Как да направим plugin за NetGenQueueService.

1. Добавя се нова C# Class Library в папка вътре в NetGenQueueService.
2. В Properties / Build / Output path се задава output папката на NetGenQueueService за избраната конфигурация.
3. Библиотеки, които са важни и вероятно трябва да бъдат добавени.
   1. AppConfigLibrary – чрез нея могат да се създават custom config sections в app.config файла (т.е. конфигурацията на plugin-а се улеснява).
   2. През nuget се добавя log4net (команда Install-Package log4net).
   3. Ако е необходим достъп до база данни най-добре използвайте DbEngine (това не е наложително, но помага за прилагането на унифициран подход за достъп до SQL Server).
   4. Ако ще се работи в RabbitMQ се използва RabbitMQPluginLibrary. Там са имплементирани помощни класове, които спестяват много писане. През nuget добавете EasyNetQ (команда Install-Package EasyNetQ) ;
   5. Задължително трябва да се добави reference към NetGenQueueServicePluginInterface. Там е дефиниран интерфейса IPlugin.
4. В съответния app.config трябва да се добави настройка за използване на конкретния plugin. Пример:

<pluginsettings>

<plugins>

<plugin alias="Producer"

assembly="SandboxPlugin"

type="SandboxPlugin.Logger2"

number-of-instances="1" />

<plugin alias="Consumer"

assembly="SandboxPlugin"

type="SandboxPlugin.RabbitToCommandLine"

number-of-instances="1" />

</plugins>

</pluginsettings>

1. Специално внимание трябва да се обърне на реализацията на функцията RabbitMQProducerPluginBase.Produce. Тя трябва да има най-общо следния вид:

protected override void Produce(CancellationTokenSource cancelationTokenSource)

{

// ако имаме цикъл

while (true)

{

// при всяка итерация се проверява дали service е поискал прекъсване на операцията

// ако е така - прекъсваме

if (cancelationTokenSource.IsCancellationRequested)

{

break;

}

try

{

// тук вършим полезната част

// всички команди да обвити от try / catch

log.Info("Publishing a test message");

// тук вършим нещо полезно (примерно генерираме обект и го изпращаме на заека)

//bus.Publish(generateTestNotificationToken());

// този sleep е за тест, в реална ситуаци

//Thread.Sleep(1000);

}

catch (Exception ex)

{

// ако е възникнало изключение

// се прави проверка дали service изисква

// прекъсване – ако не е така – хвърляме exception-а

// и оставяме service-а да се оправя (а той ще пробва да рестартира този plugin)

if (!cancelationTokenSource.IsCancellationRequested)

{

throw ex;

}

}

}

}

1. Enjoy!