

Allowing services to customize the retry configuration

Script start

As you may recall, our current `AddMassTransitWithRabbitMq` method is configured so that any time a message can't be consumed, MassTransit will retry consuming the message a few times.

(SHOW THE LINE)

However, you will soon need write a few MassTransit consumers where you might need to throw exceptions that should not be retried, like the exception that we will throw when trying to debit more gil than what the user has available.

So, what we want to do here is add a new extension method that will allow our services to specify how exactly retries should be configured, including what to do with some specific exceptions.

In Common repo

1. Open `MassTransit\Extensions.cs`
2. Extract `configure.UsingRabbitMq` into a new method:

```
public static void UsingPlayEconomyRabbitMQ(
    this IServiceCollectionBusConfigurator configure,
    Action<IRetryConfigurator> configureRetries = null)
{
    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));

        if (configureRetries == null)
        {
            configureRetries = (retryConfigurator) => retryConfigurator.Interval(3, TimeSpan.FromSeconds(5));
        }

        configurator.UseMessageRetry(configureRetries);
    });
}
```

3. Fix AddMassTransitWithRabbitMQ:

```
public static IServiceCollection AddMassTransitWithRabbitMq(
    this IServiceCollection services,
    Action<IRetryConfigurator> configureRetries = null)
{
    services.AddMassTransit(configure =>
    {
        configure.AddConsumers(Assembly.GetEntryAssembly());
        configure.UsingPlayEconomyRabbitMQ(configureRetries);
    });

    services.AddMassTransitHostedService();

    return services;
}
```

4. Create a new Play.Common NuGet package version:

```
dotnet pack -p:PackageVersion=1.0.4 -o ../../..\packages
```

In the next lesson we will update our Inventory microservice so that it can grant items via asynchronous messages.