



open-gorotto project
http://open-gorotto.jp/

open-gorotto 上のスクリプトの初期化について

open-gorotto/pub/以下に存在する一般スクリプトファイル

hogehoge.phtml

```
<?php
/*
 * (#) index.phtml
 * -----
 * html source : _index.html
 *
 * Copyright (C) 2005 open-gorotto project., all rights reserved
 * programmed by 2005.10.13 kobayashi,takao (aba@akadigi.jp)
 */
/*
 * SNSバージョン
 */
require_once(dirname(__FILE__)."/../sys/include/openg.php");
```

open-gorotto/sys/include/

openg.php

config.php

```
/*
 * 設定 : ホスト名を array() で列挙してください。
 */
$VHOSTS = array( "tokyo.interfair.jp" => True ,
                 "kumamoto.interfair.jp" => True ,
                 "aichi.interfair.jp" => True ,
                 "open-gorotto.jp" => True );
```

\$SERVER["SERVER_NAME"]と一致する
config を require する。

config.tokyo.interfair.jp.php

config.aichi.interfair.jp.php

config****.php には、virtual host毎の設定がされています。

各モジュールのディレクトリ直下の、_modules.php がincludeされます。
_modules.php 内で、\$OPENG_LINKS にリンク先を設定します。
return で、moduleid を返します。

modules/d/_module.php

```
global $OPENG_LINKS, $OPENG_BOOTURL;

define( MODVER_DIARY , "openg diary module 1.0(STD)" );
define( CONF_DIARY_TOP_LIMIT , 5 );

$OPENG_LINKS["diary.main"] = array("w">68,"h">32,"alt">"日記",
                                   "img">$OPENG_BOOTURL/skin/openg_diary.png",
                                   "img0">$OPENG_BOOTURL/skin/openg_diary.d.png",
                                   "img1">$OPENG_BOOTURL/skin/openg_diary.s.png",
                                   "img2">$OPENG_BOOTURL/skin/openg_diary.d.png",
                                   "href">$OPENG_BOOTURL/diary.phtml");

$OPENG_LINKS["diary.navi"] = array("w">110,"h">15,"alt">"日記",
                                   "img">$OPENG_BOOTURL/skin/diary_navi.png");

return "diary";
```

moduleid を modules.php に返します。
以降、この moduleid で、各モジュールにアクセスします。
\$OPENG_MODULESLOADED に、利用可能なモジュールが設定されます。

すべてのmodules/*/_module.php の、
module(moduleid)_initialize() が呼び出されます。

modules/d/_module.php

```
/* INITIALIZE */
function module_diary_initialize()
{
    return True;
}
```

modules.php

```
$OPENG_MODULES = array( "${OPENG_MODULESBOX}/X0" ,
                        "${OPENG_MODULESBOX}/top" ,
                        "${OPENG_MODULESBOX}/assns" ,
                        "${OPENG_MODULESBOX}/interest" ,
                        "${OPENG_MODULESBOX}/r" ,
                        "${OPENG_MODULESBOX}/e" ,
                        "${OPENG_MODULESBOX}/m" ,
                        "${OPENG_MODULESBOX}/b" ,
                        "${OPENG_MODULESBOX}/d" ,
                        "${OPENG_MODULESBOX}/c" ,
                        "${OPENG_MODULESBOX}/ca" ,
                        "${OPENG_MODULESBOX}/g" ,
                        "${OPENG_MODULESBOX}/a" ,
                        "${OPENG_MODULESBOX}/f" ,
                        "${OPENG_MODULESBOX}/ci" ,
                        "${OPENG_MODULESBOX}/n" ,
                        "${OPENG_MODULESBOX}/p" ,
                        "${OPENG_MODULESBOX}/l" ,
                        "${OPENG_MODULESBOX}/i" ,
                        "${OPENG_MODULESBOX}/bio" ,
                        "${OPENG_MODULESBOX}/pushcorn" ,
                        "${OPENG_MODULESBOX}/popcorn" ,
                        "${OPENG_MODULESBOX}/x" ,
                        "${OPENG_MODULESBOX}/X9" );
```

モジュール読込&初期化

```
modules_load_modules();
modules_links_initialize();
modules_call_initialize_function();
```

