

## Asynchronous communication

### (Demo prep)

- Return Antidote to price 7
- Remove CatalogItemUpdated exchange
- Clear all queues
- Remove Play.Common package from %userprofile%/nuget/packages

### Refactoring into Common

Let's move the code that configures MassTransit and RabbitMQ into our Common library so that things get easier and cleaner for Catalog and any future microservices.

1. Copy RabbitMQSettings to Common
2. Fix RabbitMQSettings namespace to **Play.Common.Settings**
3. dotnet add package MassTransit.AspNetCore
4. dotnet add package MassTransit.RabbitMQ
5. Create a MassTransit directory
6. Add Extensions.cs
7. **Collapse the navigation pane**
8. Paste AddMasstransit code:

```
namespace Play.Common.MassTransit
{
    public static class Extensions
    {
        public static IServiceCollection AddMassTransitWithRabbitMq(this IServiceCollection services)
        {
            services.AddMassTransit(configure =>
            {
                configure.AddConsumers(Assembly.GetEntryAssembly());

                configure.UsingRabbitMq((context, configurator) =>
                {
                    var configuration = context.GetService<IConfiguration>();
                    var serviceSettings = configuration.GetSection(nameof(ServiceSettings)).Get<ServiceSettings>();
                    var rabbitMQSettings =
configuration.GetSection(nameof(RabbitMQSettings)).Get<RabbitMQSettings>();
```

```

        configurator.Host(rabbitMQSettings.Host);
        configurator.ConfigureEndpoints(context, new
KebabCaseEndpointNameFormatter(serviceSettings.ServiceName, false));
    });
});

services.AddMassTransitHostedService();

return services;
}
}
}

```

9. dotnet pack -p:PackageVersion=1.0.1 -o ../../packages\

10. Back to Catalog

11. Remove MassTransit references from csproj

12. Bump Play.Common to 1.0.1

13. Delete Settings dir

14. Update Startup → ConfigureServices:

```

public void ConfigureServices(IServiceCollection services)
{
    serviceSettings = Configuration.GetSection(nameof(ServiceSettings)).Get<ServiceSettings>();

    services.AddMongo()
        .AddMongoRepository<Item>("items")
        .AddMassTransitWithRabbitMq();

    ...
}

```

15. **dotnet run** in Terminal

16. Try an update to an item in Postman

17. Package the **Catalog.Contracts** project:

18. dotnet pack -o ../../packages\

In the next lesson we will update our Inventory service so that it can consume the messages published by Catalog.