

○ 「 WebRequest()の使い方 (その 2) ;

POST 機能で、ローカル・サーバーに自動ログインする」

2015. 04. 28

・アメンボです、

とんでもなく長丁場になってしまいました！

前回「(その1)」から、4か月以上もかかるなんて！

たった1つの関数の調査・解説なのですが、小生には未知の技術範囲が広がった！

解析している間に WebRequest の新バージョン(7引数)までアナウンスされてしまうとは！！

・本稿での解説は、

自 PC 内のローカル・サーバー「localhost」を「HTTP」アクセスする場合に限定します、理由は WebRequest(9 引数) の基本動作の解説に絞るためです。(応用が効くはずです)

・次稿では、

ローカル・サーバーではなく、外部の WEB サーバへの POST を試す予定です。

・それにしても、

WebRequest()関数は、何て関係する技術範囲が広いんでしょう！

メタ・クウォーツ社がもう少し技術解説してくれてるといいんですが。

正直言うと、「バージョン2 ; 7引数」のサンプル・コードは現時点でコンパイル出来ないのですが、コード内容はとても参考になりました。

(このサンプル・コードが無かったら、ずい分と遠回りをしたかもしたかもしれません)

・本稿はやたらと長いですが、

基本的に理解してほしい内容は少なく、且つ単純です。

まずは全体を大雑把に流した後に、知りたい部分のみを熟読してください。

<WebRequest() 関数を解析していく手順>

※上記の結果から、これから解析・調査していく手順を「現状では」下記の様に修正。

- ステップ 1 ; 解析環境とツール類の整備 ・ ・ 前稿「(その1)」
- ステップ 2 ; WebRequest()関数の解析 ; 9引数 ・ ・ **本稿**
- ステップ 3 ; WebRequest()関数の解析 ; 9引数 ・ ・ 外部のサイトをアクセスする
- ステップ 4 ; WebRequest()関数の解析 ; 7引数 ・ ・ 「バージョン2」の解析
- ステップ 5 ; ターゲットから得られるタグ・データの中から必要な情報を入手する
- ステップ 6 ; 得られた情報を E A の判断条件に加える方法について考察する

<動作確認環境>

※本稿作成に使用したソフト類のバージョンは以下です。(いずれも無料で入手できます)

- ・ MT4 ; version 4.00 Build765、 MetaEditor ; version 5.00 Buid1035
- ・ Firefox (ウエブ・ブラウザ) ; v37.0.2
- ・ XAMMP (ローカル・サーバー) ; v3.2.1
- ・ Fiddler Web Debugger (プロキシ・サーバー) ; v4.5.0.0

<本稿で使用したスクリプト>

まず、本稿に添付した下記「ZIP」ファイルを解凍してください。

 new_mql4_2015_04_01.zip	2015/04/24 20:25	ZIP ファイル	21 KB
---	------------------	----------	-------

解凍後表示 ; 「WebRequest (その2) 用「PHP と MQL4」スクリプト一式」フォルダ内の構成

<input type="checkbox"/> 名前	更新日時	種類	サイズ
 MQL4スクリプト	2015/04/17 0:49	ファイル フォルダ	
 疑似サーバー用 ; PHPスクリプト_05[セッション版]セット	2015/04/17 0:37	ファイル フォルダ	
 疑似サーバー用 ; PHPスクリプト_05[セッション無し]版セット	2015/04/18 0:38	ファイル フォルダ	
 最初に考えたPHPスクリプト_04[セッション版]セット	2015/04/24 19:37	ファイル フォルダ	

I. 「MQL4 スクリプト」フォルダ ;

<input type="checkbox"/> 名前	更新日時	種類	サイズ
 _post_login.mq4	2015/04/07 21:02	MQL4 Source File	8 KB
 _post_logout.mq4	2015/04/07 21:22	MQL4 Source File	6 KB
 _post_main.mq4	2015/04/07 22:42	MQL4 Source File	6 KB

II. 「疑似サーバー用 ; PHP スクリプト_05[セッション版]セット ;

<input type="checkbox"/> 名前	更新日時	種類	サイズ
 _login_05.php	2015/04/06 20:23	PHP ファイル	2 KB
 _logout_05.php	2015/04/07 21:34	PHP ファイル	1 KB
 _main_05.php	2015/04/05 17:50	PHP ファイル	3 KB
 _menu_05.php	2015/04/06 22:11	PHP ファイル	1 KB
 _no_05.php	2015/04/05 18:21	PHP ファイル	1 KB
 _sub_05.php	2015/04/05 18:08	PHP ファイル	1 KB

III. 「疑似サーバー用 ; PHP スクリプト_05[セッション版]セット ;

※収録ファイル名は、上記「2.」と全く同じ名称・構成なので、取り扱いにご注意！

IV. 「最初に考えた PHP スクリプト_04」 [セッション版]セット ;

<input type="checkbox"/> 名前	更新日時	種類	サイズ
 _login_04.php	2015/04/05 17:34	PHP ファイル	2 KB
 _logout_04.php	2015/03/25 20:16	PHP ファイル	1 KB
 _main_04.php	2015/03/25 19:22	PHP ファイル	3 KB
 _start_04.html	2015/03/25 19:16	Firefox HTML D...	1 KB

※「*_04.php」は参考として添付しました、LOGIN システムとしては、こちらの方が本来のコード構造なのだと思いますが、実は POST 動作確認に使いずらかったのです、その理由は伏せておきますので読者で考えてみてください。(頭の体操)

目 次 :

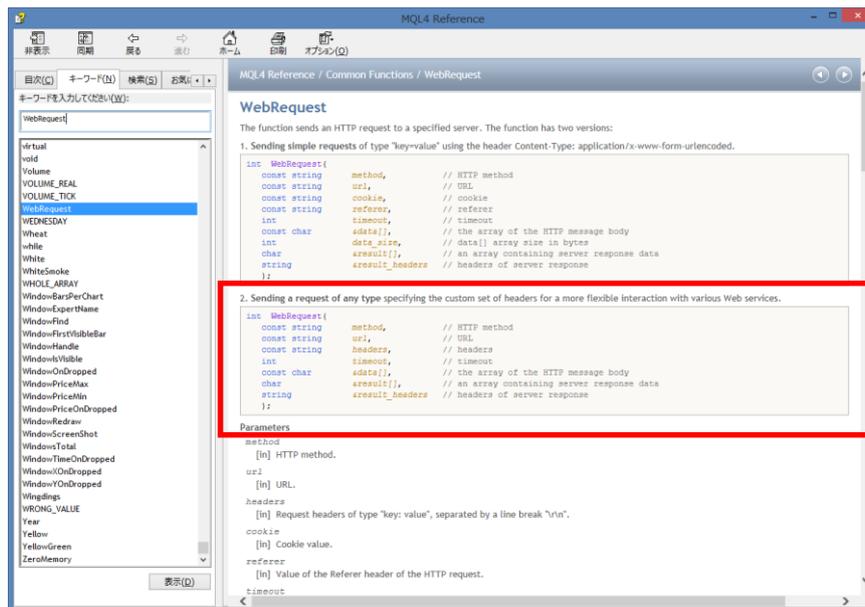
1. WebRequest()に新規バージョンが追加される? . . . P 4
 - (1) MQL4 Reference 内容が変更された! (追加された)
 - (2) 2つのバージョンの違いは何か?
 - (3) 引数の意味は何か?
2. 基礎知識の補足と再確認 . . . P 8
 - (1) HTTP セッションを理解する (そしてセッション・ハイジャックとは)
 - (2) Fiddler (バイオリン弾き) の補足
 - (3) HTTP と HTTPS の違い
 - (4) URL エンコード
3. WebRequest()解析用の疑似サーバー・スクリプトと MQL4 スクリプト . . . P 1 2
 - (1) ローカル・サーバーの仕様
 - (2) セッション ID (クッキー) のボタンタッチ
 - (3) スクリプト・サンプル (php コード) 概要
 - (4) MQL4 スクリプト概要
4. 動作・解析環境の準備 . . . P 1 4
5. ブラウザで「サンプル php コード (セッション版)」をアクセスしてみる . . . P 1 5
 - (1) 「ログイン画面」から順にアクセスする
 - (2) ブラウザで不正アクセスをしてみる (セッションの意味を確認)
6. 「サンプル php コード (セッション無し版)」で、不正アクセスしてみる . . . P 2 0
7. ブラウザによる「php コード (セッション版)」アクセスを Fiddler で観察する . . . P 2 1
 - (1) 初めに「localhost/_login_05.php」をアクセスしたとき
 - (2) 「ID」と「パスワード」を入力し、Login を選択したとき
 - (3) メニュー画面で、[投稿画面へ]を選択したとき
 - (4) 投稿をしているとき (2回、投稿した)
 - (5) 「ログアウト」を選択したとき
8. MT4 (MQL4) から「サンプル php コード」をアクセスする . . . P 3 8
 - (1) 準備と確認;
 - (2) 実行結果 (Fiddler の記録一覧)
 - (3) [_post_login.mq4] 実行結果
 - (4) [_post_main.mq4] 実行結果
 - (5) [_post_logout.mq4] 実行結果
9. スクリプト・サンプル; コード詳細 . . . P 5 7
 - (1) PHP コード (セッション版)
 - (2) PHP コード (セッション無し版)
 - (2) MQL4 コード