全モジュールのロードチェックを行います。

モジュール読込&初期化

```
modules_load_modules();
modules_links_initialize();
modules_call_initialize_function();
```

以上で、open-gorotto の初期化終了

require_once(dirname(__FILE__)."/../sys/include/openg.php");
実行後は、以下のようなグローバル変数が用意されます。

global変数

\$_VHOSTS	array::domain	VirtualHost ドメインリスト
\$OPENG_URI	string	アクセスされたURI
\$OPENG_ENCODEDURI	string	URLエンコードされた \$OPENG_URI
\$OPENG_MODULES	array::dir	ロードするモジュールリスト
\$OPENG_MODULESLoaded	array::moduleid	ロードされたモジュールリスト
\$OPENG_MODULEURL	array::moduleid => url	ロードされたモジュールのベースURL
\$OPENG_MODULESDIR	array::moduleid => dir	ロードされたモジュールのディレクトリ
\$OPENG_LINKS	array::moduleid.id => array:: w => num h => num img => string imga => string imgs => string priority => num href => string alt => string html => string	各モジュールによって定義された、各モジュール機能へのリンク先、画像、代替テキスト等 画像、テキストなど、部品の幅(px) 画像、テキストなど、部品の高(px) 通常の画像 マウスオーバーしたときの画像 機能実行中のときの画像 画像を並べる順番の優先順位リスト CONF_LAYOUTQUEUE_LEFT CONF_LAYOUTQUEUE_TOPPRIORITY (省略時)CONF_LAYOUTQUEUE_NORMALPRIORITY CONF_LAYOUTQUEUE_RIGHT 左端(1) 優先(50) 通常(100) 右端(999) クリックしたときのリンク先 画像の代替テキスト HTMLの記述
\$OPENG_TEMPLATES	array::	_template.html または _dtemplate.html に適用する HtmlTemplate データ
\$OPENG_ONLOAD_FUNCTIONS	array::	<body onload=""></body> 時に実行される JavaScript
\$OPENG_INHEADS	array::	<head> ... </head> 内に差し込む html など。
\$OPENG_LAYOUTQUEUE	array::	暫定レイアウト

require_once(dirname(__FILE__)."/../sys/include/openg.php");
実行後は、以下のような定数が用意されます。

const 定数

config.php (システム共通で定義される定数)

CR	"\n"
LF	"\r"
CONF_OGVERSION	string::ex"0.20.01(beta)" open-gorotto バージョン情報
CONF_SECURITY_GUEST	0 ゲストのセキュリティレベル
CONF_SECURITY_OUTER	10 サイト適用外住民のセキュリティレベル
CONF_SECURITY_INNER	20 サイト適用内住民のセキュリティレベル
CONF_SECURITY_COOPERATER	30 サイトボランティアのセキュリティレベル
CONF_SECURITY_HPWG	40 運営グループのセキュリティレベル
CONF_SECURITY_ADMIN	50 管理者のセキュリティレベル

config.<domain.tld>.php (各VirtualHost毎に定義される定数)

