Материалы к модулю Messaging:

1. Ознакомьтесь со сценариями использования Messaging, а также основными паттернами и стандартами:

Материал	**Время на изучение**
• Messaging - Message services scenario.mp4 • Messaging - Patterns.mp4 • Messaging - Standards.mp4	50 минут

2. Исходя из ваших предпочтений, изучите материалы по одному из известных брокеров сообщений, который вам наиболее интересен (выберите один из предложенных ниже или же какой-нибудь другой брокер):

Брокер	**Материал**	**Время на изучение**
Kafka	Lynda Course: Kafka Essential Training	1 час 20 минут
Azure Service Bus	Lynda Course: Azure Service Bus	46 минут
RabbitMQ	Official site: RabbitMQ	-

Также вы можете выбрать MSMQ или Windows Service Bus – несколько устаревшие, тем не менее, до сих пор используемые технологии:

Брокер	**Материал**	**Время на изучение**
MSMQ	Message Queues - MSMQ.mp4	22 минуты
Windows Service Bus	Message Queues - ServiceBus.mp4	30 минут

3. Выполните приведенное ниже задание, используя выбранный брокер сообщений.

Задание к модулю Messaging

Общее

В данной работе мы разработаем систему обработки результатов потокового сканирования.

Традиционно принято подобного рода системы разбивать на отдельные независимые службы, которые могут устанавливаться как на один компьютер, так и на разные. Например, может существовать следующая конфигурация:

- **Сервера захвата (ввода) документов**. Обычно их бывает несколько, и они устанавливаются на разные компьютеры – туда, где происходит ввод документов (изображений). Их задача – собирать документы и передавать на сервера трансформации
- **Сервера трансформации**. Их также может быть несколько, но по другой причине – таким образом балансируют нагрузку. Такие сервера могут производить различные типы обработок: конвертация, извлечение текста (OCR), отправка в СЭД, …
- **Центральный управляющий сервер**. Как правило он один и его задача – мониторинг состояния работы остальных серверов и передавать им настройки.

Мы будем реализовывать чуть более простую модель, состоящую из 2-х элементов: служб ввода и центрального сервера.

Примечание!!! Прежде чем приступать к выполнению задания, рекомендуется обсудить с ментором детали реализации:

- Какую(-ие) использовать очередь(и) сообщений (MSMQ/RabbitMQ/Kafka…)
- Архитектуру решения (где, сколько и каких очередей использовать, …).

* вместо «службы», в целях экономии времени, можно реализовать обычное консольное приложение, сосредоточившись на Message Queues.

Задание: Централизация сбора результатов работы служб ввода

1. Разработайте службу центрального сервера, которая будет делать следующее:
- При установке (или первом запуске) создавать очередь для приема результатов (готовых документов) от служб ввода
- Слушать эту входящую очередь и сохранять на диск все пришедшие документы

2. Разработайте службу ввода, которая будет слушать определенную директорию на рабочей станции, вычитывать из нее документы определенного формата (скажем, PDF) и отсылать их через службу сообщений на центральный сервер.

Для упрощения архитектуры примем, что в сети одновременно могут работать несколько агентов (на разных компьютерах), но только 1 центральный сервис.

Примечание!!! Одна из сложностей данного задания: лимит на размер одного сообщения. Суть в том, что как правило, очереди сообщений лимитируют размер одного сообщения (особенно это касается облачных платформ) и получаемый файл, скорее всего, этим лимитам удовлетворять не будет.

Для пересылки больших объемов данных в очереди (там, где имеется лимит и невозможно использовать другие способы передачи) используется подход Message Sequence (по ссылке приведен только рисунок, но общую идею он проясняет).

Обсудите с ментором, какой вариант борьбы с ограничением на размер сообщения вы выберите.

Создайте UML схему для вашего решения.