

メール送信フォームを作成したい

さくらインターネット XREA ロリポップ

メール送信 | mb_send_mail() 関数 | ワンタイムチケット

4 5 5.3

関連

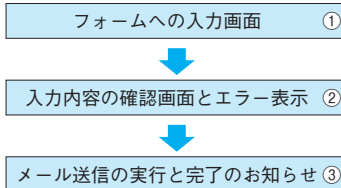
- 203 添付ファイル付きのメールを送りたい P.508
- 233 セーフモードとは? P.587
- 236 XSSとは? P.597

利用例 「お問い合わせ」フォームを作成したい場合

メール送信フォームを作成するには、mb_send_mail() 関数を使用します。

メール送信フォームは、CSRF対策 [レシピ238](#) や、予期しないキー操作や入力ミスを事前防ぐための確認画面表示をるようにします ([図6.4](#))。

図6.4 メール送信フォームの機能



NOTE

CSRF (クロスサイトリクエストフォージェリ)

攻撃用のWebページなどを作成して、Webページの閲覧者に意図せず別のWebサイト上で何らかの操作などを行なわせる攻撃手法のことです。このトピックで作成するメール送信フォームでも、CSRF対策をしていないと、他のWebサイトなどから送信操作が行なわれてしまう可能性があります。

メール送信フォームのHTMLとCSRF対策用ワンタイムチケットをuniqid() 関数 [レシピ221](#) を軸にして作成し、セッション [レシピ193](#) にその値を格納します。また、<input> タグの隠しフィールドにも同じ値を入れておきます。

●メール送信フォーム① (16-1.php)

```

<?php
セッションを開始します。
session_start();

ワンタイムチケットを生成します。
$ticket = md5(uniqid(mt_rand(), TRUE));

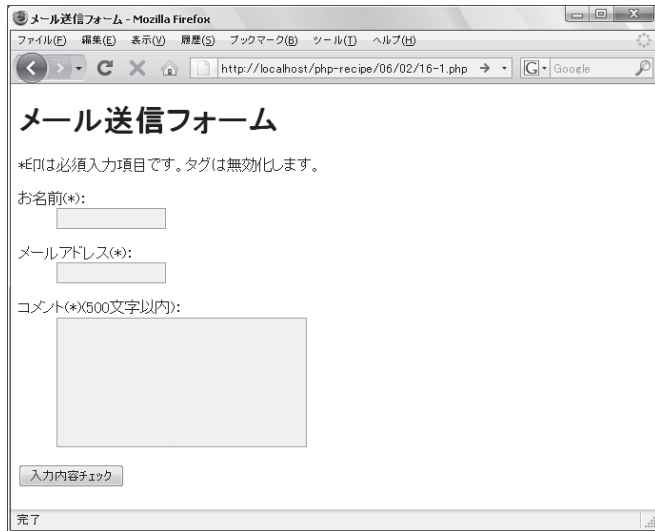
セッション変数にチケットを代入します。
  
```

```

$_SESSION['ticket'][] = $ticket;

function h($string) { // HTMLでのエスケープ処理をする関数
    return htmlspecialchars($string, ENT_QUOTES);
}
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>メール送信フォーム</title>
</head>
<body>
<div id="mailbox">
  <h1>メール送信フォーム</h1>
  <?php
エラーがあったら表示します。
    if(isset($_SESSION['error'])) {
        foreach ($_SESSION['error'] as $value) {
            echo ' <font color="red">' . h($value) . '</font><br />' . "\n";
        }
    }
?>
  <p>*印は必須入力項目です。タグは無効化します。</p>
  <form action="16-2.php" method="post">
    <dl>
      <dt><label for="name">お名前(*): </label></dt>
      <dd><input type="text" name="name" id="name"
        value="<?php echo h($_SESSION['name']);?>" /></dd>
    </dl>
    <dl>
      <dt><label for="email">メールアドレス(*): </label></dt>
      <dd><input type="text" name="email" id="email"
        value="<?php echo h($_SESSION['email']);?>" /></dd>
    </dl>
    <dl>
      <dt><label for="comment">コメント*(500文字以内): </label></dt>
      <dd><textarea rows="6" cols="30" id="comment" name="comment"><?php echo
        h($_SESSION['comment']);?></textarea></dd>
    </dl>
    <input type="hidden" name="ticket" value="<?php echo h($ticket);?>" />
    <input type="submit" name="submit" value="入力内容チェック" />
  </form>
</div>
</body>
</html>
  
