



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'Apache::Session::Lock::Sybase.3pm'

\$ man Apache::Session::Lock::Sybase.3pm

Apache::Session::Lock::Sybase(3User Contributed Perl DocumentatApache::Session::Lock::Sybase(3pm)

NAME

Apache::Session::Lock::Sybase - Provides mutual exclusion using Sybase

SYNOPSIS

```
use Apache::Session::Lock::Sybase;
```

```
my $locker = Apache::Session::Lock::Sybase->new;
```

```
$locker->acquire_read_lock($ref);
```

```
$locker->acquire_write_lock($ref);
```

```
$locker->release_read_lock($ref);
```

```
$locker->release_write_lock($ref);
```

```
$locker->release_all_locks($ref);
```

DESCRIPTION

Apache::Session::Lock::Sybase fulfills the locking interface of Apache::Session. Mutual exclusion is achieved through the use of Sybase's `sp_getapplock` and `sp_releaseapplock` functions. Sybase does not support the notion of read and write locks, so this module only supports exclusive locks. When you request a shared read lock, it is instead promoted to an exclusive write lock.

CONFIGURATION

The module must know how to connect to your MySQL database to acquire locks. You must provide a datasource name, a user name, and a password. These options are passed in the usual Apache::Session style, and are very similar to the options for Apache::Session::Store::Sybase. Example:

```
tie %hash, 'Apache::Session::Sybase', $id, {  
    LockDataSource => 'dbi:sybase:database',  
    LockUserName  => 'database_user',  
    LockPassword  => 'K00!'  
};
```

Instead, you may pass in an already opened DBI handle to your database.

```
tie %hash, 'Apache::Session::Sybase', $id, {  
    LockHandle => $dbh  
};
```

AUTHOR

This module was written by Oliver Maul <oli@42.nu>.

SEE ALSO

Apache::Session