CONF_SITENAME	string	サイト名 (open-gorotto)
CONF_SITEEASY	string	略記用サイト名 (open-gorotto)
CONF_SITEABSTRACT	string	サイトの特徴 (自律分散型ソーシャルネットワーキングシステム)
CONF_PGSQL_DB	string	PostgreSQL データベース名
CONF_PGSQL_USER	string	PostgreSQL データベース・ユーザ名
CONF_PGSQL_PASSWD	string	PostgreSQL データベース・パスワード
CONF_PGSQL_HOST	string	PostgreSQL サーバアドレス
CONF_DIR	string	open-gorotto の ディレクトリ位置
CONF_UNIQID	string	キャッシュ識別用ID (変更すると、すべてのキャッシュがクリアされます)
CONF_SMTP	string	SMTP(25) MTA の存在するサーバアドレス
CONF_PUB	string	Internet上に公開するディレクトリ位置
CONF_SYS	string	open-gorotto システム周りが存在するディレクトリ位置
CONF_TMP	string	キャッシュ等、テンポラリファイルを保存するディレクトリ位置
CONF_IMGS	string	画像イメージが存在する URL
CONF_GANGAN	string	がんがんのリソースが存在する URL
CONF_CONVERT	string	ImageMagick の convert 実行用パス(/usr/local/bin/convert など)
CONF_URLBASE	string	サイトのURL (最後の / がないもの) http://open-gorotto
CONF_URL	string	サイトのURL (最後の / があるもの) http://open-gorotto/
CONF_EMAIL	string	open-gorotto 代表問い合わせ先 / 送信元 emailアドレス
CONF_ERRMAIL	string	open-gorotto から送られるメールがエラーメールとして帰ってくる、帰り先
CONF_DOMAIN	string	サイトのドメイン名 (open-gorotto.jp)
CONF_GANGANDOMAIN	string	がんがん用ドメイン名 (open-gorotto.jp)
CONF_SIG	string	サイトからユーザに送られるインターネットメールに添付されるシグネチャ
CONF_ISIG	string	サイトからユーザに送られる携帯メールに添付されるシグネチャ
CONF_CTGID_BBS	numeric	BBSスレッドを立てたときにデフォルトで選択されるカテゴリID (削除予定)
CONF_CTGID_DIARY	numeric	日記を作成したときにデフォルトで選択されるカテゴリID (削除予定)
CONF_GEOGAMIC_INTERFACE	string	geomagic GISサーバのインタフェースアドレス(削除予定)
CONF_GEOGAMIC_TIMEOUT	numeric	geomagic GISサーバの通信タイムアウト時間(秒)(削除予定)
CONF_GOOGLEMAPS_KEY	string	google map を利用する際に、googleから発行されるキー
CONF_ASSNS_X	float	地図を開いたときに中心となる google map 上の x座標値
CONF_ASSNS_Y	float	地図を開いたときに中心となる google map 上の y座標値
CONF_STYLESHEET	string	スタイルシートファイル名

ログイン状態・ログオフ状態

ログイン状態は、PHPの session機能 で保持されます。

ログイン状態の時

<code>\$_SESSION["_sUID"]</code>	会員番号(0以外)
<code>\$_SESSION["_sROST"]</code>	リモートホストIPアドレス

ログオフ状態

<code>\$_SESSION["_sUID"]</code>	0 または ""
----------------------------------	----------

関連関数

関数 現在ログインしている会員番号を取得する(openg.php)
`get_usercd()`

返値

0	未ログイン状態(ゲスト)
数値	会員番号

関数 ログイン状態かどうか判定する(openg.php)
`check_user()`

返値

false	未ログイン状態(ゲスト)
true	ログイン状態

画面遷移セッション管理

open-gorotto には、遷移管理ID/バッファが、セッション上に1個存在します。(`$_SESSION["_sid"]`)
`start_security(遷移管理ID)/check_security(遷移管理ID)/end_security(遷移管理ID)`

関連関数

関数 画面遷移を開始する。
`start_security(string 遷移管理ID)`

返値

なし。
遷移管理ID/バッファ=遷移管理ID の場合、遷移管理ID/バッファに遷移管理ID を上書きします。
遷移管理ID/バッファ<>遷移管理ID の場合、`$_SESSION` の [a-z]で始まるキーのデータを全てクリアし、遷移管理ID/バッファに遷移管理IDを上書きします。

関数 画面遷移をチェックする
`check_security(string 遷移管理ID)`

返値

なし。
遷移管理ID/バッファ=遷移管理ID の場合、なにもしません。
遷移管理ID/バッファ<>遷移管理ID の場合、"不正な処理"ページを表示して終了します。

関数 画面遷移終了
`end_security()`

返値

なし。
遷移管理ID/バッファに "" を上書きします。

関数 操作可能なセキュリティレベルかチェックします。
`check_security_level(string テーブル名,boolean ゲストリカバリ=false)`

返値

`select security from テーブル名 where usercd=ログイン中会員ID;` を実行し、レコードが存在すれば、そのまま素通りします。
レコードが存在しないとき、`ゲストリカバリ=true` の場合、ログイン画面に遷移します。ログインが完了したら、また元のスクリプトが実行されます。`ゲストリカバリ=false` の場合、ログインが必要な旨もしくは権限無しの画面のみを出力し、終了します。

データベースは、接続が必要になったとき自動的に接続が開始され、ページ終了時に自動的に接続を切断します。

関連関数

関数 ■PostgreSQL に SQL を発行する。

query_psql(string SQL)

