing consumers

That is, receivers typically receive and process messages in the order in which they were added to the queue

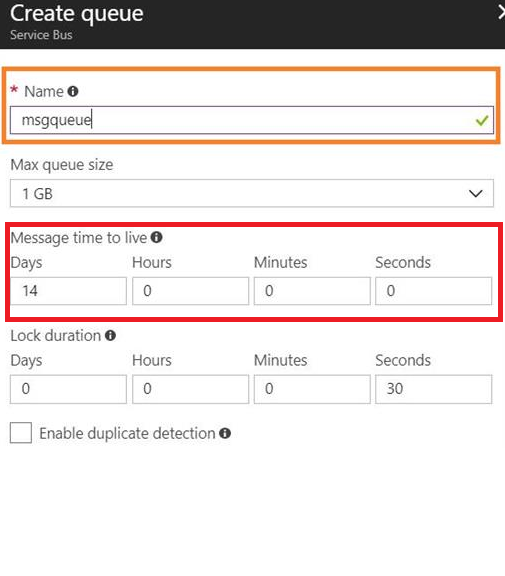
The producers (senders) and consumers (receivers) do not have to be sending and receiving messages at the same time, because messages are stored durably in the queue

**QUEUE**

**How long messages will be in queue?**

Default it will be in 14 days.

If you want to change, can configure in **Message in live**

****

**Real Time Scenario**

<https://www.c-sharpcorner.com/article/azure-service-bus-queue-with-real-world-scenario/>

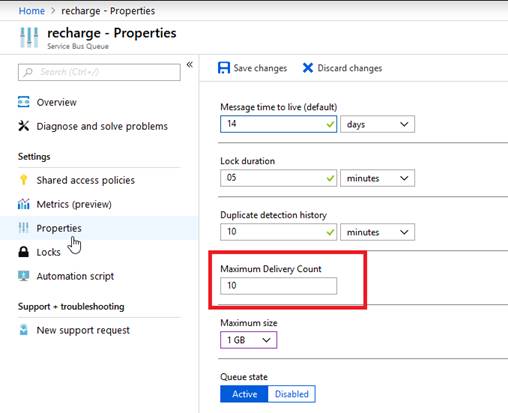
Bill Payments Company wants to build an application, which will do the mobile recharge for end users. The application should support all the operators, which are available in the market.

The company is looking for something like where all the requests are parked when they come and process one by one.

Here in order to meet the business requirement, we will use the queue mechanism where all the requests will be added in the queue and another side queue listener will process the message by reading from the queue.

The benefit of using queue is sender and receiver should not be available at the same time while sending the message, as messages are stored in a queue

You can configure the reprocess count from properties by setting Maximum Delivery Count. Previously it was read 10 times because, by default, Maximum Delivery Count is set to 10.



Once the 10-time process is over, it is no longer available in the queue, and it will be added in the Dead-letter queue

**Difference**

**Storage queue**

When I say a lot, imagine queues that can reach not 1 GB, not 1TB and not 10TB. Using Azure Storage Queues, we can have a queue that has 200 TB

We have the ability to store large amounts of data in queues, without thinking about the size of the queue

If the size of queue can be very large, the maximum TTL (Time to Leave) of a message is 7 days. This means that a message needs to be consumed in 7 days or renewed; otherwise, the message will be removed

We do not have support for duplicate detection or transaction support.

An interesting feature of Azure Storage Queues is the logging capabilities. Users have the ability to activate the loggings mechanism and track all the actions that are happening on the queue

Azure Storage Queues are great when you need a messaging system that is able to track all the actions that are happening on it. It can be a good solution for use cases when you know that the size of the queue will be bigger than 80-100 GB

**Service bus queue**

The size of a queue is limited to 80 GB

The size of a message can be 256 KB, larger in comparison with Azure Storage Queues

There is support for dead lettering, which allows us to move automatically a message in a secondary queue, if the message expires or clients cannot consume a specific message

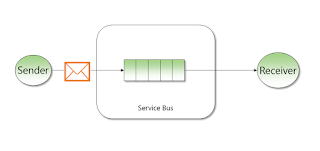
There is full support for transactions, handling a specific number of messages in the same transaction

An interesting feature of Azure Service Bus Queues is duplicate detection. Once it is activated, we can detect duplicate messages. The moment we want to add a message that already exists in the system, the message will not be added

We can peek at a message from a queue and make it unavailable for the rest of the clients until we confirm that we consumed it with success or abort the action

Bus Queues are great when we need duplicate detection, transaction support or to store messages for an unlimited period of time

Each message will be receive only by one receiver



Real-time example: mobile recharge based on network, offer will send messages to customer

<https://www.c-sharpcorner.com/article/azure-service-bus-queue-with-real-world-scenario/>

**TOPICS**

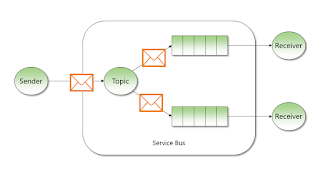
Azure Service Bus Queues that allow us to deliver one-to-one, Azure Service Bus Topics allow us to deliver messages one-to-many

This means that we can deliver the same message to multiple clients that are called subscribers

Each topic that is used to send messages can have maximum 2000 subscribers. This means that the same message can be received by 2000 subscribers

An interesting feature is the filter support. We can attach a filter to each subscription. That filter will allow only the messages that respect the filter rule to reach that specific subscription. In this way, each subscription can listen to specific messages

This type of queue will be used when a **single** message has to be received by multiple receivers



Real-time example: sample recharge application, company wants to give some offer on regular customer

So now will send same format message much number of customers

<https://www.c-sharpcorner.com/article/azure-service-bus-topic-and-subscription-pub-sub/>