1. WebRequest() に新規バージョンが追加される？

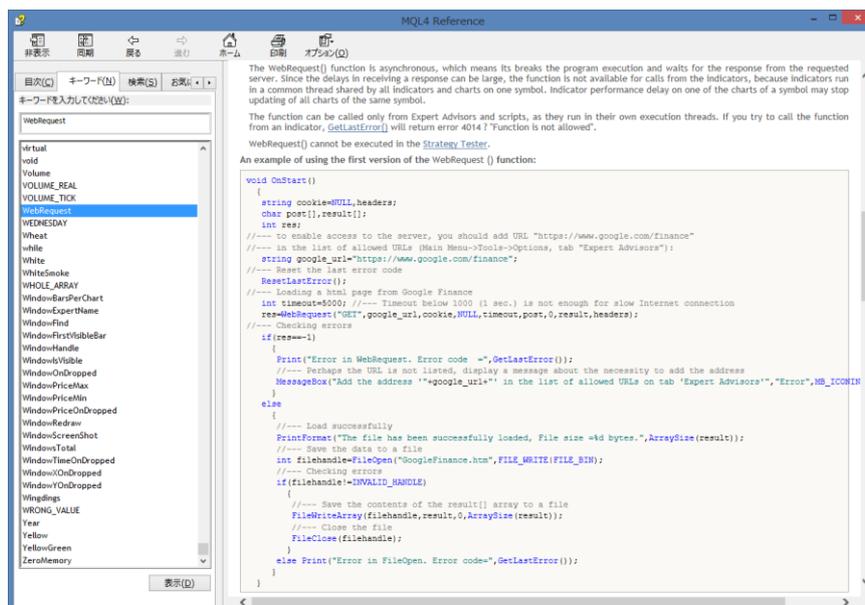
- ・小生の PC 上では、MT4 バージョンアップ (MT4 Build ; 745→765、MetaEditor ; 996→1035) に伴って MQL4 Reference が変更され、WebRequest() 新規バージョンが記載されているのですが、未だ新規バージョンの WebRequest() のコンパイルは出来ません。
- ・WebRequest() の現状バージョンも、突如 MQL4 Reference に現れたあと、しばらくの間はコンパイルが通りませんでした。

(1) MQL4 Reference 内容が変更された！ (追加された)

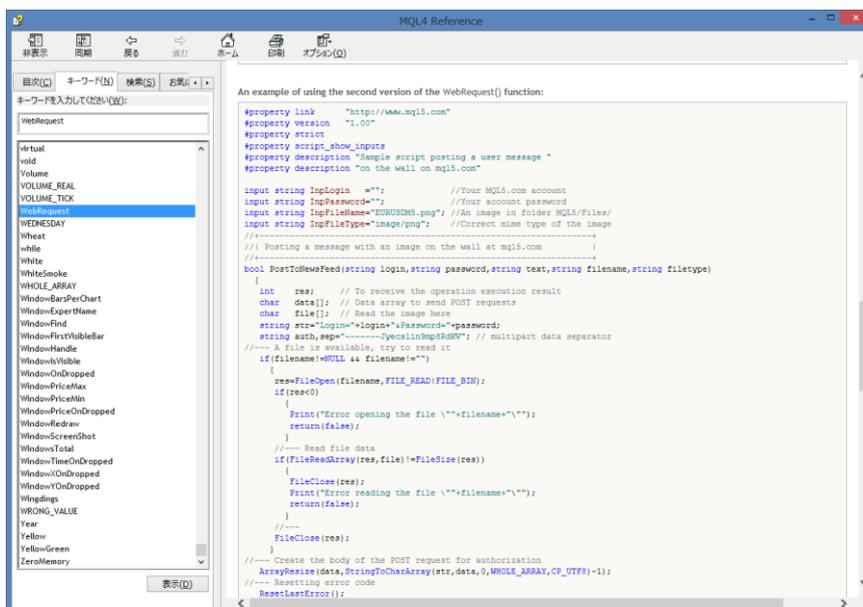
- ・すぐに気が付くのは、Build745 からあった「引数が 9 個」の関数に加えて、「引数が 7 個」の WebRequest() が追加されています。(解説では第 2 バージョンと記載されています)



↓ スクロール (バージョン 1 のスクリプト例)



↓ 更にスクロール (バージョン 2 のスクリプト例)



※第2バージョンのサンプル・コード（スクリプト）を動かそうとしたのですが、
 残念ながら「MT4;Build765、MetaEditor;1035」の現状ではコンパイル出来ませんでした。
 （バージョン1も MQL4 Reference に登場してからコンパイル出来るまで時間差がありました）

Build765 ;

バージョン		引数	コンパイル
1	WebRequest()	9 個	OK
2	WebRequest()	7 個	NG (未だダメ)

(2) 2つのバージョンの違いは何か？

- WebRequest()で POST 機能を利用して、何かデータを送付しようとするとき、バージョン1と2では、送付可能なデータ種類（MIME タイプ）と送付形式が異なります。

バージョン	引数	POST (送信) 可能なデータ			データ送信形式
		データ	拡張子 (例)	MIME タイプ (例)	
1	9 個	テキストのみ	.txt	text/plain	※1
2	7 個	ファイル、テキスト	.jpg、.txt	image/jpeg、text/plain	※2

※1 ; テキスト・データは「id=data」の形式で送付される、送付するデータが複数ある場合は「&」で区切られて「id1=data1&id2=data2」の形式で送付される。

また、送付データは下記の様に「URL エンコード (初期値)」される。

- 英数字と3種類の記号「_ . -」はそのまま使用され、
- スペースは「+」に変換され、
- 上記以外の文字は「%xx」の16進形式に変換される

※2 ; 「マルチパート・データ」として「テキスト、画像、ワードファイル、音声など」を送信することが可能と思われます。(なんせ、未確認)

また、「テキスト・データ」をJSON形式で送信することも可能なようです。

※ 「(MIME) マルチパート・データ」とは；

元々は、一通の電子メールに複数の異なる種類のデータを格納する方式から始まった。

(メールに ASCII 文字以外のデータを格納するための「MIME」規格の拡張仕様)

添付ファイルとして画像を送ったりするのに使われる。

従来、メールはヘッダと本文という 2 つの部分しか持てなかったが、マルチパート仕様により、「区切り文字」を境に任意の数の部分(パート)に分割することが可能となり、

1 つのメールに本文、画像、ワープロ文書、音声をまとめて送受信することが出来るようになった。(一つのパートを複数のマルチパートに分割する「入れ子」構造も可能)

(詳細はネット上の資料か、本稿シリーズ (その1) 「2章」を参照)

(3) 引数の意味は何か？

URL Reference 内容を整理する；

WebRequest()の引数	原文解説	version 1	version 2	解説
const string method	HTTP method	○	○	①
const string url	URL	○	○	②
const string cookie	cookie	○	—	③
const string referer	referer	○	—	④
const string headers	headers	—	○	⑤
int timeout	timeout	○	○	⑥
const char &data[]	the array of the HTTP message body	○	○	⑦
int data_size	data[] array size in bytes	○	—	⑧
char &result[]	an array containing server response data	○	○	⑨
string &result_headers	headers of server response	○	○	⑩

解説；本稿シリーズ (その1) と重複する部分もありますが、改めて解説します

①～⑧；ブラウザや WebRequest()から発行する「要求 (Request)」内容を記述する部分です

⑨、⑩；サーバー側から返される「応答 (Response)」内容が収録される部分です。

	引数	内容
①	method	GET、POST などのメソッド指定 (他の HEAD などは動作未確認)
②	url	アクセス先を指定する
③	cookie	セッション継続時には、サーバーから発行されたクッキーの設定に使用
④	referer	現ページへの、アクセス元ページの url
⑤	headers	リクエスト・ヘッダーを指定する
⑥	timeout	インターネット接続までのタイムアウト設定 (ミリセカントで記述)
⑦	&data[]	メッセージ・ボディ (POST するデータを記述)
⑧	data_size	上記「&data[]」のデータ・サイズ
⑨	&result[]	応答 (Response) のボディ
⑩	&result_headers	応答のヘッダ部分

要求と応答の例 ;

[要求 (Request)]

```
POST http://kei1/_main_05.php HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
Content-Type: application/x-www-form-urlencoded
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
Accept-Charset: *,utf-8
Accept-Language: en
Cookie: PHPSESSID=f5rarnbl4kk32g4t5feot1pen5
Host: kei1
Referer: http://localhost/_menu_05.php
User-Agent: MetaTrader 4 Terminal/4.765 (Windows NT 6.2; x86)
Connection: Keep-Alive
Content-Length: 111
```

メッセージ・ヘッダ

<空行>

```
site_05=tamama&contents=gunnsou_san%OD%OA%OD%OAyoroshiku%OD%OAonegai_s
himasu&bt_05=Toukou_suru
```

メッセージ・ボディ



メッセージ・ヘッダ

[応答 (Response)]

```
HTTP/1.1 200 OK
Date: Tue, 07 Apr 2015 15:34:39 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
X-Powered-By: PHP/5.5.15
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 658
Keep-Alive: timeout=5, max=98
Connection: Keep-Alive
Content-Type: text/html
```

<空行>

```
<html>
<head>
<meta http-equiv="Content-Type" content="charset=utf-8">
<title>SITE5 TEST</title>
</head>
<body>
<p><掲示板></p>
<form method="POST" action="/_main_05.php">
<p>
お名前 : <input type="text" name="site_05"><br><br>
</p>
<p>
ご意見 : <br>
<textarea name="contents" rows="8" cols="40"></textarea><br><br>
</p>
<input type="submit" name="bt_05" value="Toukou_suru">
</form>
<ul>
<li><a href="_logout_05.php">ログアウト</a></li>
</ul>
</body>
</html>
```

