PHPLibを使ったプログラミングの実例

コバルト・ネットワークス株式会社
澁 谷　寿 夫

目次

- PHPLibとは
- セッション管理
- 認証機能
- デバッグ方法
- 実例

Web Application

- 要望
  - セッション機能
  - ユーザ毎の機能制限
  - セキュリティ
    - オートログアウト
    - 認証

PHPLibとは

- 機能
  - データベースアクセス：DB_Sqlクラス
    PostgreSQL, MySQL, Oracle, ODBCなどをサポート
  - セッション管理：Sessionクラス
    クッキーまたはGET変数を使用
  - 認証：Authクラス
    ライフタイムの設定
  - パーミッション管理：Permクラス
  - ユーザ管理：Userクラス

PHPLibの重要ファイル

- local.inc
  - Require($_PHPLIB[“libdir”] . “db_mysql.inc”);
    → db_pgsql.inc
- prepend.php3
  - 全てのページでロード
  - Php3.ini内のautoprependで設定

Databaseアクセスの抽象化

- 抽象化することの利点
  - データベースの変更が簡単
    MySQL→PostgreSQL
  - デバッグが容易
  - エラー処理

MySQL, PostgreSQLへのアクセス方法

内容	MySQL	PostgreSQL
接続	mysql_connect()	pg_connect()
クエリ	mysql_query()	pg_exec()
結果取得	mysql_fetch_array()	pg_fetch_array()
結果の行数	mysql_num_rows()	pg_num_rows()

デバッグが容易

- デバッグモード(MySQLのみ実装)

```
define(“DEBUG”, true);
$db->Debug = DEBUG;

<表示>
Debug: query = select * from test
```

エラー処理

DB_Sql 内のhaltをOverrideする

```
class test_db extend DB_Sql
{
    function halt($msg)
    printf(“データベースエラー:%s¥n", $msg);
    printf("PostgreSQL Error</b>: %s (%s)¥n",
    $this->Errno,
    $this->Error);
    die("Session halted.");
}
```

DB_Sql Example

```
$db = new Example_DB;     // 初期化
$db->connect(“test”, “localhost”, “username”, “”);        // 接
  続

$sql = “select * from test”;
$db->query($sql);    // クエリ実行

while($db->next_record()){
  while(list($key, $value) = each($db->Record)){
    print(is_string($key) ? “<b>$key</b>: $value<br>”: “”);
  }
  print(“<p>”);
}
```

Databaseへの接続設定

- 設定方法

```
$db = new Example_DB;
$db->Database= ‘test’;
$db->Host          = ‘localhost’;
$db->User          = ‘username’;
$db->Password      = ‘’;
```

- 自動接続

  $db->query()を呼ぶことにより自動接続

Session管理

- session.incで提供
- クッキーまたはGET変数を使用
- ページをまたがって変数を受け渡すことが可能

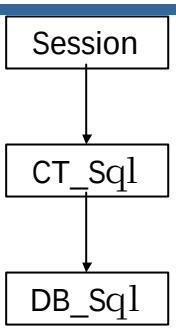
local.incの設定方法

```
class Cal_DB extends DB_Sql {
  var $Host     = "localhost";
  var $Database = "calendar";
  var $User     = "";
  var $Password = "";
}

class Cal_CT_Sql extends CT_Sql {
  var $database_class = "Cal_DB";          ## Which database to
    connect...
  var $database_table = "active_sessions"; ## and find our session
    data in this table.
}
```

local.incの設定方法(続き)

```
class Cal_Session extends Session {
  var $classname = "Cal_Session";

  var $allowcache     = "no";
  var $cookiename     = "";              ## defaults to classname
  var $magic          = "AlpHa";      ## ID seed
  var $mode           = "cookie";        ## We propagate session IDs
    with cookies
  var $fallback_mode  = "get";
  var $lifetime       = 0;               ## 0 = do session cookies, else
    minutes
  var $that_class     = "Cal_CT_Sql"; ## name of data storage
    container
  var $gc_probability = 5;
}
```