```

▼実行結果



入力値をチェックする `checkInput()` 関数を作成し、`magic_quotes_gpc` が On でも Off でも同じようにスクリプトが動作するように処理し [レシピ181](#)、NULLバイトが含まれていないか [レシピ239](#)、文字エンコードが正しいかをチェック [レシピ230](#) します。

続いて、CSRF対策用ワンタイムチケットの確認を行ないます。不正な場合は終了します。そして、メール送信フォーム①からの入力値を検証します。問題なければ、入力されたデータをセッション変数に保存し、確認画面を表示します。表示する直前に、`h()` 関数で文字参照に変換します。入力に不備がある場合は、メール送信フォーム①に戻します。

●メール送信フォーム② (16-2.php)

```
<?php
入力値に不正なデータがないかなどをチェックする関数です。
function checkInput($var) {
    if (is_array($var)) {
        return array_map('checkInput', $var);
    } else {
        if (get_magic_quotes_gpc()) { // magic_quotes_gpc対策
            $var = stripslashes($var);
        }
    }
}
```

```
if (preg_match('/%0/', $var)) { // NULLバイト攻撃対策
    die('不正な入力です。');
}
if (!mb_check_encoding($var, 'UTF-8')) { // 文字エンコードの確認
    die('不正な入力です。');
}
return $var;
}
}
```

セッションを開始します。

```
session_start();
```

POSTされたデータをチェックします。

```
$_POST = checkInput($_POST);
```

チケットを確認します。

```
if (isset($_POST['ticket']) && isset($_SESSION['ticket'])) {
    $ticket = $_POST['ticket'];
    if (!in_array($ticket, $_SESSION['ticket'])) {
        die('不正アクセスの疑いがあります。');
    }
} else {
    die('不正アクセスの疑いがあります。');
}
```

変数にPOSTされたデータを代入します。

```
$name = isset($_POST['name']) ? $_POST['name'] : NULL;
$email = isset($_POST['email']) ? $_POST['email'] : NULL;
$comment = isset($_POST['comment']) ? $_POST['comment'] : NULL;
```

エラーメッセージを保存する配列を初期化します。

```
$error = array();
```

名前欄をチェックします。

```
if (trim($name) == '') {
    $error[] = 'お名前は必須項目です。';
} else if (mb_strlen($name) > 100) {
    $error[] = 'お名前は100文字以内でお願い致します。';
}
```

メールアドレス欄をチェック [レシピ175](#) します。

```
if (trim($email) == '') {
    $error[] = 'メールアドレスは必須項目です。';
} else {
    $pattern =
        '/^[a-z0-9\+\-]+'\.[a-z0-9\+\-]+'\.[a-z0-9\+\-]+'\.[a-z]{2,6}$/iD';
```

```

if (!preg_match($pattern, $email)) {
    $error[] = 'メールアドレスの形式が正しくありません。';
}

```

コメント欄をチェックします。

```

if (trim($comment) == '') {
    $error[] = 'コメントは必須項目です。';
} else if (mb_strlen($comment) > 500) {
    $error[] = 'コメントは500文字以内でお願い致します。';
}

```

POSTされたデータとエラーメッセージをセッション変数に保存します。

```

$_SESSION['name'] = $name;
$_SESSION['email'] = $email;
$_SESSION['comment'] = $comment;
$_SESSION['error'] = $error;

```

エラー数を確認します。

```

if(count($error) > 0) {

```

エラーがある場合は、入力フォームに戻します [レシビ208](#)。

```

    $dirname = dirname($_SERVER['SCRIPT_NAME']);
    $dirname = $dirname == DIRECTORY_SEPARATOR ? '' : $dirname;
    $uri = 'http://' . $_SERVER['SERVER_NAME'] .
        $dirname . '/16-1.php';
    header('HTTP/1.1 303 See Other');
    header('Location: ' . $uri);
}

```

確認画面を表示します。

```

} else {
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>内容確認</title>
</head>
<body>
<div id="mailbox">
<p>以下の内容でよろしければ送信ボタンを押してください。</p>
<dl>
<dt>お名前:</dt>
<dd><?php echo h($name);?></dd>
</dl>
<dl>
<dt>メールアドレス:</dt>
<dd><?php echo h($email);?></dd>
</dl>

```

```

<dl>
<dt>コメント:</dt>
<dd><?php echo nl2br(h($comment));?></dd>
</dl>

<form action="16-1.php" method="post">
<input type="submit" name="submit" value="入力画面に戻る" />
</form>
<form action="16-3.php" method="post">
<input type="hidden" name="ticket" value="<?php echo h($ticket);?>" />
<input type="submit" name="submit" value="送信する" />
</form>
</div>
</body>
</html>
<?php
}