メッセージ・ボディ

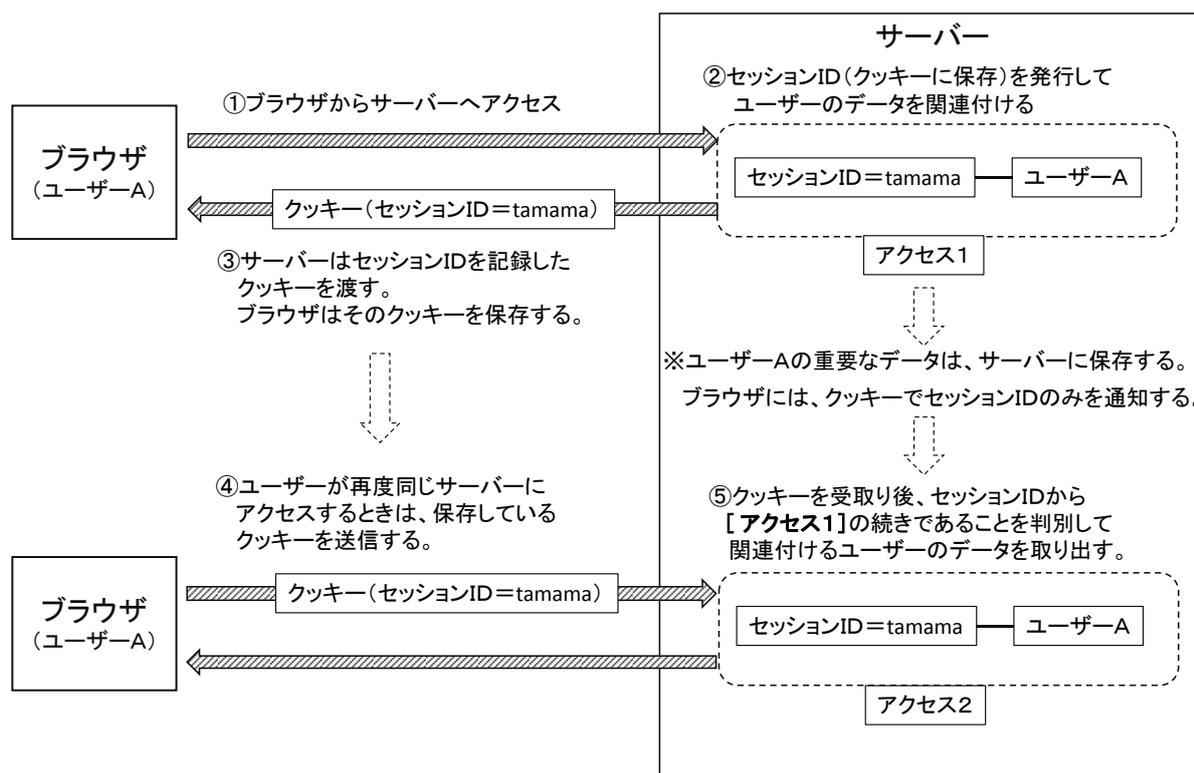
2. 基礎知識の補足と再確認

本章では `WebRequest()` 関数の挙動を理解し、利用する上で最低限必要と思われる事項を簡潔におさらいします。(と、言うか、小生自身が超初心者なので学び直した)

(1) HTTP セッションを理解する (そしてセッション・ハイジャックとは)

- ・本来 HTTP のアクセスは1つ1つ独立していて、そのままでは個別のアクセス間には何の関係性もありません。
- ・しかしながら、例えば皆さんがよく利用する「ログイン」システムをイメージすると判るのですが、各アクセス (ページ) 間を関連付けたい場合があります。
このような時に「セッション」を使います、ログイン中のユーザーがアクセスする各ページを関連付けていくのにサーバー側で発行・通知する「共通のセッションID」を使いまわすことで、同一ユーザーがアクセスしていることを保証するわけです。
- ・「セッションID」を、サーバーとブラウザ (ユーザー) 間で受渡しする方法には「Cookies (クッキー)」による方法と、「URL パラメータ」による方法の2通りがありますが、通常は「Cookies (クッキー)」による方法が安全なので、こちらを使います。

[セッション概念図]



- ・ユーザー B (セッション ID=tamama を知らない) は [アクセス 2] へ行くことが出来ません

[セッション・ハイジャック]

- ・セッションハイジャックとは、何らかの方法で他人の「セッションID」を探り出して、セッションを乗っ取ってしまうことです。
ハイジャックした後は、他人しかログイン出来ないはずのページ (情報) を自由に操れるということです。(例えば [アクセス 2] へ)

(2) Fiddler (バイオリン弾き) の補足

・Fiddler ではバージョンにより、「http://localhost」アクセスをキャプチャー出来ないことがあります。(小生は WebRequest()でのアクセス時に遭遇した)

解決策は以下の「3通り」の何れかです、

詳細は Fiddler の公式サイトの下記アドレスを参照ください。

<http://docs.telerik.com/fiddler/ObserveTraffic/Troubleshooting/NoTrafficToLocalhost#solution-3-updated-rules-file>

① 「http://localhost」の代わりに使用するパソコンの「http://マシン名」を使用する

② 「http://localhost」の代わりに、以下の何れかを使用する

http://ipv4.fiddler

http://ipv6.fiddler

http://localhost.fiddler

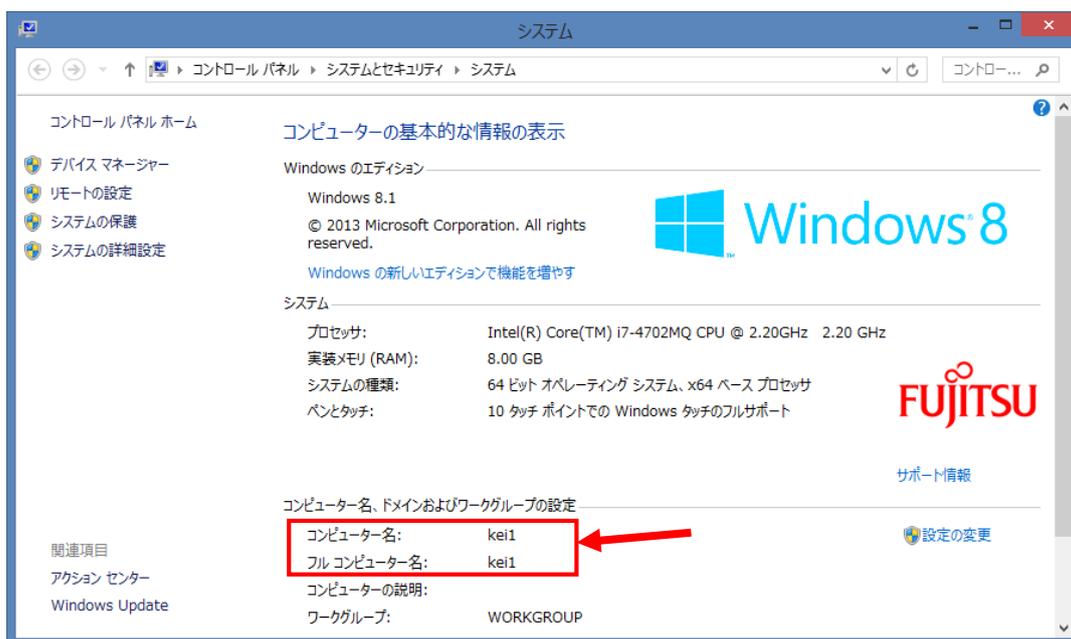
③ ルールファイルを書換える・・・小生は詳細を解析していません(理解していません)

※小生は、WebRequest()でのアクセス時には「http://マシン名」を採用しました。

マシン名とは、下記の何れかの操作で確認できます。

方法1；

デスクトップ上の「PC」－[右クリック]－[プロパティ]で表示される
下記の情報から読み取る。



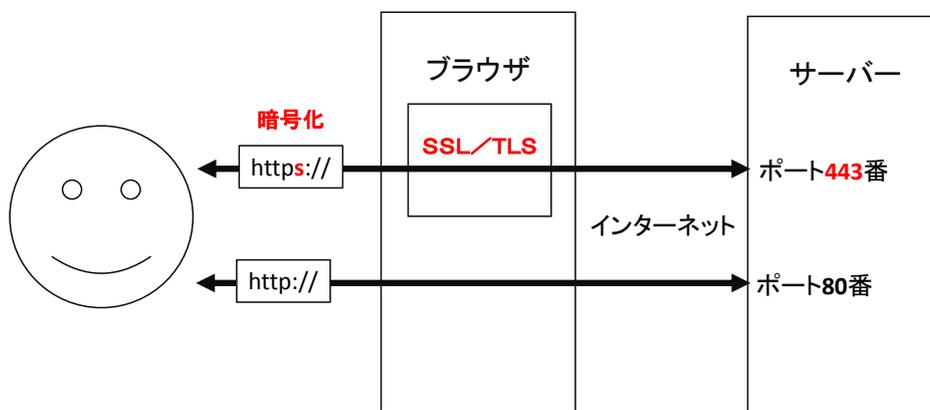
方法2；

コマンド・プロンプト窓で、「hostname」と打ち込むと



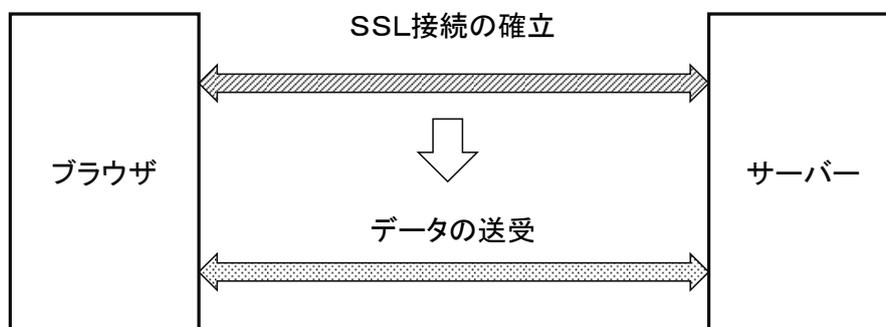
(3) HTTP と HTTPS の違い

本稿では、WebRequest()を使う上で最低必要な概念のみを述べます。
なお、本稿の localhost では「HTTP」アクセスのみを実行しています。
概念図；



- HTTP プロトコル (http://...) では、直接にサーバーと TCP/IP 通信を行います、一方 HTTPS プロトコル (https://...) では、ウェブブラウザは、自身が備えている SSL (Secure Sockets Layer) 機能を介してサーバーと TCP/IP 通信を行います。
- サーバー側は、HTTP は TCP のポート番号 80 番を、HTTPS では TCP のポート番号 443 番を割り当てて対応します。

通信；



- HTTPS 通信では、SSL 接続確立後にデータ（暗号化済み）の送受が始まります。（HTTP 通信は、SSL 接続をせずにそのままデータ（暗号化無し）の送受が始まります）

(4) URL エンコード

例えば、「ログイン画面」で「ユーザーID」と「パスワード」のデータ（テキストや数値）をブラウザからサーバーに送付（POST）する場合を考えます。

何通りか方法があるらしいのですが、代表的な方法としては以下の2つがあります。

- I. URL エンコード
- II. JSON 形式

[URL エンコード]

