



# PHP::Session

---

Tatsuhiko Miyagawa

<[miyagwa@bulknews.net](mailto:miyagwa@bulknews.net)>

Livin' On The EDGE, Co.,Ltd.

2003/01/31 Shibuya.pm Technical Talks

# もくじ

---

- PHP::Session とは
- なぜ必要なのか
- PHPのセッション管理
- PHP::Session の実装
- 活用事例
- 今後の展開

# PHP::Session とは

---

## **DESCRIPTION**

PHP::Session provides a way to read / write PHP4 session files, with which you can make your Perl Application session shared with PHP4.

PHP::Session は PHP4 のセッションファイルの読み書きができ、これを利用することで、Perl アプリケーションのセッションをPHP4と共有することができる。

なぜ必要なのか

---

**基本的には必要ない(えー**

# 必要になるケース

---

- 他社の製作したPHPコードに機能追加
- PHPで書きたくない
- セッションの情報が必要

**Perl から PHPセッションを操作!**

# PHPのセッション管理

---

- PHP3 では PHPLIB というライブラリを使用する
- PHP4 では ファイルベースのセッション管理がデフォルト (PHPLIBも使用可能)
- 現在ではPHP4が主流

# PHP4のセッション管理実装

- `/tmp/sess_{$SESSION_ID}` というファイルに書き出される(セキュアなのか?)
- 排他処理は `flock` を使用
- 謎のエンコードフォーマット

```
baz|0:3:"foo":2:{s:3:"bar";s:2:"ok";s:3:"yes";s:4:"done";}arr|a:1:{i:3;0:3:"foo":2:{s:3:"bar";s:2:"ok";s:3:"yes";s:4:"done";}}!foo|
```

# エンコードフォーマット

変数名      Object  
                 バイト数 クラス名

```
baz|0:3:"foo":2:{s:3:"bar";s:2:"ok"  
;s:3:"yes";s:4:"done";}arr|a:1:{i:3  
;0:3:"foo":2:{s:3:"bar";s:2:"ok";s:  
3:"yes";s:4:"done";}}!foo|
```



# エンコードフォーマット (cont.)

key value  
string 長さ 値 string 長さ 値

```
baz|0:3:"foo":2:{s:3:"bar";s:2:"ok";s:3:"yes";s:4:"done";}arr|a:1:{i:3;0:3:"foo":2:{s:3:"bar";s:2:"ok";s:3:"yes";s:4:"done";}}!foo|
```

# エンコードフォーマット (cont.)

変数名 array int

```
baz|0:3:"foo":2:{s:3:"bar";s:2:"ok"  
;s:3:"yes";s:4:"done";}arr|a:1:{i:3  
;0:3:"foo":2:{s:3:"bar";s:2:"ok";s:  
3:"yes";s:4:"done";}}!foo|
```

値  
添字

# エンコードフォーマット (cont.)

---

```
baz|0:3:"foo":2:{s:3:"bar";s:2:"ok"  
;s:3:"yes";s:4:"done";}arr|a:1:{i:3  
;0:3:"foo":2:{s:3:"bar";s:2:"ok";s:  
3:"yes";s:4:"done";}}!foo|
```

↑  
undef

# 注意すること

---

- PHPには配列とハッシュの区別がない

```
$a[1] = "foo";  
$a[2] = new baz;  
$a[3] = "bar";  
$a[4] = -1.2;  
$data['hi'] = 'foo';  
$data['foo'] = "bar";
```

# エンコードパーサの実装

- 基本はシンプル
  - 正規表現でイける！

```
my $var_re = '(¥w+)¥|';  
my $str_re = 's:¥d+:"(.*)"';  
my $int_re = 'i:(-?¥d+);';  
my $dbl_re = 'd:(-?¥d+(?:¥.¥d+)?)';  
my $arr_re = 'a:(¥d+):';  
my $obj_re = 'O:¥d+:"(.*)"':¥d+:';  
my $nul_re = '(N)';  
my $bool_re = 'b:([01]);';
```

# エンコードパーサの実装(cont.)

---

- ハッシュの実装は再帰的
  - 正規表現だけでは無理 (5.6 以上なら?)
  - PHP のソースは1文字ずつガリガリ
    - Perl っぽくない
  - Parse::RecDescent
    - 重い (Parse::FastDescent?)
    - よくわからない

<宣伝> 「Perlデータマニピュレーション」に詳しい</宣伝>

# エンコードパーサの実装 (cont.)

---

- Text::Balanced を採用
  - Damian Conway 作
  - Parse::RecDescent の軽量版
  - 対応するカッコを見つけてくれる
    - これを再帰的に実行すればOK

```
my $bracket = extract_bracketed($data, '{}');
```

# PHP::Session の実装

---

- Serializer をPluggableな実装にした
  - PHP では WDDX 方式もある  
(PHP::Session では未実装だが、  
WDDX.pmを使用すれば容易)
  - Strategy Pattern
- sess\_\* ファイルがない場合はerror
  - create => 1 オプションで生成



# API

```
use PHP::Session;

my $session = PHP::Session->new($id);

# session id

my $id = $session->id;

# get/set session data

my $foo = $session->get('foo');

$session->set(bar => $bar);

# remove session data

$session->unregister('foo');
```

# API (cont.)

```
# remove all session data
$session->unset;

# check if data is registered
$session->is_registered('bar');

# save session data
$session->save;

# destroy session
$session->destroy;
```

# PHPの実装

---

- PHPの配列はPerlのハッシュ
  - Perl(配列) > PHP (配列) > Perl(ハッシュ)
- PHPのオブジェクトは  
PHP::Session::Object クラスに bless
  - `_class` キーにPHP側のクラス名を保持

# Apache::Session::PHP

---

- Apache::Session
  - セッション管理フレームワーク
  - Store, Lock, Generate, Serialize
- Serialize と Store だけつくればよい
  - Serialize: PHP::Session のモノ
  - Store: ファイル名規則を PHPとあわせる  
(`sess_${SESSION_ID}`)

```
tie %session, 'Apache::Session::PHP' ...;
```

# 活用事例

---

- 使ってる人はいるのか？
  - Bug Report や Patch はいっぱいくる
- Y! inc で採用(?)
  - PHPをテンプレートエンジンとして使う
  - セッションで変数の受け渡しを行う
- PHPのWebアプリケーションユニットテスト
  - セッションをシミュレート

# テンプレートエンジン

```
use Apache::Cookie;
use Apache::Request;
use Digest::MD5 qw(md5_hex);
use PHP::Session;

my $r = Apache::Request->new(Apache->request);
my $sid = get_sid() || gen_sid();
my $sess = PHP::Session->new($sid, { create => 1 });
$sess->set(name => 'miyagawa');
$sess->set(email => 'miyagawa@bulknews.net');
$sess->save();
```

# テンプレートエンジン (cont.)

```
my $cookie = Apache::Cookie->new($r,  
    -name => 'PHPSESSID',  
    -value => $sid, -path => '/',  
);  
$cookie->bake;  
$r->internal_redirect("/php-session/index.php");  
  
sub get_sid {  
    my %cookies = Apache::Cookie->fetch;  
    return ? $cookies{PHPSESSID} ?  
        $cookies{PHPSESSID}->value : undef;  
}
```

# テンプレートエンジン (cont.)

---

```
<?php
    session_start();
?>
Name: <?= $name ?><br>
Email: <?= $email ?><br>
SID: <?= $PHPSESSID ?>
```



# 今後の展開

---

- PHP::Session::MySQL
  - ストレージを MySQL にしただけ
- WDDX サポート
- Test::PHP
- Inline::PHP
- Tomcat::Session