ManyChat

Инерциальные очереди

```php
public function actionUpdate(UpdateRequest $request)
{
    $user = new User($request->getUserId());
    $user->updateData($request->getData());

    $queue = new RedisQueue(new Redis());
    $queue->send( queueName: 'userAnalytics', new UserAnalyticsMessage($request->getUserId()));
}
```

ManyChat

Инерциальные очереди

```php
class RedisInertialQueue extends AbstarctRedisService
{
    public function send(string $queueName, IMessage $message, int $timeout = 0)
    {
        $arguements = [$queueName, time() + $timeout, $this->serialize($message)];
        // redis> ZADD key score member
        return (boolean)$this->redis->rawCommand( command: 'ZADD', $arguements);
    }

    public function receive(string $queueName): ?IMessage
    {
        $queueMessage = $this->redis->eval( script: "
local val = redis.call('ZRANGEBYSCORE', KEYS[1], 0, ARGV[1], 'LIMIT', 0, 1)[1]
if val then
    redis.call('ZREM', KEYS[1], val)
end
return val
            ", [$queueName, time()], numKeys: 1);

        if (false === $queueMessage) {
            return null;
        }
        return $this->unserialize($queueMessage);
    }

    protected function serialize(IMessage $message): string{...}

    protected function unserialize(string $value): IMessage{...}
}
```