本稿の例では、「ログイン画面」と「投稿画面」で、POST するのに使います。

ログイン画面の場合；

- ・ Fiddler の POST キャプチャー画面に

「userid=keroro&password=keroro&login>Login」

と、記録されているのが該当します。

「**=データ&++=データ&XX=データ」と、POST するデータが「&」で繋がれていきます。

問題は、「英数字とアルファベット」等の1バイト・データはそのまま読み取れるのですが、2バイトデータや特殊文字はURL エンコードされたものが使われます。

例； 文字コードは UTF-8 の場合

エンコード前 = 「アメンボ」

URL エンコード後 = 「%e3%82%a2%e3%83%a1%e3%83%b3%e3%83%9c」

正直、何が何だか判りません！

これが、_post_main.mq4 で投稿内容にアルファベットを使った理由です、読みやすさを優先しました。

[JSON 形式]

例えば、URL エンコード形式なら

name=アメンボ&adress=123

と、「&」で繋げて表現されるデータならば、JSON 形式では、以下の形式になります。

```
{
  "name": "アメンボ",
  "adress": 123
}
```

※ただし、実際の動作確認はしていないので「2バイト・データ」がどの様に処理されるのかは理解していません。

※バージョン2のWebRequest()関数「引数7」ではこのJSON形式が使えるのかもしれませんが、未確認です。

3. WebRequest()解析用の疑似サーバー・スクリプトと MQL4 スクリプト

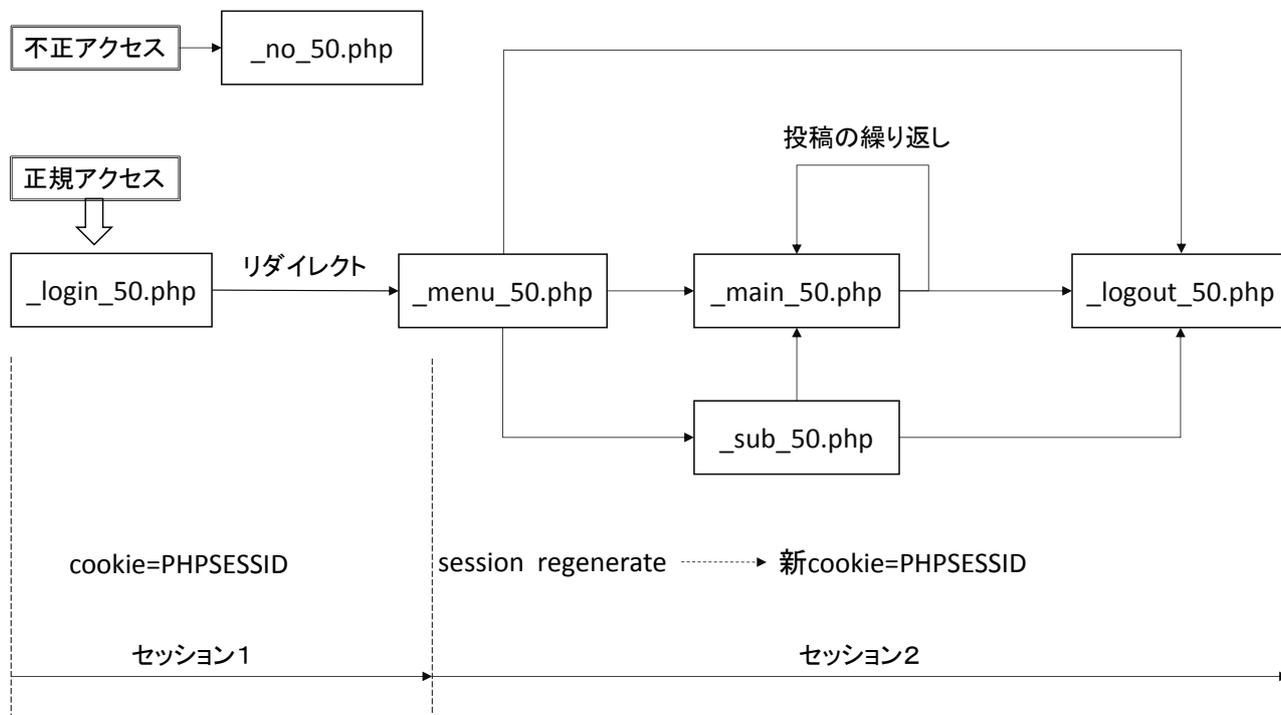
- WebRequest()の仕様（動作）を理解するには、内容が全て既知で単純化され、且つ自由に色々試してみることが可能な環境（疑似サーバー）が必須と考えました。いきなり、ターゲットとするサーバーをアクセスしても、相手の構造が判らないので、結果の解析が手に負えません。
- そこで、自身のPC内に「XAMPP」によるローカル・サーバーを設定し、「PHP」で簡単な「ログイン」システムを構築しました。

(1) ローカル・サーバーの仕様

[概要]

- ログイン（ID、パスワード必須）すると、まずメニュー画面が表示され、[投稿画面へ]を選択すると、テキスト・メッセージを投稿することが出来ます。
- ログインすることなく、いきなり [投稿画面] 等へアクセスしようとしても、不正アクセスと見做されて投稿画面にはたどり着けません。

[システム概念図]



(2) セッション ID (クッキー) のボタンタッチ

- ① 「_login_50.php」で正規の「ID、パスワード」でログインすると、セッションを開始後に「_menu_05.php」へ「リダイレクト（無条件ジャンプ）」します。
(通常はココで「セッション」を再発行するのですが、本稿では都合によりしません)
- ② 「_menu_05.php」では「セッション」を再発行（リジェネレート）をしています。
この発行した「セッション ID (クッキー)」を後工程「_main_05.php、_sub_05.php、_logout_05.php」に次々とボタンタッチしていき、一つのセッションとしてまとめます。

※上記の「セッション ID (クッキー)」を伴わないアクセスは、不正アクセスと解釈します。

(3) スクリプト・サンプル (php コード) 概要

「疑似サーバー用 ; PHP スクリプト_05[セッション版]セット」フォルダー内

名前	更新日時	種類	サイズ
_login_05.php	2015/04/06 20:23	PHP ファイル	2 KB
_logout_05.php	2015/04/07 21:34	PHP ファイル	1 KB
_main_05.php	2015/04/05 17:50	PHP ファイル	3 KB
_menu_05.php	2015/04/06 22:11	PHP ファイル	1 KB
_no_05.php	2015/04/05 18:21	PHP ファイル	1 KB
_sub_05.php	2015/04/05 18:08	PHP ファイル	1 KB

PHP スクリプトは、「セッション版」と「セッション無し」版を用意しました。

フォルダー名以外は全く同一名にしてあるので、ご注意ください。(詳細は「第9章」を参照)

セッション版のホルダー

= 「疑似サーバー用 ; PHP スクリプト_05[セッション版]セット」

セッション無し版のホルダー

= 「疑似サーバー用 ; PHP スクリプト_05[セッション無し]版セット」

また、HTTP アクセスを実行する場合は下記の「[keil](#)」部分を、読者が実際に試すPCのマシン名に書き直してください。(読者の環境では、もしかすると「localhost」で動くかも)

```
string local_url="http://keil/_main_05.php";
```

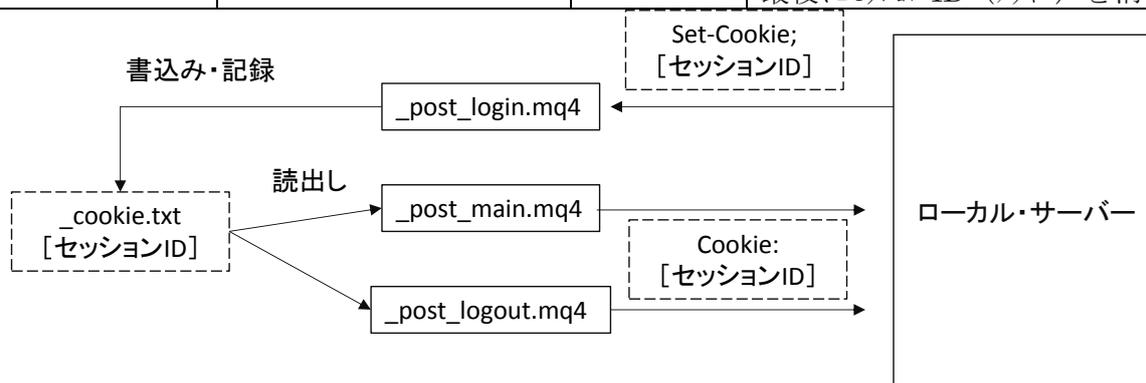
(4) MQL 4 スクリプト概要

「_post_login.mq4、_post_main.mq4、_post_logput.mq4」の3つ(スクリプト)を準備しました。

何れも「バージョン1 ; 9個の引数」の WebRequest()による POST メソッドの確認用。

※動作の詳細は「第9章」のコードを参照ください。

MQL4 スクリプト	用途	セッション	動作補足
_post_login.mq4	ログインに使用します	開始する	サーバーのレスポンス・ヘッダーから、セッション ID (クッキー) を読み出し、_cookie.txt に保存する
_post_main.mq4	メッセージ投稿用です	継続する	_cookie.txt からセッション ID (クッキー) を読み出して、サーバーへ送付する。
_post_logput.mq4	ログアウトに使用します	終了する	_cookie.txt からセッション ID (クッキー) を読み出して、サーバーへ送付し、最後にセッション ID (クッキー) を消去。

