1 APR::ThreadMutex - Perl API for XXX

1.1 Synopsis

```
use APR::ThreadMutex ();
```

META: to be completed

1.2 Description

META: to be completed

1.3 API

`APR::ThreadMutex` provides the following functions and/or methods:

1.3.1 `DESTROY`

META: Autogenerated - needs to be reviewed/completed

Destroy the mutex and free the memory associated with the lock.

```
$mutex->DESTROY();
```

- **arg1: `$mutex` (`APR::ThreadMutex`)**

  the mutex to destroy.

- **ret: no return value**

1.3.2 `lock`

META: Autogenerated - needs to be reviewed/completed

Acquire the lock for the given mutex. If the mutex is already locked, the current thread will be put to sleep until the lock becomes available.

```
$ret = $mutex->lock();
```

- **arg1: `$mutex` (`APR::ThreadMutex`)**

  the mutex on which to acquire the lock.

- **ret: `$ret` (integer)**

1.3.3 ***`pool_get`***

META: Autogenerated - needs to be reviewed/completed

Get the pool used by this thread_mutex.

```
  $ret = $ob->pool_get();
```

- **arg1: `$ob` (`APR::ThreadMutex`)**
- **ret: `$ret` (`APR::Pool`)**

  apr_pool_t the pool

1.3.4 ***`trylock`***

META: Autogenerated - needs to be reviewed/completed

Attempt to acquire the lock for the given mutex. If the mutex has already been acquired, the call returns immediately with APR_EBUSY. Note: it is important that the APR_STATUS_IS_EBUSY(s) macro be used to determine if the return value was APR_EBUSY, for portability reasons.

```
  $ret = $mutex->trylock();
```

- **arg1: `$mutex` (`APR::ThreadMutex`)**

  the mutex on which to attempt the lock acquiring.

- **ret: `$ret` (integer)**

1.3.5 ***`unlock`***

META: Autogenerated - needs to be reviewed/completed

Release the lock for the given mutex.

```
  $ret = $mutex->unlock();
```

- **arg1: `$mutex` (`APR::ThreadMutex`)**

  the mutex from which to release the lock.

- **ret: `$ret` (integer)**

1.4 See Also

mod_perl 2.0 documentation.

1.5 Copyright

mod_perl 2.0 and its core modules are copyrighted under The Apache Software License, Version 1.1.

1.6 Authors

The mod_perl development team and numerous contributors.

Table of Contents: