

```
---+ Package ==
```

```
=head1 NAME
```

CGI::Session::Driver::DBI - Base class for native DBI-related CGI::Session drivers

```
=head1 SYNOPSIS
```

```
require CGI::Session::Driver::DBI; @ISA = qw( CGI::Session::Driver::DBI );
```

```
=head1 DESCRIPTION
```

In most cases you can create a new DBI-driven CGI::Session driver by simply creating an empty driver file that inherits from CGI::Session::Driver::DBI. That's exactly what L<sqlite|CGI::Session::Driver::sqlite> does. The only reason why this class doesn't suit for a valid driver is its name isn't in lowercase. I'm serious!

```
=head2 NOTES
```

CGI::Session::Driver::DBI defines `init()` method, which makes DBI handle available for drivers in `I` - object attribute regardless of what `C<%dsn_args>` were used in creating session object. Should your driver require non-standard initialization you have to re-define `init()` method in your `F<.pm>` file, but make sure to set 'Handle' - object attribute to database handle (returned by `DBI->connect(...)`) if you wish to inherit any of the methods from CGI::Session::Driver::DBI.

```
=head1 STORAGE
```

Before you can use any DBI-based session drivers you need to make sure compatible database table is created for CGI::Session to work with. Following command will produce minimal requirements in most SQL databases:

```
CREATE TABLE sessions ( id CHAR(32) NOT NULL PRIMARY KEY, a_session TEXT NOT NULL );
```

Your session table can define additional columns, but the above two are required. Name of the session table is expected to be `I` by default. You may use a different name if you wish. To do this you have to pass `I` as part of your `C<%dsn_args>`:

```
$s = new CGI::Session("driver:sqlite", undef, {TableName=>'my_sessions'}); $s = new
CGI::Session("driver:mysql", undef, { TableName=>'my_sessions',
DataSource=>'dbi:mysql:shopping_cart'});
```

```
=head1 DRIVER ARGUMENTS
```

Following driver arguments are supported:

```
=over 4
```

```
=item DataSource
```

First argument to be passed to L<DBI|DBI>->L<connect()|DBI/connect()>. If the driver makes the database connection itself, it will also explicitly disconnect from the database when the driver object is DESTROYed.

```
=item User
```

User privileged to connect to the database defined in `C`.

=item Password

Password of the I privileged to connect to the database defined in C

=item Handle

An existing L database handle object. The handle can be created on demand by providing a code reference as a argument, such as C<<sub{DBI->connect}>>. This way, the database connection is only created if it actually needed. This can be useful when combined with a framework plugin like L<CGI::Application::Plugin::Session>, which creates a CGI::Session object on demand as well.

C will override all the above arguments, if any present.

=item TableName

Name of the table session data will be stored in.

=back

=head1 LICENSING

For support and licensing information see L<CGI::Session|CGI::Session>

This topic: TWiki > CGISessionDriverDBIDotPm

Topic revision: r1 - 2008-01-22 - TWikiContributor



Copyright © 1999-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback

Note: Please contribute updates to this topic on TWiki.org at TWiki:TWiki.CGISessionDriverDBIDotPm.