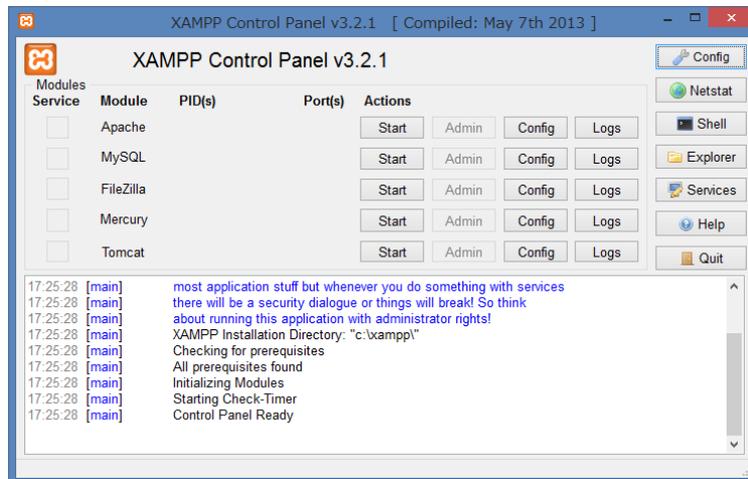


4. 動作・解析環境の準備

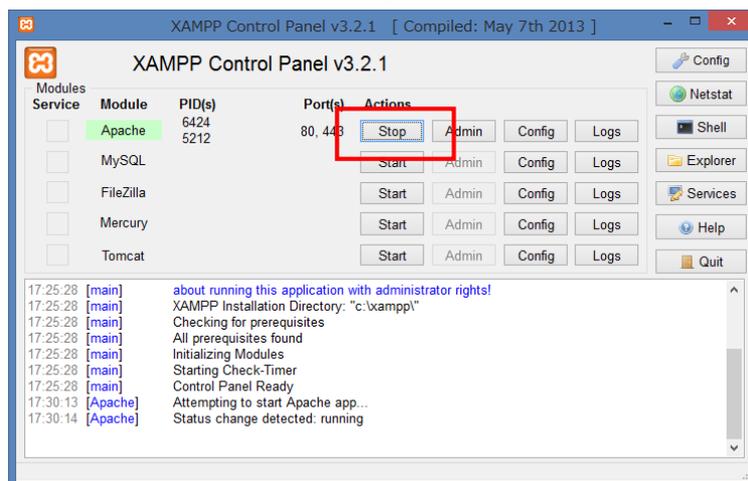
- 以下の手順で `WebRequest()` の機能やセッションの意味を理解していきます。

(詳細は「`WebRequest()`の使い方 (その1)」を参照ください)

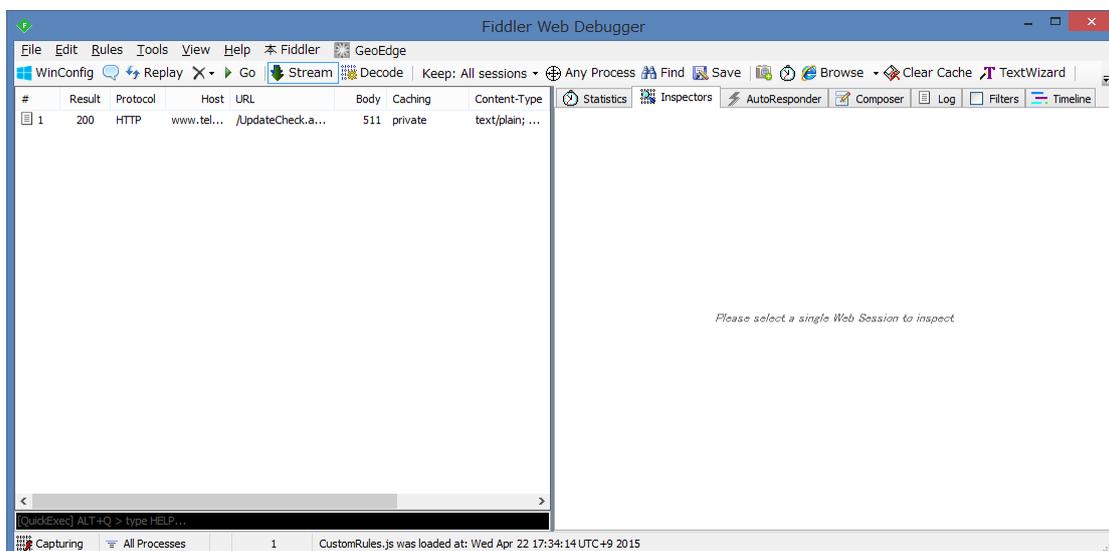
[ステップ1]; XAMPP (ローカル・サーバー) を立ち上げます。



[Apache] の [Start] をクリックして [Stop] を表示させます



[ステップ2]; Fiddler (バイオリン弾き ; プロキシ・サーバー) を立ち上げます



[ステップ 3];

PHP スクリプト_05[セッション版]セット」一式;

名前	更新日時	種類	サイズ
MQL4スクリプト	2015/04/17 0:49	ファイル フォルダー	
疑似サーバー用; PHPスクリプト_05[セッション版]セット	2015/04/17 0:37	ファイル フォルダー	
疑似サーバー用; PHPスクリプト_05[セッション無し]版セット	2015/04/18 0:38	ファイル フォルダー	
最初に考えたPHPスクリプト_04[セッション版]セット	2015/04/24 19:37	ファイル フォルダー	

名前	更新日時	種類	サイズ
_login_05.php	2015/04/06 20:23	PHP ファイル	2 KB
_logout_05.php	2015/04/07 21:34	PHP ファイル	1 KB
_main_05.php	2015/04/05 17:50	PHP ファイル	3 KB
_menu_05.php	2015/04/06 22:11	PHP ファイル	1 KB
_no_05.php	2015/04/05 18:21	PHP ファイル	1 KB
_sub_05.php	2015/04/05 18:08	PHP ファイル	1 KB

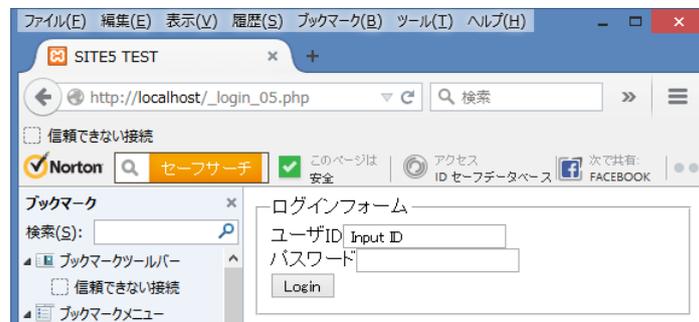
上記を、xampp 直下の htdocs ホルダーにコピーします。

5. ブラウザで「サンプル php コード (セッション版)」をアクセスしてみる

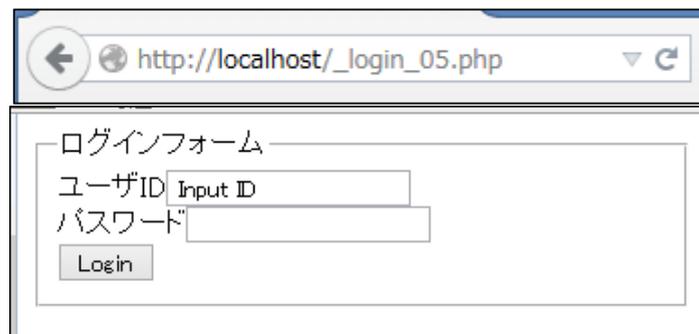
・PHP スクリプト_05[セッション版]セット」を動作させてみます。

(1) 「ログイン画面」から順にアクセスする

[ステップ 1]; 「localhost/_login_05.php」をアクセスします



拡大;



[ステップ 2]; ユーザー ID とパスワードを入力します

- ・ユーザー ID = keroro
- ・パスワード = keroro

※ 「keroro」は左詰で入力してください、「InputID」の左に半角スペースがあるのに注意!

ログインフォーム

ユーザID

パスワード

[ステップ 3]; をクリック ⇒ 「_menu_05.php」へジャンプ (セッション開始)

<メニュー画面>

以下のメニューから、選択してください;

- [投稿画面へ](#)
- [予備画面へ](#)
- [ログアウト](#)

チェック:USER_ID= keroro です

[ステップ 4]; 「[投稿画面へ](#)」をクリック ⇒ 「_main_05.php」へジャンプ (セッション継続)

<掲示板>

お名前:

ご意見:

- [ログアウト](#)

[ステップ5]; テキストで投稿します

<掲示板>

お名前:

ご意見:

読者の皆さん、
お元気ですか？

アメンボは元気です

Toukou_suru

• [ログアウト](#)

[ステップ6]; をクリックします ⇒ 投稿内容が保存・表示されます

<掲示板>

お名前:

ご意見:

Toukou_suru

• [ログアウト](#)

投稿者:アメンボ

内容:

読者の皆さん、
お元気ですか？

アメンボは元気です

※更に「投稿」を続けると、次々に登録・表示されます。

The image shows two side-by-side screenshots of a forum interface. The left screenshot is a post view with the title '<掲示板>', the author 'ケロロ軍曹であります', and the content 'ご意見: タママ、ドロロ 元気ですか?'. Below the content is a 'Toukou_suru' button and a 'ログアウト' link. An arrow points from this view to the right screenshot, which shows the post details. The top section is for the author '投稿者:アメンボ' with content '読者の皆さん、お元気ですか? アメンボは元気です'. The bottom section is for the poster '投稿者:ケロロ軍曹であります' with content 'タママ、ドロロ 元気ですか?'.

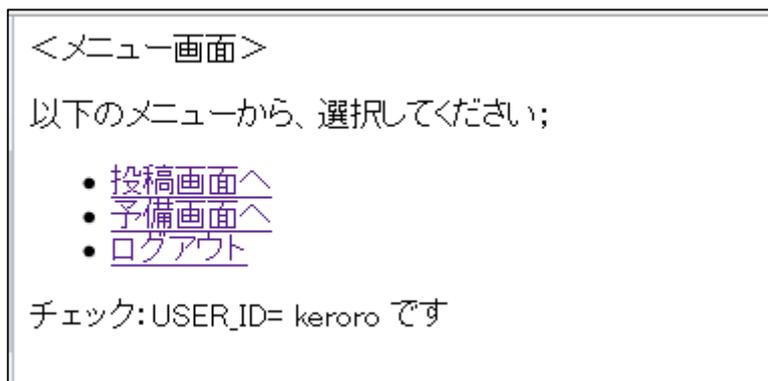
[ステップ7]; **ログアウト**をクリックします ⇒ 「_logout_05.php」へ (セッション終了)

The image shows a browser address bar with the URL 'http://localhost/_logout_05.php'. Below it is a confirmation message box that says '正常にログアウトしました。' followed by two links: 'ログイン画面に戻る' and 'いつものホームへ'.

