

# Masstransit Essentials

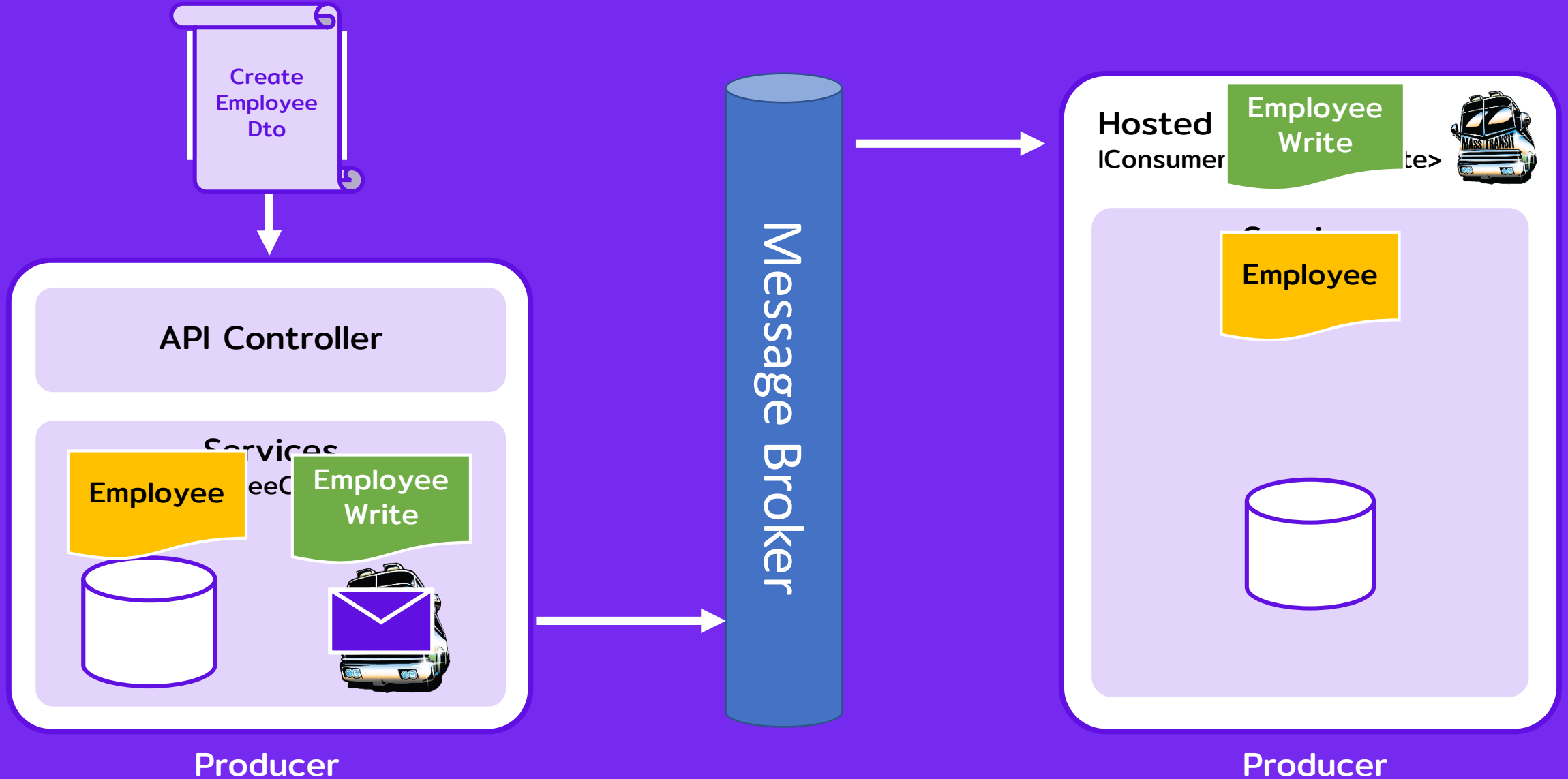


# MassTransit คืออะไร?

- เป็น NuGet Package ที่ใช้สำหรับการส่งข้อมูลผ่าน Message Broker สำหรับ C#
- รองรับการทำงาน Message Broker หลายตัว
- ทำให้เราสามารถทำงานร่วมกับ RabbitMQ ได้ง่ายขึ้น



