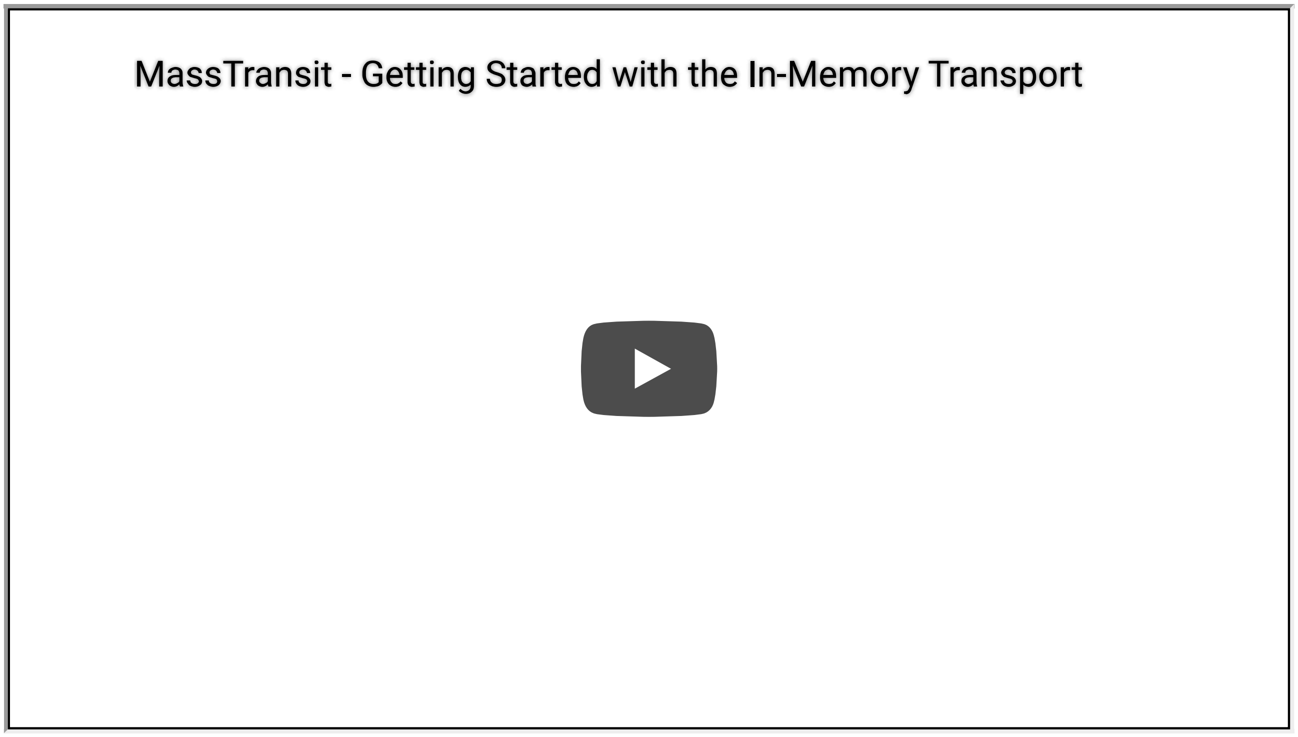
In Memory

> This tutorial will get you from zero to up and running with **In Memory** and MassTransit.



MassTransit - Getting Started with the In-Memory Transport

Prerequisites

This example requires the following:

- a functioning installation of the dotnet runtime and sdk (at least 6.0)

Install MassTransit Templates

MassTransit includes project and item **templates** simplifying the creation of new projects. Install the templates by executing `dotnet new -i MassTransit.Templates` at the console. A video introducing the templates is available on **YouTube** .

```
dotnet new --install MassTransit.Templates
```

Initial Project Creation

Create the worker project

To create a service using MassTransit, create a worker via the Command Prompt.

```sh
$ dotnet new mtworker -n GettingStarted
$ cd GettingStarted
$ dotnet new mtconsumer
```

Overview of the code

When you open the project you will see that you have 3 class files.

- `Program.cs` is the standard entry point and here we configure the host builder.
- `Consumers/GettingStartedConsumer.cs` is the MassTransit **Consumer**
- `Contracts/GettingStarted.cs` is an example message

Add A BackgroundService

In the root of the project add `Worker.cs`

```
namespace GettingStarted;

using System;
using System.Threading;
using System.Threading.Tasks;
using Contracts;
using MassTransit;
using Microsoft.Extensions.Hosting;

public class Worker : BackgroundService
{
    readonly IBus _bus;
```

```
        _bus = bus;
    }

    protected override async Task ExecuteAsync(CancellationToken stoppingToken)
    {
        while (!stoppingToken.IsCancellationRequested)
        {
            await _bus.Publish(new GettingStarted { Value = $"The time is {DateTimeOffs

            await Task.Delay(1000, stoppingToken);
        }
    }
}
```

Register Worker

In `Program.cs` at the bottom of the `ConfigureServices` method add

```
services.AddHostedService<Worker>();
```

Update your Consumer

In your `Consumers` folder, edit the `GettingStartedConsumer` with a logging statement that looks like this.

```
namespace GettingStarted.Consumers;

using System.Threading.Tasks;
using Contracts;
using MassTransit;
using Microsoft.Extensions.Logging;

public class GettingStartedConsumer :
    IConsumer<GettingStarted>
{
    readonly ILogger<GettingStartedConsumer> _logger;
    {
        _logger = logger;
    }

    public Task Consume(ConsumeContext<GettingStarted> context)
    {
        _logger.LogInformation("Received Text: {Text}", context.Message.Value);
        return Task.CompletedTask;
    }
}
```

Run the project

```sh
$ dotnet run
```

The output should have changed to show the message consumer generating the output (again, press Control+C to exit).

```
Building...
info: MassTransit[0]
      Configured endpoint Message, Consumer: GettingStarted.MessageConsumer
info: MassTransit[0]
      Bus started: loopback://localhost/
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /Users/chris/Garbage/start/GettingStarted
info: GettingStarted.MessageConsumer[0]
      Received Text: The time is 3/24/2021 12:02:01 PM -05:00
info: GettingStarted.MessageConsumer[0]
      Received Text: The time is 3/24/2021 12:02:02 PM -05:00
```

Last Updated: 4/11/2025, 11:23:05 AM

Configuration →