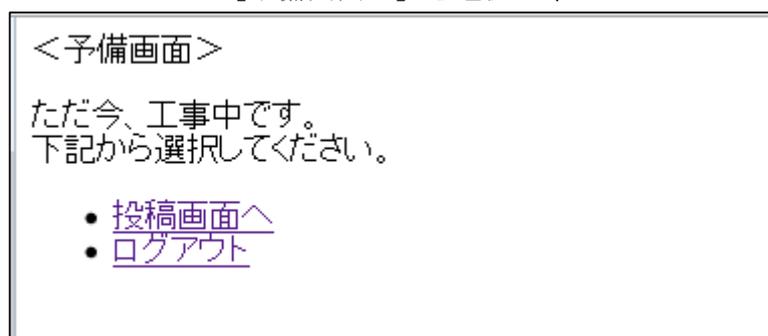
※ログイン画面で、IDかパスワードを間違えた場合; 下記の画面が表示されます

The image shows a login form titled 'ログインフォーム' with an error message: 'ユーザID、又はパスワードに誤りがあります。'. Below the message are input fields for 'ユーザID' (containing 'Input ID') and 'パスワード', and a 'Login' button.

※メニュー画面で、「予備画面」を選択した場合；



[予備画面へ] を選択 ↓

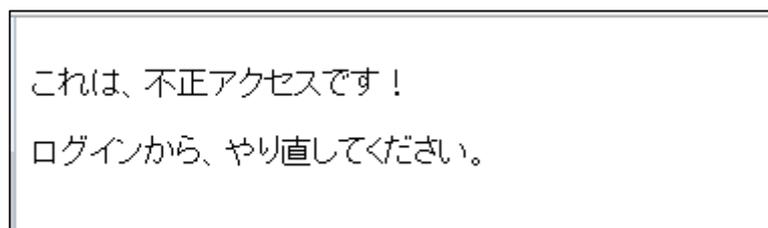
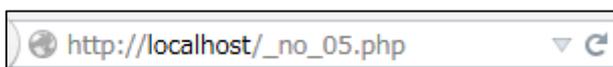


(2) ブラウザで不正アクセスを試みる (セッションの意味を確認)

・「ログイン」しないで、いきなり「_main_05.php」をアクセスしてみます



↓



※「セッション」が確立されていないので、不正なアクセスと判定されます。

6. 「サンプル php コード (セッション無し版)」で、不正アクセスしてみる

[ステップ1]; PHP スクリプト_05 を、[セッション無し版]セット一式に入れ替えます。

名前	更新日時	種類	サイズ
MQL4スクリプト	2015/04/17 0:49	ファイル フォルダー	
疑似サーバー用; PHPスクリプト_05[セッション版]セット	2015/04/17 0:37	ファイル フォルダー	
疑似サーバー用; PHPスクリプト_05[セッション無し]版セット	2015/04/18 0:38	ファイル フォルダー	

※PHP コード一式を、xampp 直下の htdocs ホルダーにコピーし、入れ替えます。

※各 PHP コードのファイル名は、セッション版と全く同一ですので、使用後は必ず「セッション版」に戻して (入れ替えて) おいてください。

[ステップ2]; 「ログイン」を経由せずに「_main_05.php」をいきなりアクセスしてみる

<掲示板>

お名前:

ご意見:

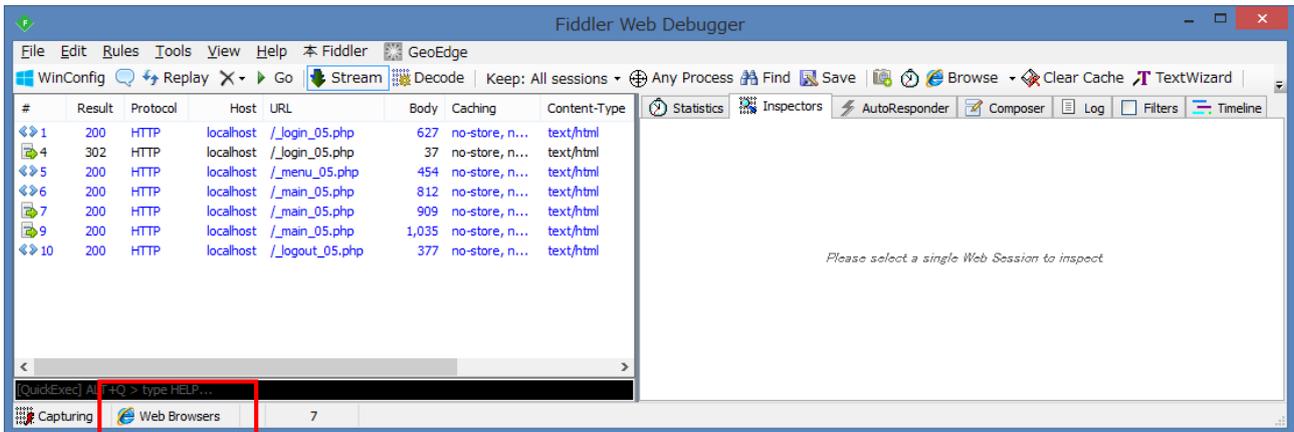
Toukou_suru

• [ログアウト](#)

・・・アクセス出来てしまいました！

7. ブラウザによる「php コード (セッション版)」アクセスを Fiddler で観察する

※ 「_login_05.php」⇒「_menu_05.php」⇒「_main_05.php」⇒「_logout_05.php」を観察する
Fiddler によるキャプチャー結果；余計なストリーム（2、3、8）は削除済み⇒番号跳びの理由



・キャプチャーする対象を [Web Browser] に指定して、なるべく余計なプロセスは省きます。

（全てをモニターする場合は、クリックして [All Processes] にしてください）

※以下、Fiddler がキャプチャーした内容を抜粋します。（アメンボが適当に選択）

(1) 初めに「localhost/_login_05.php」をアクセスしたとき

赤枠部分がキャプチャーされます。

#	Result	Protocol	Host	URL	Body	Caching	Content-Type
1	200	HTTP	localhost	/_login_05.php	627	no-store, n...	text/html
4	302	HTTP	localhost	/_login_05.php	37	no-store, n...	text/html
5	200	HTTP	localhost	/_menu_05.php	454	no-store, n...	text/html
6	200	HTTP	localhost	/_main_05.php	812	no-store, n...	text/html
7	200	HTTP	localhost	/_main_05.php	909	no-store, n...	text/html
9	200	HTTP	localhost	/_main_05.php	1,035	no-store, n...	text/html
10	200	HTTP	localhost	/_logout_05.php	377	no-store, n...	text/html

Request:

[Headers]

Request Headers [Raw] [Header Definitions]

GET /_login_05.php HTTP/1.1

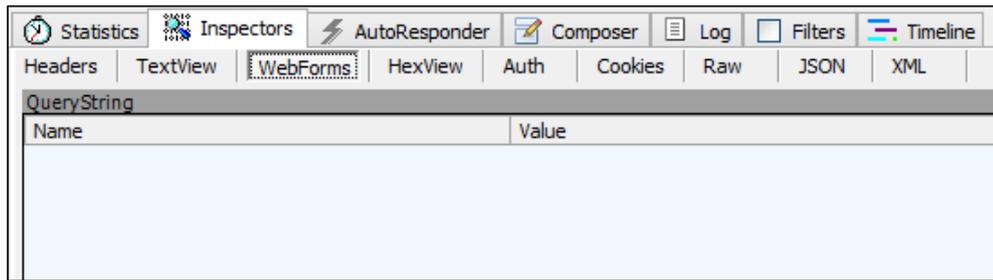
Client

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
- Accept-Encoding: gzip, deflate
- Accept-Language: ja,en-US;q=0.7,en;q=0.3
- User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0

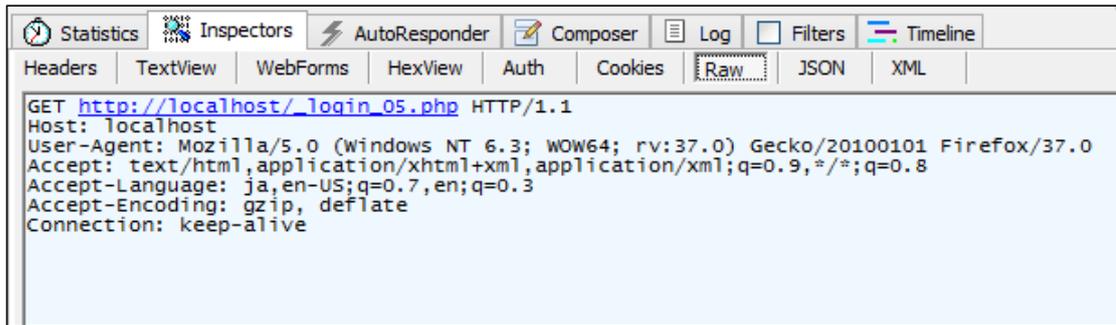
Transport

- Connection: keep-alive
- Host: localhost

[WebForms]

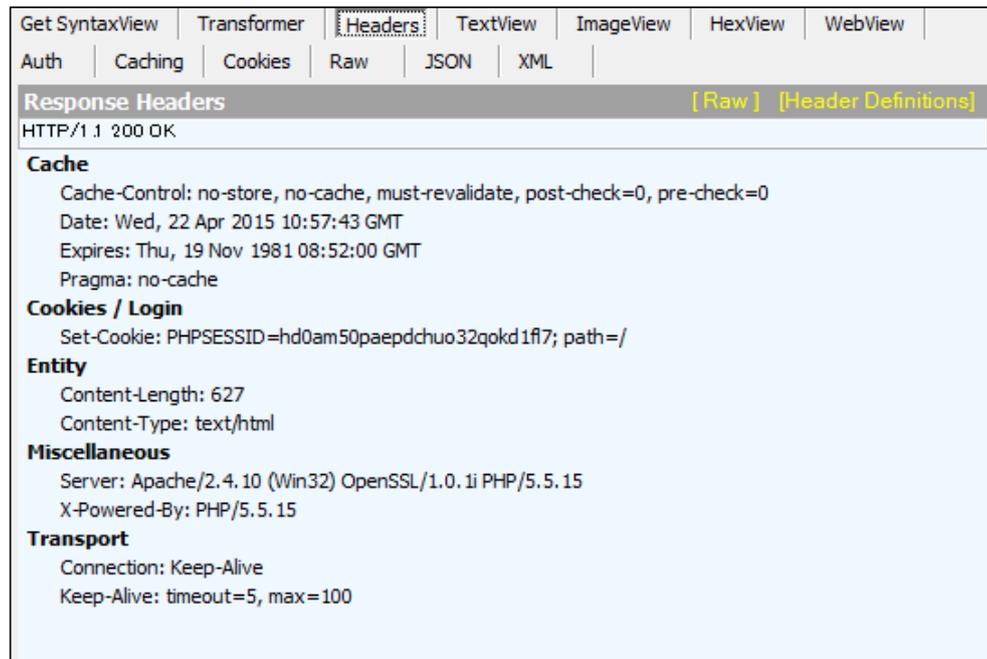


[Raw]