返値

false 実行失敗
数値 実行結果レコード数

関数 ■PostgreSQL に、フォーマット付き書式でSQL を発行する。

queryf_psql(string フォーマット付きSQL[, string 引数])

フォーマット付きSQL sprintf と同じフォーマット
引数 フォーマットに適用する引数

返値

false 実行失敗
数値 実行結果レコード数

関数 ■PostgreSQL に、フォーマット付き書式でSQL 1レコード専用。

queryfni_psql(string フォーマット付きSQL[, string 引数])

フォーマット付きSQL sprintf と同じフォーマット
引数 フォーマットに適用する引数

返値

false 実行失敗
array 1レコードの入った配列

例

```
if( !($f=queryfni_psql("select * from test where usercd=%s",pgnum(0))) ) echo "error";
else echo "USERCD=${f["usercd"]}";
```

関数 ■SQLの実行が成功したか？

success_psql()

返値

true 成功
false 失敗

関数 ■SQLの実行が失敗したか？

failure_psql()

返値

true 失敗
false 成功

関数 ■直前の query_psql 又は queryf_psql で発行したSQLに対して1レコード フェッチする。

fetch_psql()

返値

false これ以上レコードはない
array 1レコードの入った配列

関数 ■直前の query_psql 又は queryf_psql で発行したSQLに対して全レコードを配列にフェッチする。

fetchall_psql()

返値

array 全レコードの入った配列

例

```
$fall = fetchall_psql();
foreach( $fall as $recidx => $f ) echo ${f["usercd"]};
```

関数 ■直前の query_psql 又は queryf_psql で発行したSQLに対する結果レコード数。

recs_psql()

返値

numeric レコード数

データベースは、接続が必要になったとき自動的に接続が開始され、ページ終了時に自動的に接続を切断します。

関連関数

関数 ■ 文字列をPostgreSQL用にサニタイジングした文字列を返す。
 pgstr(string 文字列)

返値
 string '(文字列)' or NULL

関数 ■ 数値をPostgreSQL用にサニタイジングした文字列を返す。
 pgnum(numeric 数値)

返値
 string 数値 or NULL

関数 ■ 数値をPostgreSQL用にサニタイジングした文字列を返す。
 pgbool(boolean ブール値)

