

Mutex

```
class RedisLock extends AbstractRedisService
{
    private $locks = [];
    private $lockValues = [];

    public function acquire(string $name, int $timeout = 0, int $expire = 30): bool{...}

    protected function acquireLock(string $name, int $timeout, int $expire): bool{...}

    public function release(string $name): bool{...}

    protected function releaseLock(string $name): bool{...}
}
```

Mutex

```
public function acquire(string $name, int $timeout = 0, int $expire = 30): bool
{
    if ($this->acquireLock($name, $timeout, $expire)) {
        $this->locks[] = $name;
        return true;
    }
    return false;
}

protected function acquireLock(string $name, int $timeout, int $expire): bool
{
    $value = md5( str: $name . random_int(0, PHP_INT_MAX));
    $waitTime = 0;
    $arguments = [$name, $value, 'NX', 'PX', $expire * 1000];
    while (!$this->redis->rawCommand( command: 'SET', ...$arguments)) {
        $waitTime++;
        if ($waitTime > $timeout) {
            return false;
        }
        sleep( seconds: 1);
    }
    $this->lockValues[$name] = $value;
    return true;
}
```