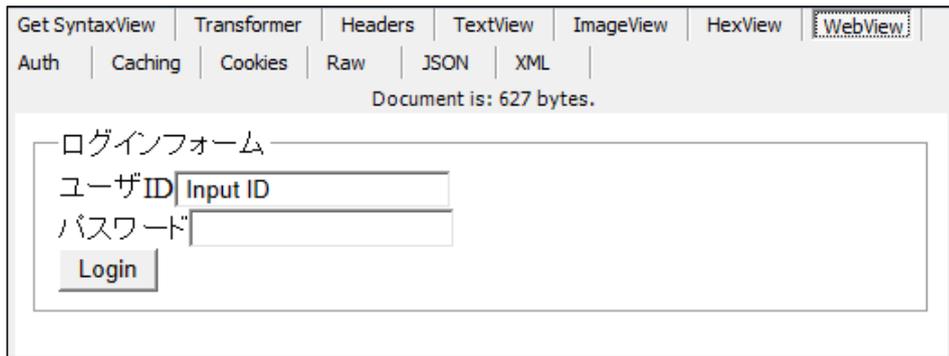
Response;

[Headers]



※ 「Set-Cookei」部；サーバー側でセッション ID が発行され、クッキーとしてブラウザ側に送信されています。

[WebView]



[Raw]



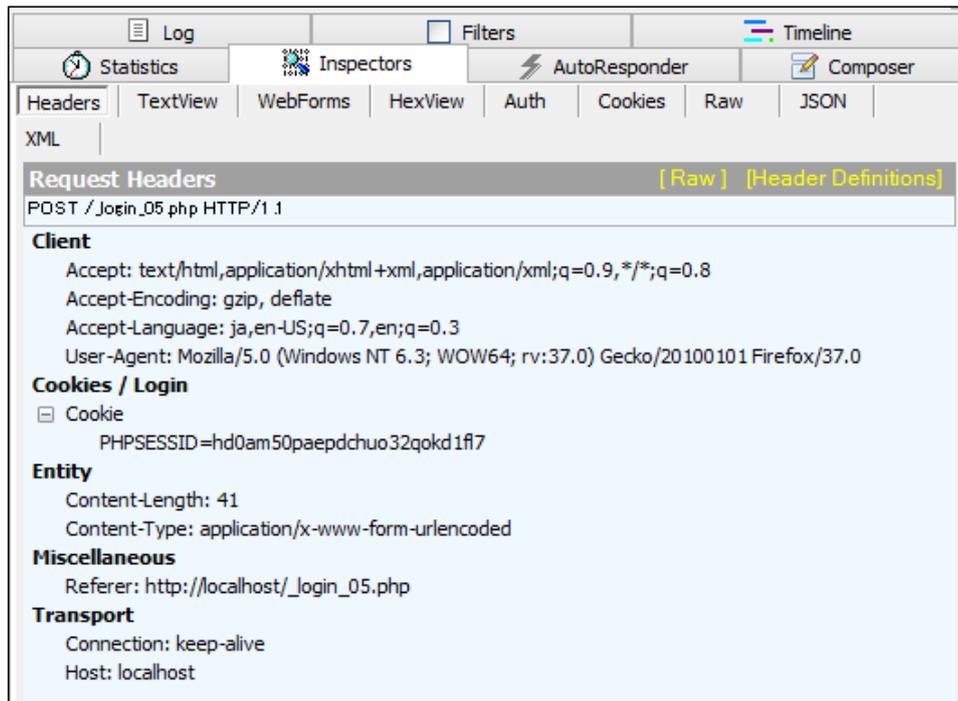
(2) 「ID」と「パスワード」を入力し、**Login**を選択したとき

#	Result	Protocol	Host	URL	Body	Caching	Content-Type
1	200	HTTP	localhost	/_login_05.php	627	no-store, n...	text/html
4	302	HTTP	localhost	/_login_05.php	37	no-store, n...	text/html
5	200	HTTP	localhost	/_menu_05.php	454	no-store, n...	text/html
6	200	HTTP	localhost	/_main_05.php	812	no-store, n...	text/html
7	200	HTTP	localhost	/_main_05.php	909	no-store, n...	text/html
9	200	HTTP	localhost	/_main_05.php	1,035	no-store, n...	text/html
10	200	HTTP	localhost	/_logout_05.php	377	no-store, n...	text/html

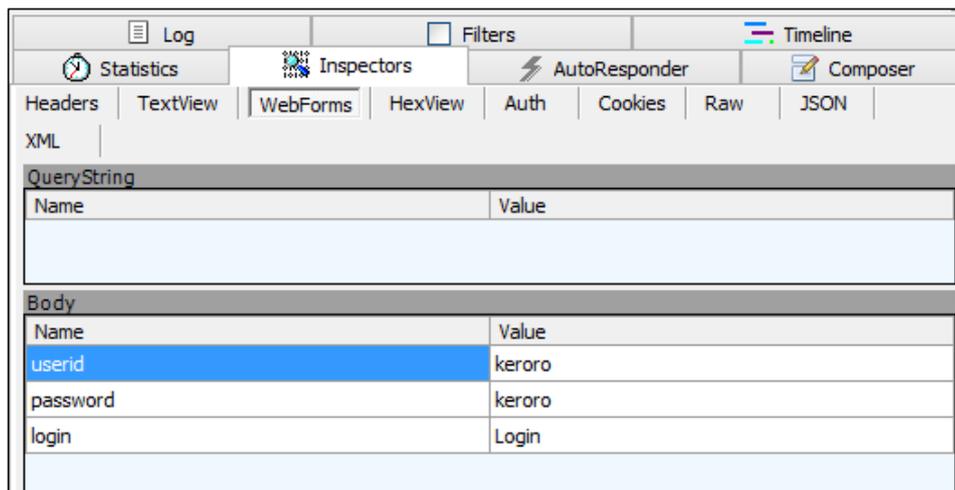
4

Request;

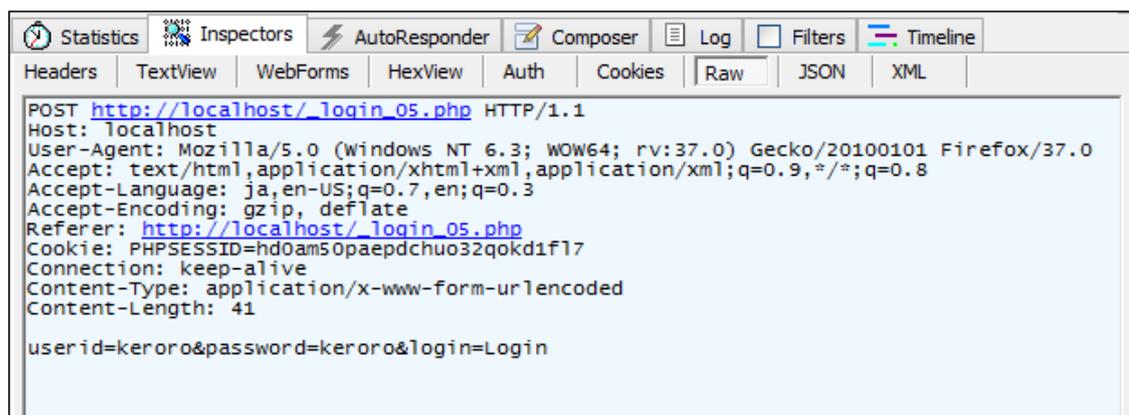
[Headers]



[WebForms]



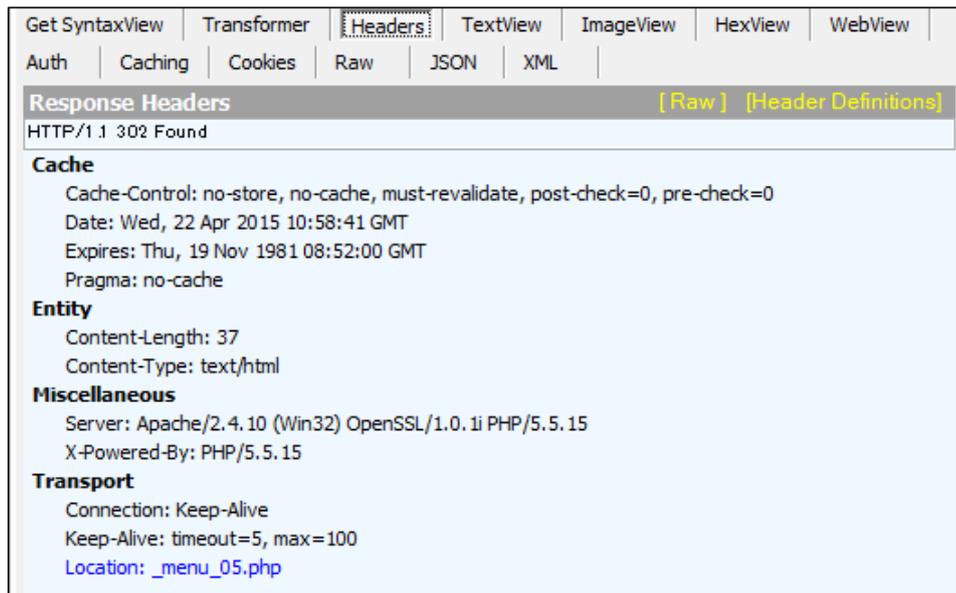
[Raw]