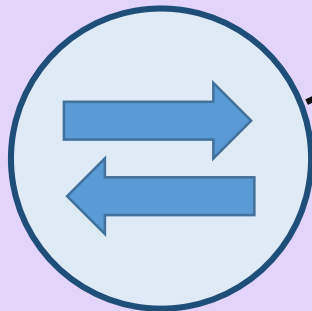
# การทำงานของ MassTransit



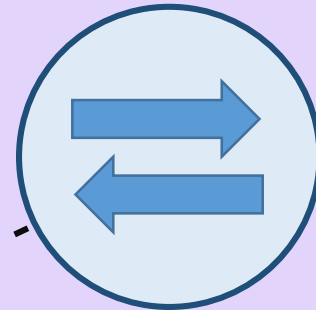
# Masstransit zu RabbitMQ



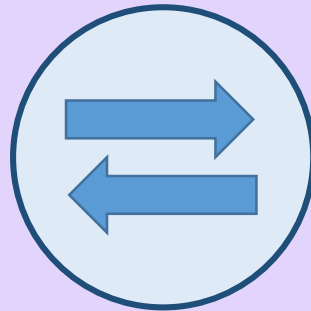
EXCHANGE



`IPT.EmployeeAPI.Contracts:  
EmployeeWrite`

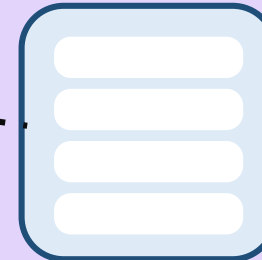


`auth-employee-write`



`auth-employee-delete`

QUEUES



`auth-employee-write`



`auth-employee-delete`

Consumer



# Contracts

Contract คือ DTO กลาง ที่ใช้ส่งข้อมูลระหว่าง Services ของ Message Broker เวลาที่เราส่ง เราจะต้องใช้ DTO เดียวกัน เพื่อที่จะคุยหากันรู้เรื่อง

Contract อาจจะมีข้อมูลทุก Fields แต่ Service สามารถเลือกเอาไปใช้เฉพาะบาง Fields ได้

CONTRACTS

# MasstransitExample.API.Contracts

- ให้ทำการเพิ่ม ไฟล์ ExampleEvent.cs

```
using System;

namespace MasstransitExample.API.Contracts
{
    public class ExampleEvent
    {
        public int ExampleID { get; set; }

        public String ExampleString { get; set; }
    }
}
```

**ASP.NET CORE API**



# Standard Setup

# Nuget Package

```
dotnet add package MassTransit.AspNetCore --version 7.2.2
```

```
dotnet add package MassTransit.RabbitMQ --version 7.2.2
```



**MassTransit.AspNetCore** by Chris Patterson

7.2.2

MassTransit hosting for ASP.NET Core; MassTransit is a message-based distributed application framework for .NET <http://masstransit-project.com>



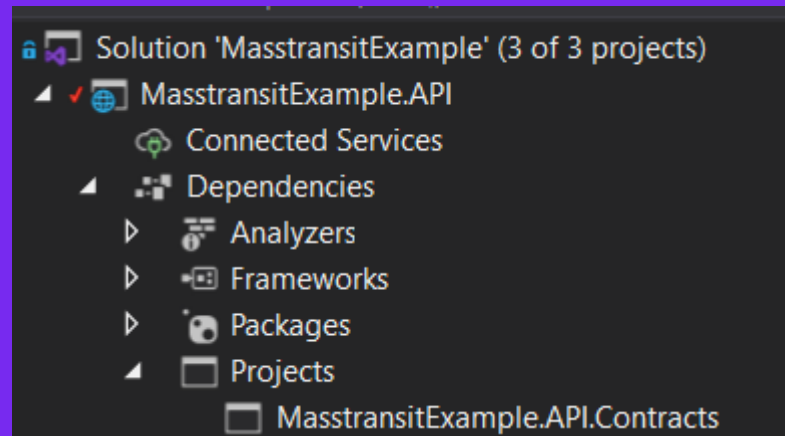
**MassTransit.RabbitMQ** by Chris Patterson

7.2.2

MassTransit RabbitMQ transport support; MassTransit is a message-based distributed application framework for .NET <http://masstransit-project.com>

# Contract Project References

```
dotnet add ..\MasstransitExample.API.Contracts\MasstransitExample.API.Contracts.csproj
```