返値
 string True
 False
 NULL

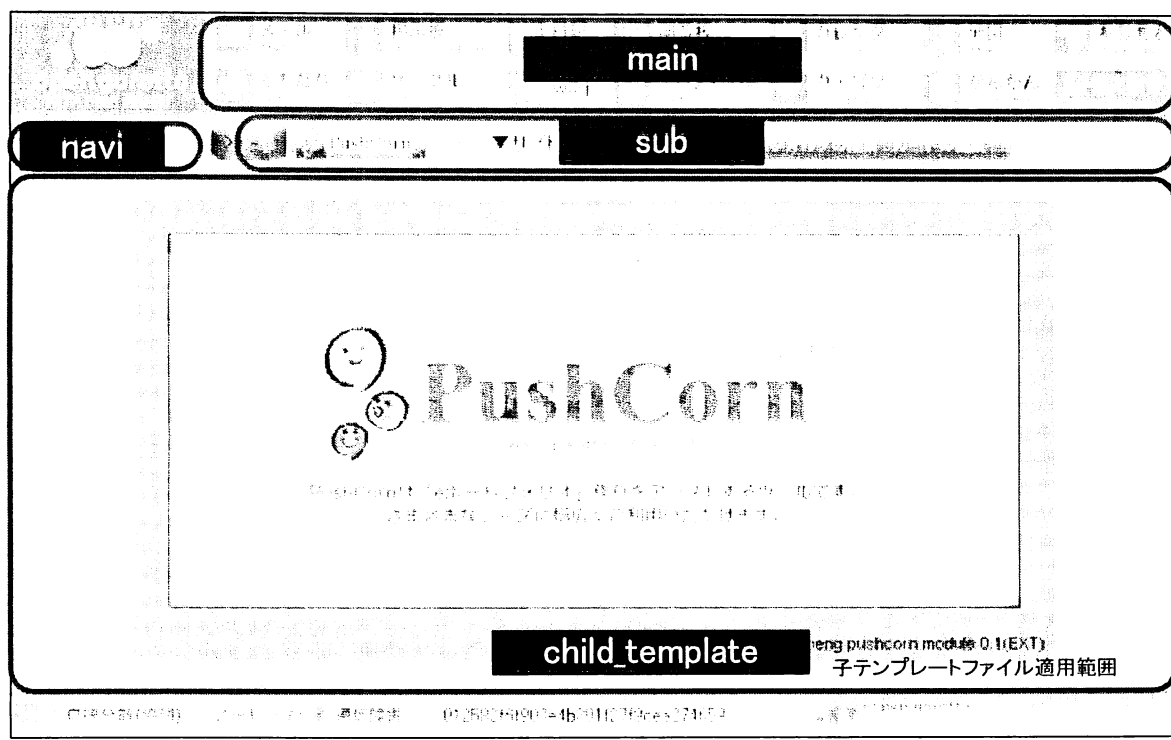
注意事項

queryを発行する文字列にはセキュリティホールにならないよう、必ず pgstr, pgnum, pgbool でサニタイジングを施した文字列を発行して下さい。

例

```
queryf_pgsql("select * from base where usercd=%s and handle=%s and flag=%s;" ,  
              pgnum($usercd), pgstr($string), pgbool(true));
```

テンプレートの適用について



template

pub/_template.html
pub/_dtemplate.html

親テンプレートファイル
適用範囲

hoge hoge.phtml

```
/*
 * テンプレート読み込み
 */
make_screen("PUSHCORN", "_site_edit.html", $data);
```

画面識別子

子テンプレートファイル

画面識別子を、全てのモジュールの module_hoge hoge_menu に送ることで、各モジュールが反応し、必要なアイコンのデザインを行った後、その配列を return します。

makescreen.php

```
/*
 * call on_before_queueing
 */
modules_call('on_before_queueing', $type);
```

\$OPENG_LAYOUTQUEUE = array() として初期化され、ロードされている全てのモジュールの、modules.*_on_before_queueing(画面識別子) が呼び出されます。

```
/*
 * queued
 */
$callnum = 0;
$result = modules_call('menu', $type);
foreach( $result as $moduleid => $menu )
{
    foreach( $divs as $div => $box )
```

ロードされている全てのモジュールの、modules.*_menu(画面識別子) が呼び出されます。各module では、
array("main" => array(\$OPENG_LINKS...,
"sub" => array(\$OPENG_LINKS...,
"navi" => array(\$OPENG_LINKS...))
の配列を返すことによって、\$OPENG_LAYOUTQUEUE に追加されていきます。

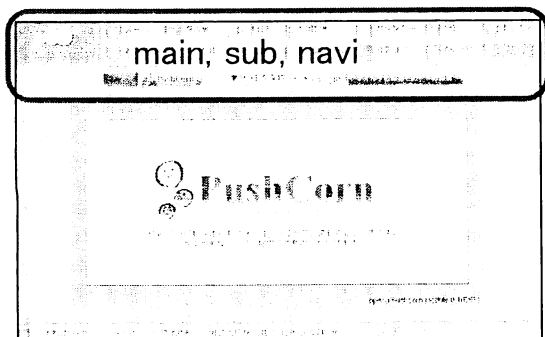
```
foreach( $divs as $div => $box ) usort($OPENG_LAYOUTQUEUE[$div], "makescreen_sort_cmp")
```

\$OPENG_LAYOUTQUEUE の main, sub, navi の各項目が、\$OPENG_LINKS の priority の順に並び替えられます。

```
/*
 * call on_after_queueing
 */
modules_call('on_after_queueing', $type);
```

