

Основное приложение

Основное приложение тестового задания: **task**.

Типы задач

Основное приложение может работать как в режиме *генератора*, так и в режиме *обработчика сообщений*.

Обработчик сообщений

При запуске приложения, оно стартует в режиме *обработчика* и генерирует свой идентификатор запросом:

```
INCR taskId
```

Затем пытается установить флаг активности генератора:

```
SET generatorIsActive 1 NX EX  
<2*таймаут_обновления_флага_активности_генератора>
```

Если приложению удаётся присвоить значение ключу, оно переключается в режим *генератора*. Это происходит с первым подключившимся приложением к базе и в дальнейшем, когда генератор завершает работу и флаг устаревает (см. ниже), с тем, которое успеет первым.

Если же флаг активности генератора уже создан и еще не устарел, приложение остаётся в режиме *обработчика* и периодически запрашивает у БД сообщение для обработки:

```
BLPOP messages <таймаут_ожидания_сообщения>
```

Также приложение периодически повторяет попытки стать генератором.

Генератор

Если задача становится генератором, она перестаёт запрашивать сообщения, и с периодичностью *<таймаут_обновления_флага_активности_генератора>* выполняет запрос:

```
SET generatorIsActive 1 EX  
<2*таймаут_обновления_флага_активности_генератора>
```

И конечно генерирует сообщения:

```
RPUSH messages <сообщение>
```

Монитор

Для удобства запуска и анализа поведения экземпляров основного приложения (задания) в процессе отладки создано приложение **monitor**. Приложение использует модуль **ansi-escape-sequences**.

Обмен сообщениями

Обмен сообщениями между монитором и задачами осуществляется через STDIN/STDOUT в формате JSON с разделением посылок CRLF.

Направление		Сообщение	Комментарии
Отправитель	Получатель		
Задача	Монитор	{"msg": "createdTaskId", "taskId": <число>}	Сообщение от задачи, после получения ей уникального идентификатора в БД.
Задача	Монитор	{"msg": "switchedToGenerator Mode"}	Сообщение от задачи-обработчика, после того, как она становится генератором.
Задача	Монитор	{"msg": "generated", "total": <число>}	Статистика от задачи-генератора, о количестве созданных сообщений в БД. Значение нарастающим итогом.
Задача	Монитор	{"msg": "processed", "total": <число>, "errors": <число>}	Статистика от задачи-обработчика, о количестве обработанных сообщений в БД и количестве ошибок. Значение нарастающим итогом.

Установка, настройка и запуск

Установка

Выполнить **npm install** для

- **/task/app**
- **/monitor/app**

Настройка

В конфигурационном модуле **/task/app/modules/config.js** установить:

- реквизиты доступа к redis
- таймаут генерации сообщения, по умолчанию - 500 мс

В конфигурационном модуле **/monitor/app/modules/config.js** установить:

- количество запускаемых заданий

Запуск

Запуск основного приложения (**/task/app/app.js**) самостоятельно:

- генератор/обработчик: **node app.js**
- получение ошибок: **node app.js getErrors**

Запуск монитора (**/monitor/app/app.js**):

- **node app.js**

При запуске монитора задания запускаются автоматически. Запущенные самостоятельно с монитором не взаимодействуют.

В запущенном мониторе доступно управление:

- клавиша **[q]**: завершение монитора и всего дерева порождённых им процессов
- клавиша **[k]**: завершение текущего генератора.