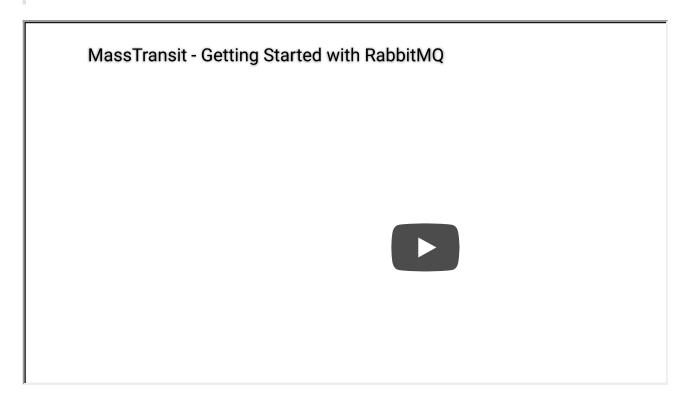


# RabbitMQ

This tutorial will get you from zero to up and running with RabbitMQ and MassTransit.



• The source for this sample is available on GitHub □\_.

## **Prerequisites**

#### **NOTE**

The following instructions assume you are starting from a completed In-Memory Quick Start

This example requires the following:

- a functioning installation of the dotnet runtime and sdk (at least 6.0)
- a functioning installation of docker with docker compose support



# Get RabbitMQ up and running

For this quick start we recommend running the preconfigured Docker image maintained by the MassTransit team . It includes the delayed exchange plug-in, as well as the Management interface enabled.

```
$ docker run -p 15672:15672 -p 5672:5672 masstransit/rabbitmq
```

If you are running on an ARM platform

```
$ docker run --platform linux/arm64 -p 15672:15672 -p 5672:5672 masstransit/rabbitmq
```

Once its up and running you can **log into** the broker using guest, guest. You can see message rates, routings and active consumers using this interface.

# Change the Transport to RabbitMQ

Add the MassTransit.RabbitMQ package to the project.

```
$ dotnet add package MassTransit.RabbitMQ
```

#### **Edit Program.cs**

Change UsingInMemory to UsingRabbitMq



```
{
    cfg.Host("localhost", "/", h => {
        h.Username("guest");
        h.Password("guest");
    });
    cfg.ConfigureEndpoints(context);
    });
};
services.AddHostedService<Worker>();
});
```

localhost is where the docker image is running. We are inferring the default port of 5672 and are using \ as the virtual host \( \text{localhost} \) guest and guest are the default username and password to talk to the cluster and management dashboard \( \text{localhost} \).

### Run the project

```
$ dotnet run
```

The output should have changed to show the message consumer generating the output (again, press Control+C to exit). Notice that the bus address now starts with <code>rabbitmq</code> .



NECCEIVED TEXT. THE CHILE IS STATICAL IZ.II.IO THE OS.OC

At this point the service is connecting to RabbbitMQ on localhost and publishing messages which are received by the consumer.



Help us by improving this page! ☐ Last Updated: 1/17/2023, 11:25:03 AM

Configuration  $\rightarrow$