

Классическая очередь

```
public function actionUploadAsync(UploadRequest $request)
{
    $queue = new RedisQueue(new Redis());

    $queueMessage = new ImageProcessingMessage($request->getUploadedFile());
    $queue->send( queueName: ImageProcessingService::QUEUE_NAME, $queueMessage);

    return $this->sendResponse();
}
```

Классическая очередь

```
class RedisQueue extends AbstractRedisService
{
    public function send(string $queueName, IMessage $message)
    {
        // redis> RPUSH key value
        return $this->redis->rPush($queueName, $this->serialize($message));
    }

    public function receive(string $queueName): ?IMessage
    {
        // redis> LPOP key
        $queueMessage = $this->redis->lPop($queueName);
        if (false === $queueMessage) {
            return null;
        }
        return $this->unserialize($queueMessage);
    }

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

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