

Задание к модулю Messaging

Общая информация

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: службы ввода и центрального сервера.

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

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

** вместо «службы», в целях экономии времени, реализуем обычные консольные приложения.*

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 (например, PDF) и отсылать их через службу сообщений на центральный сервер.

После того, как ваша система начнет работать стабильно - поэкспериментируйте с более тяжелыми файлами (300-500 Мб), при необходимости - перенастройте систему под другой формат (например, .mp4).

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

Примечание.

Одна из сложностей данного задания - лимит на размер одного сообщения. Message Queues имеют ограничения на максимальный размер сообщения, а размер файлов вполне может превышать этот размер.

Один из подходов, используемых для пересылки больших объемов данных в очередях - [Message Sequence \(по ссылке приведен только рисунок, но общую идею он проясняет\)](#).

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

Создайте UML схему для вашего решения ([Component diagram](#)).

<http://draw.io/> - один из инструментов для создания UML, approved freeware.