各クラスの関係

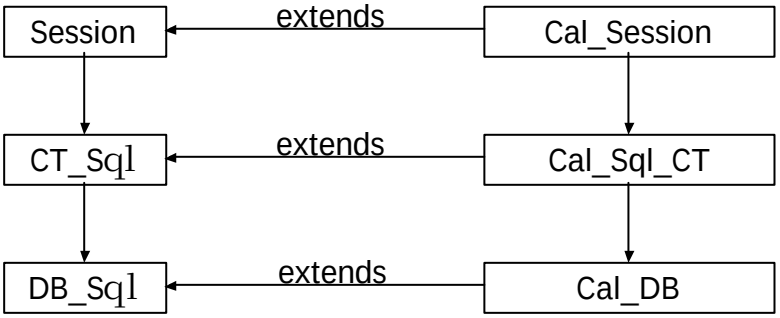


セッションID管理, 値のSerializing, 古いセッションの消去

セッション, ユーザ認証等データのロード, 保存

データベースへの接続, エラー処理

派生クラスの関係



local.incで定義

page_open()

```
page_open(array("sess"=>"Cal_Session",
                "auth"=>"Cal_Auth",
                "perm"=>"Cal_Perm"));

[…]

page_close();
```

注意点：page_open()はクッキー，HTTPヘッダを送るのでデータ出力前に呼ぶ必要がある。

スクリプトの表示前に空行が入ることがよくある。

purl(), url(), pself_url()

- GET変数でのセッションID

```
$link = $sess->url("script.php3");
```

```
script.php3?Cal_Session=ecfdeaaf850f79b1f4402ff48a702559
```

認証機能

- HTTP Basic認証の問題
  - ライフタイムの設定ができない(自動ログアウト)
  - 強制的にログアウトできない
  - グループでのパーミッション管理が必要な場合はアプリケーションで全てを作成する必要がある
  - ポップアップダイアログのデザインの変更ができない
  - PHPで利用するにはモジュール版しか利用できない

PHPLibでの認証

- PHPLibでの認証の利点
  - 認証するためのデータはたった一度送るだけで、その後は認証データはサーバ上のセッションに保存される
    HTTP Basic認証では毎回HTTPヘッダで送信される
  - 認証するために使うデータはDatabase以外にもLDAPやDBM等のほかの方式を利用することができる
    Authクラス内のauth_validatelogin()で実装できる
  - ディレクトリ単位ではなくファイル単位で制限をかけることができ、認証レベルを設定することができる
  - CGI版PHPでも利用することができる
  - 認証情報を破棄することができる(ログアウトボタン)
  - アイドルタイムを設定することで、自動ログアウトが可能
    セキュリティの向上ができる

認証実例

```
page_open(array("sess"=>"Cal_Session",
                "auth"=>"Cal_Auth",
                "perm"=>"Cal_Perm"));

printf("Your session id: %s<p>¥n", $sess->id);
printf("Your user ID: %s<p>¥n", $auth->auth["uid"]);
printf("Your user name: %s<p>¥n", $auth->auth["uname"]);
printf("Your permissions: %s<p>¥n", $auth->auth["perm"]);

page_close();
```

認証実例(結果)

Your session id:
d4d9df4d89d81a3d94992f24f8a1bb96

Your user ID:
9eda03dc8801523058abc42cc54da7df

Your user name: admin

Your permissions: admin,user

デバッグ

- DB_Sqlでのデバッグ
  - PostgreSQLの場合
    db_pgsql.incのfunction query内に

```
if ($this->Debug)
    printf("Debug: query = %s<br>¥n", $Query_String);
```

を書くことにより、以下の方法でページ中にクエリを表示することが可能

```
define(“DEBUG”, true);
$db = new DB_Example;
$db->Debug = DEBUG;
```

デバッグ

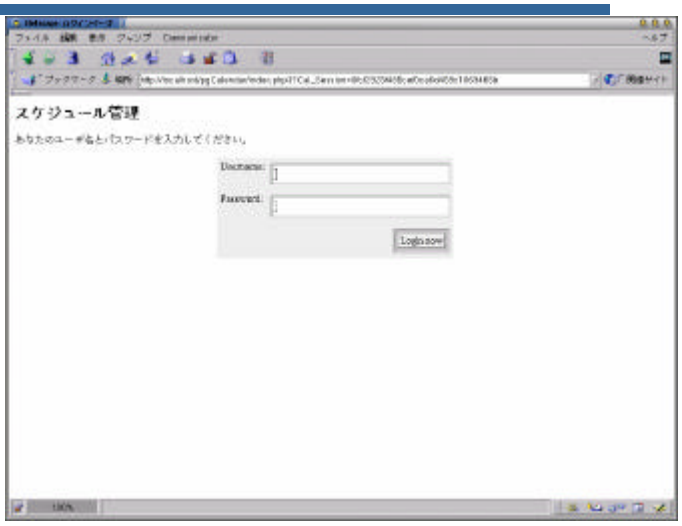
- クエリの実行にかかる時間の計測
  - db_pgsql.incのfunction query内のpg_Exec部分を以下のように変更

```
if($this->Debug)
  printf(“start time: %s<br>¥n”, localtime());

$this->Query_ID = pg_Exec($this->Link_ID,
  $Query_String);

if($this->Debug)
  printf("Debug: query = %s<br>¥n", $Query_String);
 printf(“start time: %s<br>¥n”, localtime());
```