# Startup.cs

```
using GreenPipes;  
using MassTransit;  
using MassTransit.Definition;
```

# Startup.cs

```
// เพิ่ม MassTransit ใน Service Pipeline
services.AddMassTransit(configure =>
{
    // เพิ่ม Config ของ RabbitMQ
    configure.UsingRabbitMq((context, configurator) =>
    {
        // Login เข้า RabbitMQ
        configurator.Host("192.168.200.50", h => {
            h.Username("rabbitmq");
            h.Password("Abcd1234#");
        });

        // ตั้ง Endpoint ของ RabbitMQ
        configurator.ConfigureEndpoints(context, new KebabCaseEndpointNameFormatter("MasstransitExample.API", false));

        // ตั้งการซ้ำ หากไม่สามารถรับค่าจาก RabbitMQ ได้สำเร็จ จะทำการซ้ำ
        configurator.UseMessageRetry(retryConfigurator =>
        {
            // ทำครั้ง 3 ครั้ง ทุก 5 นาที ก่อนเลิกทำ
            retryConfigurator.Interval(3, TimeSpan.FromSeconds(5));
        });
    });
});

// เพิ่ม Hosted Service ของ MassTransit เพื่อคุยกับ RabbitMQ
services.AddMassTransitHostedService();
```

# Publisher Setup

# ExampleController.cs

```
using MassTransit;  
using MasstransitExample.API.Contracts;  
using Microsoft.AspNetCore.Mvc;  
using System.Threading.Tasks;
```

# ExampleController.cs

```
namespace MasstransitExample.API.Controllers
{
    [ApiController]
    [Route("Example")]
    public class ExampleController : ControllerBase
    {
        IPublishEndpoint _publishEndpoint { get; set; }
        public ExampleController(IPublishEndpoint publishEndpoint)
        {
            _publishEndpoint = publishEndpoint;
        }
        [HttpGet("Publish")]
        public async Task<IActionResult> Publish()
        {
            var message = new ExampleEvent() {
                ExampleID = 6234,
                ExampleString = "WANATAT C"
            };

            await _publishEndpoint.Publish(message);

            return Ok("Message Published");
        }
    }
}
```



# Consumer Setup

# Startup.cs

```
// ➊ MassTransit ➋ Service Pipeline
services.AddMassTransit(configure =>
{
    // ➌ Config ➍ RabbitMQ
    configure.UsingRabbitMq((context, configurator) =>
    {
        // ➎ Consumer
        configurator.AddConsumer<ExampleConsumer>();
        .
        .
    });
});
```

# Consumers/ExampleConsumer.cs

```
using MassTransit;
using MasstransitExample.API.Contracts;
using System;
using System.Threading.Tasks;

namespace MasstransitExample.API.Consumers
{
    public class ExampleConsumer : IConsumer<ExampleEvent>
    {
        public Task Consume(ConsumeContext<ExampleEvent> context)
        {
            var message = context.Message;
            Console.WriteLine($"ExampleID : {message.ExampleID}");
            Console.WriteLine($"ExampleString : {message.ExampleString}");

            return Task.CompletedTask;
        }
    }
}
```

# .NET FRAMEWORK MVC

# Contract

# Multi-targets SDK

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <TargetFrameworks>netcoreapp3.1; net461</TargetFrameworks>
  </PropertyGroup>

</Project>
```

# MVC Project

# Nuget Package

```
dotnet add package MassTransit --version 7.2.2
```

```
dotnet add package MassTransit.RabbitMQ --version 7.2.2
```



**MassTransit** ✓ by Chris Patterson, **19.5M** downloads

7.2.2

MassTransit is a message-based distributed application framework for .NET <http://masstransit-project.com>



**MassTransit.RabbitMQ** ✓ by Chris Patterson, **12.8M** downloads

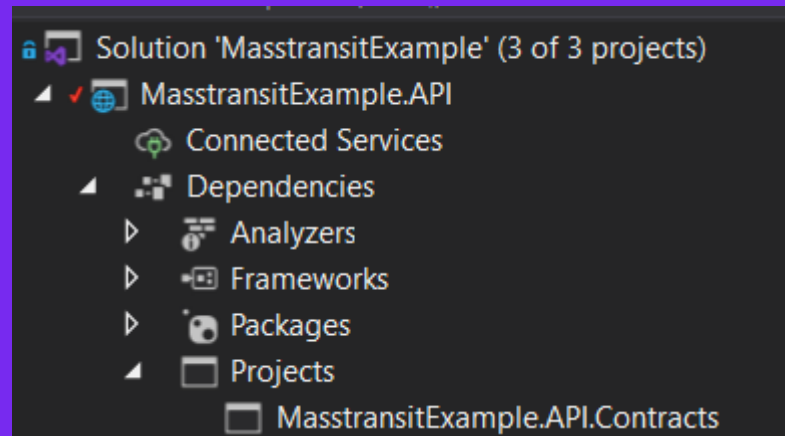
7.2.2

MassTransit RabbitMQ transport support; MassTransit is a message-based distributed application framework for .NET <http://masstransit-project.com>



# Contract Project References

