

При написании разрешено использовать любые встроенные nodejs модули(за исключением cluster), а также logger, underscore, async, step, redis, minimist.

Задача написать приложение, работающее с redis, и умеющее как генерировать сообщения, так и обрабатывать. Параллельно может быть запущено сколько угодно приложений.

Обмен любой информацией между приложениями осуществлять через redis.

Все запущенные копии приложения кроме генератора, являются обработчиками сообщений и непрерывно пытаются получить сообщение из redis.

Все сообщения должны быть обработаны, причём только один раз, только одним из обработчиков.

Генератором должно быть только одно из запущенных приложений. Т.е. каждое приложение может стать генератором. Но в любой момент времени может работать только один генератор. Если текущее приложение-генератор завершить принудительно (исключить обработчик завершения приложения), то одно из приложений должно заменить завершённое (упавшее) и стать генератором. Для определения кто генератор нельзя использовать средства ОС, считается что все приложения запущены на разных серверах и общаются только через redis.

Сообщения генерируются раз в 500 мс.

Для генерации сообщения использовать данную функцию:

```
function getMessage(){  
    this.cnt = this.cnt || 0;  
    return this.cnt++;  
}
```

Приложение, получая сообщение, передаёт его в данную функцию:

```
function eventHandler(msg, callback){  
    function onComplete(){  
        var error = Math.random() > 0.85;  
        callback(error, msg);  
    }  
    // processing takes time...  
    setTimeout(onComplete, Math.floor(Math.random()*1000));  
}
```

Если eventHandler выполняется с ошибкой (возвращает в callback(error_code)) то сообщение (msg) следует поместить в redis в место для ошибочных сообщений.

Если приложение запустить с параметром 'getErrors', то оно заберет из redis все сообщения с ошибкой и выведет их на экран и завершится, при этом из базы сообщения удаляются.

Проверить что приложение может обработать 1000000 сообщений

(для проверки 500мс таймаут можно убрать)