```

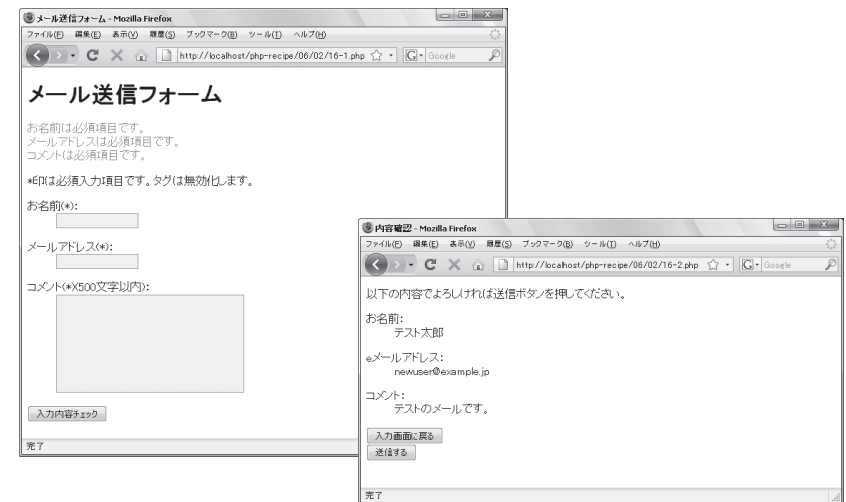
```

function h($string) { // HTMLでのエスケープ処理をする関数
    return htmlspecialchars($string, ENT_QUOTES);
}

```

?>終了タグ省略 [レシビ001](#)

▼実行結果 (上: エラーがある場合、下: 確認画面)



ここでも念のため、入力値をチェックした後、最終的な CSRF 対策用ワンタイムチケットの確認を行ないます。不正な場合は終了します。そして、mb_send_mail() 関数で実際のメール送信を行ないます。

先頭の \$mailTo 変数に送信先メールアドレスを必ず設定してください。このフォームからの入力内容が届きます。

●メール送信フォーム③ (16-3.php)

```
<?php
// メール宛先
$mailTo = 'newuser@example.jp';
// メールのタイトル
$subject = 'お問い合わせが入りました';
// Return-Pathに指定するメールアドレス
$returnMail = 'newuser@example.jp';
```

入力値に不正なデータがないかなどをチェックする関数です。

```
function checkInput($var) {
    if (is_array($var)) {
        return array_map('checkInput', $var);
    } else {
        if (get_magic_quotes_gpc()) { // magic_quotes_gpc対策
            $var = stripslashes($var);
        }
        if (preg_match('/%0/', $var)) { // NULLバイト攻撃対策
            die('不正な入力です。');
        }
        if (!mb_check_encoding($var, 'UTF-8')) { // 文字エンコードの確認
            die('不正な入力です。');
        }
        return $var;
    }
}
```

セッションを開始します。

```
session_start();
```

POSTされたデータをチェックします。

```
$_POST = checkInput($_POST);
```

チケットを確認します。

```
if (isset($_POST['ticket']) && isset($_SESSION['ticket'])) {
    $ticket = $_POST['ticket'];
    if (!in_array($ticket, $_SESSION['ticket'])) {
        die('不正アクセスの疑いがあります。');
    }
}
```

```
}
} else {
    die('不正アクセスの疑いがあります。');
}
```

変数にセッション変数を代入します。

```
$name = $_SESSION['name'];
$email = $_SESSION['email'];
$comment = $_SESSION['comment'];
```

mbstringの日本語設定を行ないます。

```
mb_language('ja');
mb_internal_encoding('UTF-8');
Fromヘッダーを作成します。
```

```
$header = 'From: ' . mb_encode_mimeheader($name) . ' <' . $email . '>';
送信結果をお知らせする変数を初期化します。
```

```
$message = '';
```

メールの送信と結果の判定をします。セーフモードがOnの場合は第5引数が使えません。

```
if (ini_get('safe_mode')) {
    $result = mb_send_mail($mailTo, $subject, $comment, $header);
} else {
    $result = mb_send_mail($mailTo, $subject, $comment, $header, '-f' . $returnMail);
}
```