```
dotnet add ..\MasstransitExample.API.Contracts\MasstransitExample.API.Contracts.csproj
```



# Global.asax

```
using MassTransit;
using System;
using System.Web.Mvc;
using System.Web.Routing;

namespace MasstransitExample.MVC
{
    public class MvcApplication : System.Web.HttpApplication
    {
        static IBusControl _bus;
        static BusHandle _busHandle;

        public static IBus Bus
        {
            get { return _bus; }
        }
    }
}
```

```
protected void Application_Start()
{
    AreaRegistration.RegisterAllAreas();
    RouteConfig.RegisterRoutes(RouteTable.Routes);

    _bus = MassTransit.Bus.Factory.CreateUsingRabbitMq(configurator => {

        // Login រើ RabbitMQ
        configurator.Host("192.168.200.50", h => {
            h.Username("rabbitmq");
            h.Password("Abcd1234#");
        });
    });

    _busHandle = MassTransit.Util.TaskUtil.Await<BusHandle>(() => _bus.StartAsync());
}

protected void Application_End()
{
    if (_busHandle != null)
        _busHandle.Stop(TimeSpan.FromSeconds(30));
}
}
```

# MVC View

# Index.cshtml

```
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Index</h2>

@using (Html.BeginForm("Example", "Home", FormMethod.Post))
{
    <div>Example ID : <input type="number" /></div>
    <div>Example String : <input type="text" /></div>
    <button type="submit">Post</button>
}
```

# Example.cshtml

```
@{  
    ViewBag.Title = "Example";  
    Layout = "~/Views/Shared/_Layout.cshtml";  
  
}  
  
<h2>Events Published</h2>
```

# MVC Controller

# HomeController.cs

```
using System.Threading.Tasks;
using System.Web.Mvc;
using MasstransitExample.API.Contracts;

namespace MasstransitExample.MVC.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        // POST: Example
        [HttpPost]
        public async Task<ActionResult> Example(ExampleEvent message)
        {
            await MvcApplication.Bus.Publish(message);

            return View();
        }
    }
}
```



# CHECK LISTS

# Contract Checklist

- ☐ พิจารณา Contract ของ Events อย่างครอบคลุม คิดเพื่อการใช้งานในอนาคต
- ☐ Contract จะต้องเป็น Standard Library แยกจาก Project
- ☐ เพื่อ Backward Compatible ให้ทำการแก้ .csproj ให้เป็น Multi-target SDK ย้อนไปถึง .net Framework 4.6.1
- ☐ หากเป็นการย้ายข้อมูล CRUD ให้ใช้ Property ทั้งหมด แบบเดียวกับ Model
- ☐ หากเป็นไปได้พิจารณาการแยก Event ย่อยๆ เพื่อลด Load ของ Queue

# Publisher Checklist

- ☐ ติดตั้ง Nuget Package
- ☐ แก้ Startup.cs ให้ใช้ MassTransit
- ☐ ใช้ IPublishEndpoint.Publish()
- ☐ ให้ทำการ Validate ข้อมูลทุกครั้งก่อน Publish
- ☐ เมื่อทำงานกับ Database ให้ทำการ Commit ข้อมูลก่อนทำการ Publish Message

# Consumer Checklist

- ☐ ติดตั้ง Nuget Package
- ☐ แก้ Startup.cs ให้ใช้ MassTransit
- ☐ ใช้ IConsumer ใน Consumers
- ☐ เพิ่ม Consumers ใน Startup.cs
- ☐ กรณีกาทำงานร่วมกับ Database  
ให้ทำการเช็คข้อมูลก่อน CUD ลง Database
- ☐ เพื่อให้ Database ไม่เก็บข้อมูลมากกว่าที่จำเป็น  
พิจารณาใช้บาง Property จาก Contract

# .net Framework MVC Checklist

- ☐ ติดตั้ง Nuget Package
- ☐ แก้ Global.asax ให้ใช้ MassTransit
- ☐ ใช้ MvcApplication.Bus.Publish()
- ☐ ให้ทำการ Validate ข้อมูลทุกครั้งก่อน Publish
- ☐ เมื่อทำงานกับ Database ให้ทำการ Commit ข้อมูลก่อนทำการ Publish Message