

# Mutex

```
public function release(string $name): bool
{
    if ($this->releaseLock($name)) {
        $index = array_search($name, $this->locks, strict: true);
        if ($index !== false) {
            unset($this->locks[$index]);
        }
        return true;
    }
    return false;
}

protected function releaseLock(string $name): bool
{
    $releaseScript = '
        if redis.call("GET",KEYS[1])==ARGV[1] then
            return redis.call("DEL",KEYS[1])
        else
            return 0
        end
    ';
    if (!isset($this->lockValues[$name])) {
        return false;
    }
    $arguments = [$releaseScript, 1, $name, $this->lockValues[$name]];
    if (!$this->redis->rawCommand(command: 'EVAL', ...$arguments)) {
        return false;
    }
    unset($this->lockValues[$name]);

    return true;
}
```

# Mutex

```
public function actionProcessSafe(ChargeRequest $request)
{
    $redisLock = new RedisLock(new Redis());
    $lockKey = "billing_" . $request->getUserId();

    if (!$redisLock->acquire($lockKey)) {
        throw new RuntimeException(message: "Can't process billing now");
    }

    $billingProcessor = new BillingProcessor($request->getUserId());
    $billingProcessor->process($request->getData());

    $redisLock->release($lockKey);

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