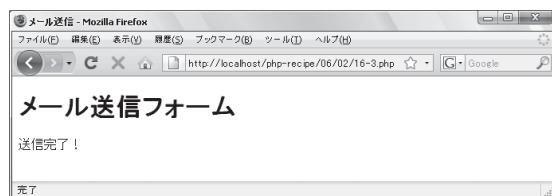
```
if ($result) {
    $message = '送信完了!';
}
```

セッション変数を破棄 [レシピ197](#) します。

```
$_SESSION = array();
session_destroy();
} else {
    $message = '送信失敗';
}
```

```
??
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>メール送信</title>
</head>
<body>
<div id="mailbox">
    <h1>メール送信フォーム</h1>
    <p><?php echo $message;?></p>
</div>
</body>
</html>
```

▼実行結果



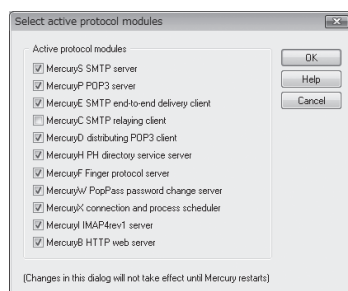
COLUMN Windowsでのメールの開発テスト

XAMPP for Windowsには、メールサーバーである「Mercury Mail Transport System」が含まれています。1台のPCでメールに関わる開発もできるようになるので、利用方法を説明します。なお、非商用なら無料で使用することができます。ライセンスについての詳細は、公式サイトを参照してください。

<http://www.pmail.com/>

1. XAMPPコントロールパネルの「Mercury」の右にある「開始」ボタンをクリックします。初回起動時には、Windowsファイアウォールのセキュリティ警告が表示されます。その場合、「ブロックを解除する」ボタンをクリックしてください。また、Windows Vistaでは「ユーザーアカウント制御」の警告が表示されますが「許可」をクリックして進んでください。「起動」と表示されれば起動済みです。
2. 「開始」ボタンの右の「Admin」ボタンをクリックすると、Mercuryの管理画面が起動します。「Configuration」メニューの「Protocol modules」を選択すると、「Select active protocol modules」ダイアログが表示されます（図A）。

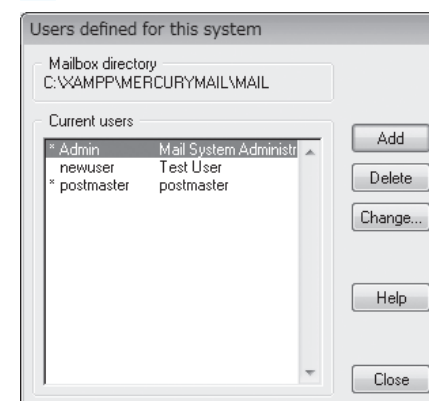
図A 「Select active protocol modules」



Apacheとの競争を防ぐため、一番最後の「MercuryB HTTP web server」のチェックを外したら、管理画面を開けてください。Mercuryは自動的に停止します。XAMPPコントロールパネルに戻り、「開始」ボタンをクリックして起動させます。

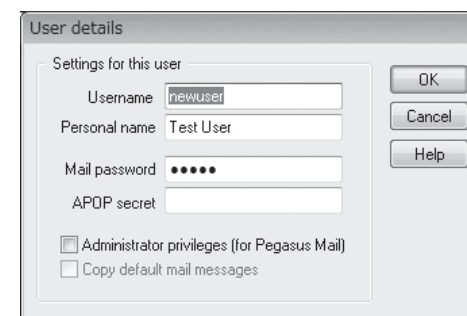
3. 続いて、開発で使用するためのメールアカウントを設定します。「Configuration」メニューの「Manage local users」を選択すると、「Users defined for this system」ダイアログが表示されます（図B）。

図B 「Users defined for this system」



4. 「Users defined for this system」ダイアログの「newuser」を一覧から選択して、「Change」ボタンをクリックすると、「User details」ダイアログが表示されます（図C）。