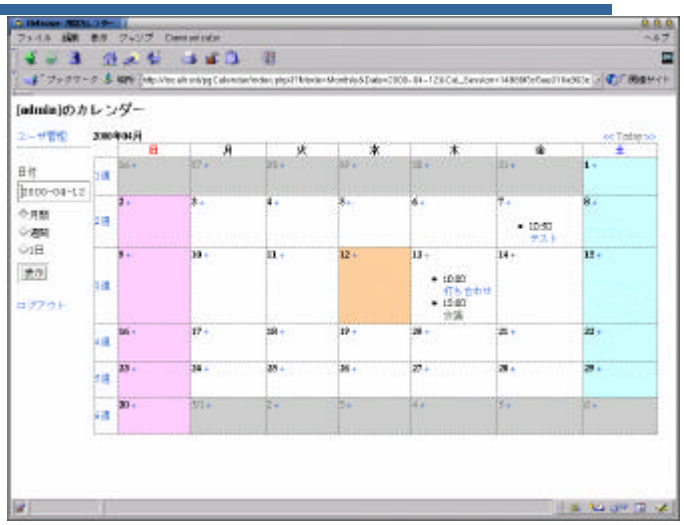
実例(スケジュール管理)

- 機能
  - 複数ユーザ対応
  - Admin権限者がユーザの追加
  - E-mail対応携帯電話等への通知機能
  - 自動ログアウト

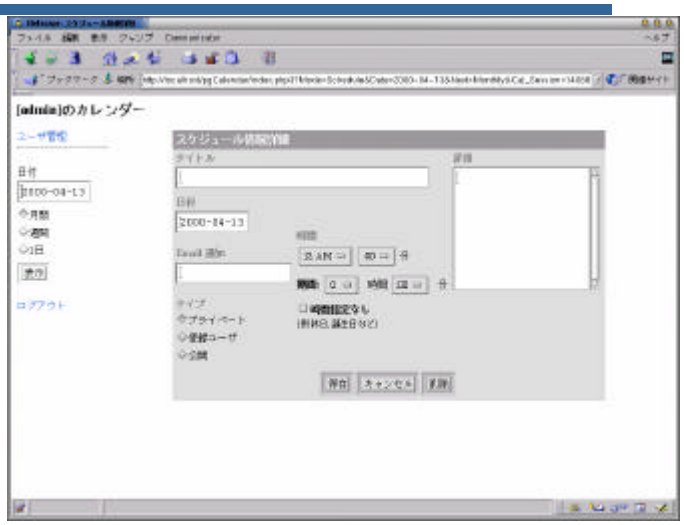
実例(続き)



実例(続き)



実例(続き)



実例(通知機能)

- PHP実行版を利用しcronで定期的に実行

```
# pgCalendar
HOME=/home/httpd/html/pgCalendar/bin
*/5 * * * * nobody /usr/bin/php -q ./notify.php3
```

他にもperlやpgbashなどでも可能です

Tips

- 1月のスケジュールの検索
  - 1日ごとに“select * from …”を実行すると30回以上クエリを実行
  - 添え字を使った連想配列に入れることにより1回のクエリで可能

例) $event["04-12"][0]["title"]

Tips(続き)

```
while($db->next_record()){
  $event_date = $db->f("date");
  if(isset($event[$event_date])){
    $count = count($event[$event_date]);
  } else {
    $count = 0;
  }

  $event[$event_date][$count]["title"]      = $db->f("title");
}
```

資料

- URL
  - 本家
    http://phplib.netuse.de/
  - PHPユーザ会上のPHPLib情報
    http://www.php.gr.jp/php/phplib/
  - pgCalendar
    http://alpha.or.jp/PHP/pgCalendar/
  - PostgreSQL, PHP-3.0.15-i18n_ja RPMS
    http://alpha.or.jp/Vine/VineSeed/