modules.*_on_after_queueing(画面識別子) が呼び出されます。

テンプレート内の、main, sub, navi の部分に適用されます。



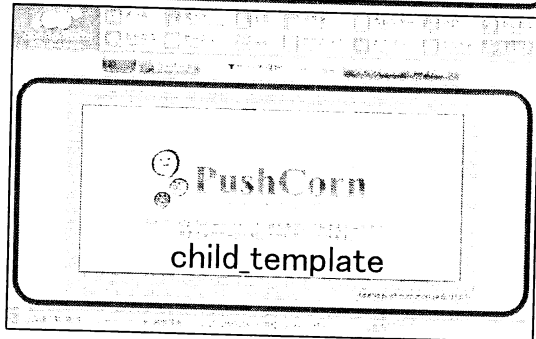
makescreen.php

```
/*
 * 強制上書きセット
 */
foreach( $OPENG_TEMPLATES as $key => $val ) $template[$key] = $val;
foreach( $OPENG_INHEADS as $script ) $template["in_heads"][] = array( "data" => $script );
foreach( $OPENG_ONLOAD_FUNCTIONS as $script ) $template["in_onload_functions"][] = array( "data" => $script );
```

親テンプレートの各部分に、
\$OPENG_TEMPLATES
\$OPENG_INHEADS
\$OPENG_ONLOAD_FUNCTIONS
の順に、内容が展開されていきます。

```
if( $encode=="EUC-JP" )
{
    HtmlTemplate::t_include( CONF_PUB."/". $template_file, $template );
}
else
{
    $buf = HtmlTemplate::t_buffer( CONF_PUB."/". $template_file, $template );
    echo mb_convert_encoding( $buf , $encode , "EUC-JP" );
}
```

header



child_template部分に、
子テンプレートファイルが適用されます。

データベースのclose

関連関数

関数 ■モジュールの指定機能呼び出し、結果を配列に格納する。

`modules_call(string 機能名 [,mixed パラメータ ...])`

返値

array 各モジュールの実行結果 array(moduleid => result)
数値 実行結果レコード数

例

```
$result = modules_call( 'version' );
```

例えば、日記モジュール、wikiモジュール、カレンダーモジュールがロードされている場合、
`$result["diary"] = module_diary_version( 'version' );`
`$result["wiki"] = module_wiki_version( 'version' );`
`$result["calender"] = module_calender_version( 'version' );`
と等価になります。ただし、各モジュールで、関数が定義されていない場合は、無視されます。

関連したトピック

トピック ■がんがんの貼り付けをおこなうボタンのhtmlを作成する場合。

例

```
foreach( modules_call( 'paste', "form", "textbox" ) as $html ) echo $html;
```

小村組

ホワワゴット完成

・今更-う-

・後継者 4人

SNF 時止に持てぬ
90-2 今更-う。

ホワワゴット 2-7 RSSの11月12日-21-22

本番入院 → やすかたに
2月末

2-7 2月末にRSS
7月

RSSの21日12月17日-21

関連関数

関数 ■エラーを表示して、終了する。
`gorotto_err_exit(string メッセージ)`

返値 なし(呼び出し後、直ちにスクリプトの実行を中止します)
エラー画面にメッセージを表示し、終了します。

関数 ■データベースをロールバックして、エラーを表示し終了する。
`gorotto_err_rollback(string メッセージ)`

返値 なし(呼び出し後、直ちにスクリプトの実行を中止します)
データベースを rollback したのち、エラー画面にメッセージを表示し、終了します。

関数 ■エラーを表示後、指定されたURLへジャンプします。
`gorotto_err_goto(string メッセージ, string URL, numeric タイマー)`

返値 なし
メッセージを表示後、タイマーで指定された時間を経過したら、自動的に URL で示されたアドレスへジャンプします。

関数 ■条件選択画面を表示し、指定されたURLへジャンプします。
`gorotto_confirm(string タイトル, string メッセージ, string YESのURL, string NOのURL)`

返値 なし
「はい」「いいえ」の確認画面を表示し、
「はい」の場合はYESのURLへ、「いいえ」の場合はNOのURLへジャンプします。

川添

RSS 追付
モジュール

(アサヒ)



・RSS 追付モジュール

OG モジュール (組込型) 対応

RSS 対応モジュール

対応品 Push Pop G/S 追付 3 次元 9 倍 SNS 対応
コンテナ

対応品、ハコ型対応品あり

RSS 追付 Module、P22222 対応 (開発中)
Push Pop 追付モジュール

3D プラント

追付 Push Pop モジュール

対応品

対応品 追付 追付

地域 SNS

対応品

小林
追付
追付
追付

追付

追付 追付

追付
追付
追付

追付 追付 追付

追付

追付 追付