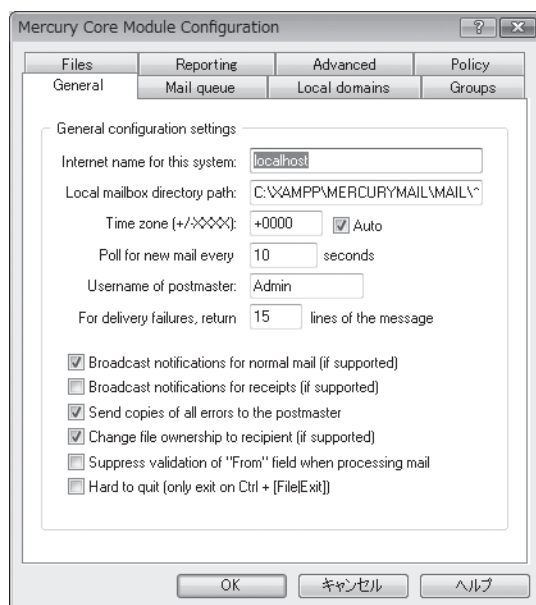
図C 「User details」



「newuser」の初期パスワードは「wampp」です。必要に応じて変更します。

5. [Configuration] メニューの [Mercury core modules] を選択すると、「Mercury Core Module Configuration」ダイアログが表示されます (図D)。

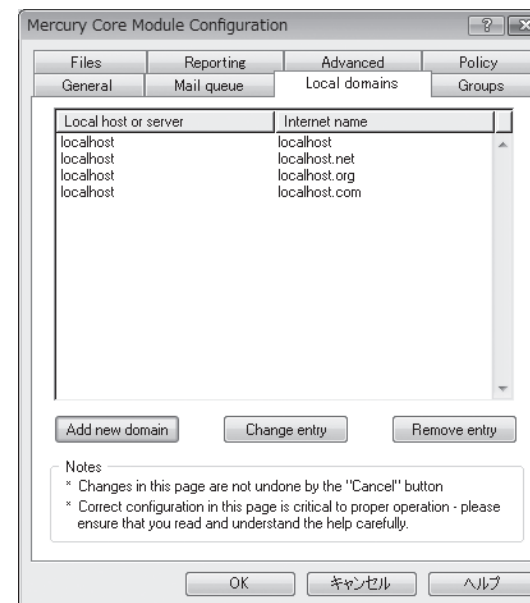
図D 「Mercury Core Module Configuration」



[General] タブの「Internet name for this system」に「example.jp」と入力します。このようなホスト名を設定することで、サンプルの実行とより実務的なメールアドレスのチェックが行なえるようになります。「example.jp」は他のホスト名でもかまいませんが、手順の「example.jp」をすべて置き換えてください。

6. 「Mercury Core Module Configuration」ダイアログの「Local domains」タブを選択します (図E)。

図E 「Local domains」タブを選択



[Add new domain] ボタンをクリックして、以下の内容を入力し、[OK] ボタンをクリックします。これでMercury内で「example.jp」が使えるようになります。

```
Localhost or server: localhost
Internet name: example.jp
```

設定が終わったら管理画面を閉じてください。Mercuryは自動的に停止します。XAMPPコントロールパネルに戻り、[開始] ボタンをクリックして起動させます。

7. [hosts] ファイルの編集を行ないます。hostsファイルはWindowsのシステムファイルの1つで、IPアドレスとホスト名の対応を記述したテキストファイルです。たとえば、「localhost」と入力しても「127.0.0.1」になるのは、ここで名前の解決が行なわれるからです。「hosts」ファイルの編集で、ご使用のパソコン内に限り、「example.jp」というホスト名が利用できるようになります。メモ帳で以下のファイルを開きます。

```
C:\Windows\system32\drivers\etc\hosts
```

NOTE

Windows Vistaでのhostsファイルの編集

OSの[スタート]メニューをクリックし、[すべてのプログラム] → [アクセサリ]の順に選択します。[メモ帳]を右クリックし、[管理者として実行]を選択します。そして、hostsファイルを開きます。

localhostの後に半角スペースと「example.jp」を加えて保存します。

```
127.0.0.1 localhost example.jp
```

8. パソコン内のDNSキャッシュをクリアします。古いDNS情報が残っていると、「example.jp」でメールの送受信が行えないことがあります。コマンドプロンプトを起動し、以下の文字列を入力して[Enter]キーを押します。処理が終わったらコマンドプロンプトのウィンドウを閉じます。

```
ipconfig /flushdns
```

NOTE

Windows Vistaでのflushdns操作

OSの[スタート]メニューをクリックし、[すべてのプログラム] → [アクセサリ]の順に選択します。[コマンドプロンプト]を右クリックし、[管理者として実行]を選択します。そして、上記のコマンドを実行します。

以上で、「newuser@example.jp」でのメールの送受信が行なえるようになります。メールクライアントソフトでは、POPサーバーとSMTPサーバーにそれぞれ「example.jp」を指定し、パスワードは手順4で設定した値を入力します。実際に送受信できるかどうか試してみましょう。