Isa.Flow.SQLExtractor

Консольное приложение должно выполнять 3 независимые функции:

1. Извлечение новых статей из БД MS SQL и размещение их в указанную очередь RabbitMq.
2. Извлечение идентификаторов удалённых из БД статей и размещение их в указанную очередь RabbitMq.
3. Извлечение изменённых статей из БД и размещение их в указанную очередь RabbitMq.

Все три функции должны иметь возможность работать независимо от других, в отдельных потоках.

После старта приложения функции могут запуститься автоматически или ждать команды на запуск в зависимости от соответствующего аргумента командной строки.

Каждую функцию можно запустить или остановить независимо от других функций. Команды на запуск/остановку должны приходить через инфраструктуру RabbitMq.

Для взаимодействия с RabbitMq программа должна использовать класс, унаследованный от BaseActor. В качестве ActorId использовать строку «SQLExtractor»

Программа должна использовать файл конфигурации sqlextractor.conf для хранения необходимых постоянных данных в формате JSON (например, параметры соединения с MS SQL и Rabbit).

Для хранения служебных данных в процессе работы программа должна использовать файл sqlextractor.json

Приложение должно вести лог – набор файлов в каталоге logs (использовать Serilog).

1. Извлечение новых статей.

Функция представляет собой бесконечное повторение одной итерации. Действия, выполняемые в итерации:

* Получаем идентификатор последней статьи на предыдущей итерации (из файла sqlextractor.json).
* Читаем из БД определённое кол-во статей, у которых ID больше чем ID последней статьи.
* Помещаем статьи в очередь RabbitMq.
* Запоминаем ID последней статьи в файле sqlextractor.json.

При запуске функции по команде извне должны быть переданы параметры запуска:

* Перерыв между итерациями в секундах(опционально)
* Кол-во статей, извлекаемое за одну итерацию(опционально)
* Имя очереди, в которую помещать извлеченные статьи (опционально)
* ID последней статьи (опционально)

Все параметры указаны, как опциональные. Т.к. все параметры запоминаются в sqlextractor.json и при следующей попытке запуска сначала берутся параметры из команды, а если не указаны, то из файла. Если параметр не указан ни там, ни там – ошибка.

Если функция запускается после старта приложения (аргумент командной строки –start), параметры запуска берутся из sqlextractor.json. Если какой-либо параметр не указан в файле, приложение стартует без запуска этой функции. Причина отказа запуска функции пишется в лог.

1. Извлечение идентификаторов удаленных статей

Функция представляет собой бесконечное повторение одной итерации. Действия, выполняемые в итерации:

* Читаем из БД из таблицы DeletedArticle все идентификаторы удаленных статей.
* Помещаем идентификаторы в очередь RabbitMq порциями по 10 штук.
* Очищаем таблицу DeletedArticle.

При запуске функции по команде извне должны быть переданы параметры запуска:

* Перерыв между итерациями в секундах(опционально)
* Размер порции, помещаемой в очередь(опционально)
* Имя очереди, в которую помещать порции (опционально)

Все параметры указаны, как опциональные. Т.к. все параметры запоминаются в sqlextractor.json и при следующей попытке запуска сначала берутся параметры из команды, а если не указаны, то из файла. Если параметр не указан ни там, ни там – ошибка.

Если функция запускается после старта приложения (аргумент командной строки –start), параметры запуска берутся из sqlextractor.json. Если какой-либо параметр не указан в файле, приложение стартует без запуска этой функции. Причина отказа запуска функции пишется в лог.

1. Извлечение изменённых статей.

Функция представляет собой бесконечное повторение одной итерации. Действия, выполняемые в итерации:

* Читаем из БД определённое кол-во статей, у которых ID указаны в таблице UpdatedArticle.
* Помещаем статьи в очередь RabbitMq.
* Удаляем из таблицы UpdatedArticle идентификаторы тех статей, которые отправлены в очередь.

При запуске функции по команде извне должны быть переданы параметры запуска:

* Перерыв между итерациями в секундах(опционально)
* Кол-во статей, извлекаемое за одну итерацию(опционально)
* Имя очереди, в которую помещать извлеченные статьи (опционально)

Все параметры указаны, как опциональные. Т.к. все параметры запоминаются в sqlextractor.json и при следующей попытке запуска сначала берутся параметры из команды, а если не указаны, то из файла. Если параметр не указан ни там, ни там – ошибка.

Если функция запускается после старта приложения (аргумент командной строки –start), параметры запуска берутся из sqlextractor.json. Если какой-либо параметр не указан в файле, приложение стартует без запуска этой функции. Причина отказа запуска функции пишется в лог.

Для функции 1 и 3 объект, котроый помещается в очередь должен содержать след. Поля:

* Идентификатор статьи
* Дата создания записи о статье (Created)
* Дата публикации
* Заголовок
* Текст статьи
* Ссылка на статью
* Идентификатор источника
* Название источника
* Ссылка на источник

Для функции 2 в очередь помещаются только идентификаторы статей (или списки идентификаторов).