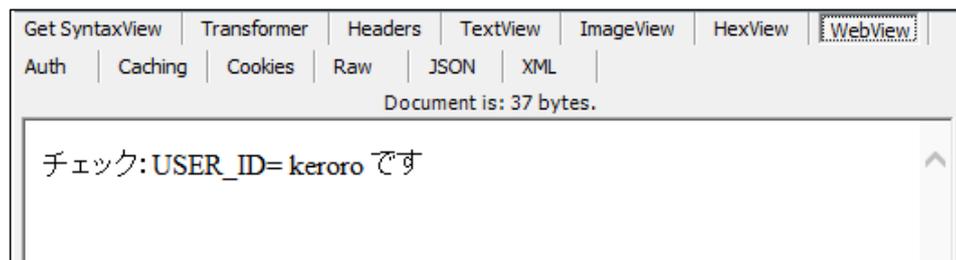
※ 「userid=keroro&password=keroro&login>Login」 の形式でフォーム・データを送信しています。

Response;

[Headers]



[WebView]



[Raw]



※一行目に「HTTP/1.1 302 Found」と表示されています。

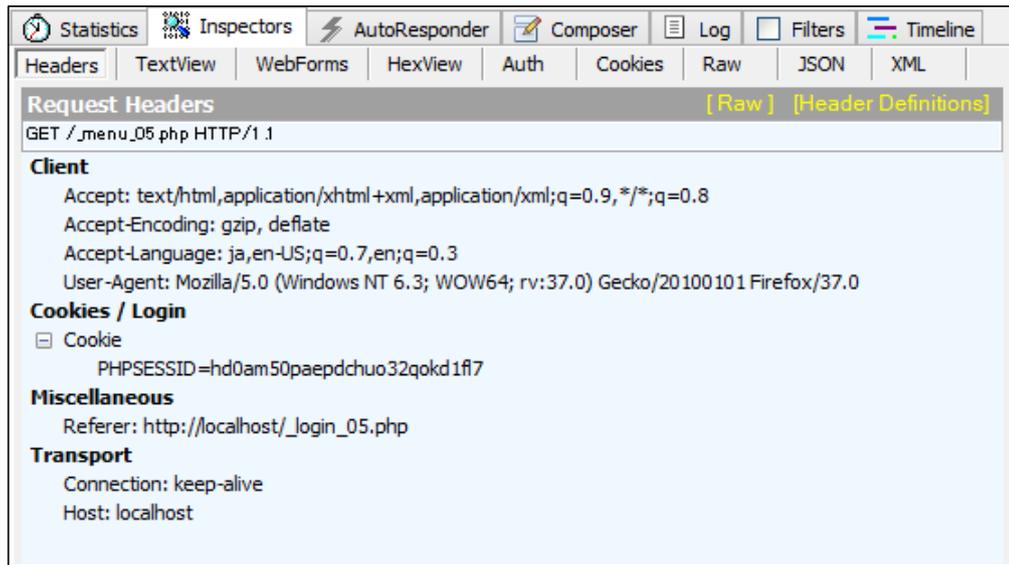
「302」とは「リダイレクト」していることを示しています、詳細は読者にて調べてみてください。

「(その1)」で解説するのを忘れていたのに気が付きました)

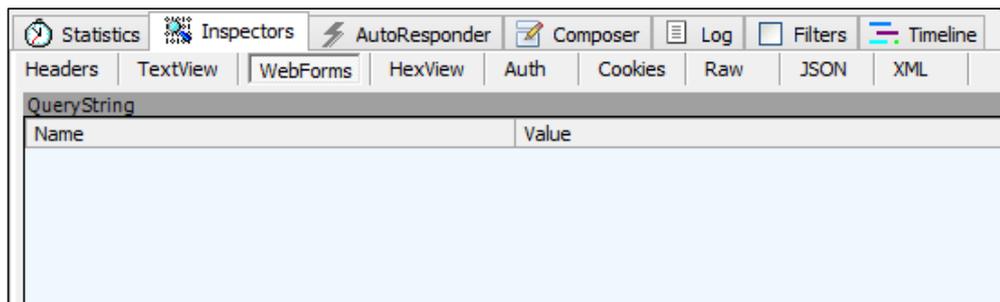
5

Request:

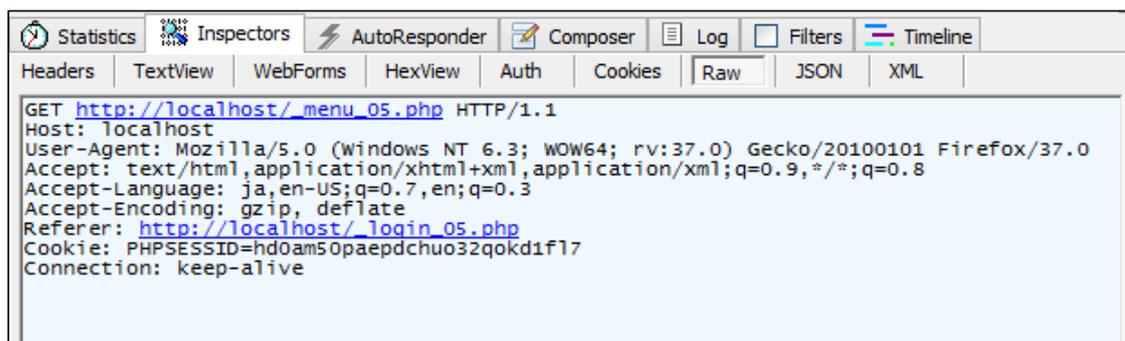
[Headers]



[WebForms]



[Raw]



Response;

[Headers]

Get SyntaxView | Transformer | **Headers** | TextView | ImageView | HexView | WebView

Auth | Caching | Cookies | Raw | JSON | XML

Response Headers [Raw] [Header Definitions]

HTTP/1.1 200 OK

Cache
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Date: Wed, 22 Apr 2015 10:58:41 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache

Cookies / Login
Set-Cookie: PHPSESSID=63dbf653qu9abbkg7b3t3jmho1; path=/

Entity
Content-Length: 454
Content-Type: text/html

Miscellaneous
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
X-Powered-By: PHP/5.5.15

Transport
Connection: Keep-Alive
Keep-Alive: timeout=5, max=99

[WebView]

Get SyntaxView | Transformer | Headers | TextView | ImageView | HexView | **WebView**

Auth | Caching | Cookies | Raw | JSON | XML

Document is: 454 bytes.

<メニュー画面>

以下のメニューから、選択してください;

- [投稿画面へ](#)
- [予備画面へ](#)
- [ログアウト](#)

チェック: USER_ID= keroro です

[Raw]

```

Get SyntaxView | Transformer | Headers | TextView | ImageView | HexView | WebView | Auth | Caching
Cookies | Raw | JSON | XML
HTTP/1.1 200 OK
Date: Wed, 22 Apr 2015 10:58:41 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
X-Powered-By: PHP/5.5.15
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Set-Cookie: PHPSESSID=63dbf653qu9abbkg7b3t3jmho1; path=/
Content-Length: 454
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: text/html

<html>
<head>
<meta http-equiv="Content-Type" content="charset=utf-8">
<title>SITE5 TEST</title>
</head>

<body>
<p><メニュー画面></p>

以下のメニューから、選択してください；

<ul>
<li><a href="_main_05.php">投稿画面へ</a></li>
<li><a href="_sub_05.php">予備画面へ</a></li>

```

※新たに「Set-Cookie:」でセッション ID が再発行されていることに注目ください。

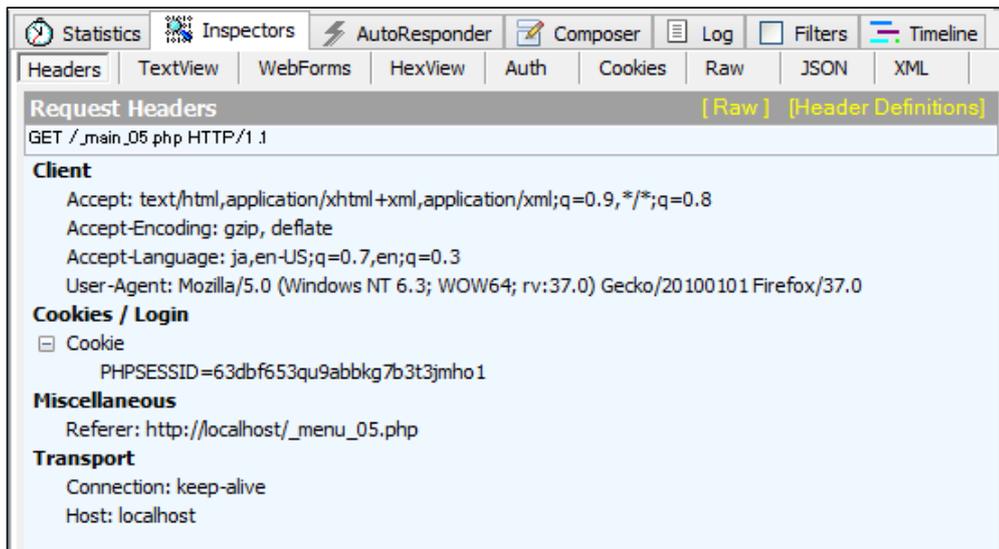
(3) メニュー画面で、[投稿画面へ] を選択したとき

#	Result	Protocol	Host	URL	Body	Caching	Content-Type
1	200	HTTP	localhost	/_login_05.php	627	no-store, n...	text/html
4	302	HTTP	localhost	/_login_05.php	37	no-store, n...	text/html
5	200	HTTP	localhost	/_menu_05.php	454	no-store, n...	text/html
6	200	HTTP	localhost	/_main_05.php	812	no-store, n...	text/html
7	200	HTTP	localhost	/_main_05.php	909	no-store, n...	text/html
9	200	HTTP	localhost	/_main_05.php	1,035	no-store, n...	text/html
10	200	HTTP	localhost	/_logout_05.php	377	no-store, n...	text/html

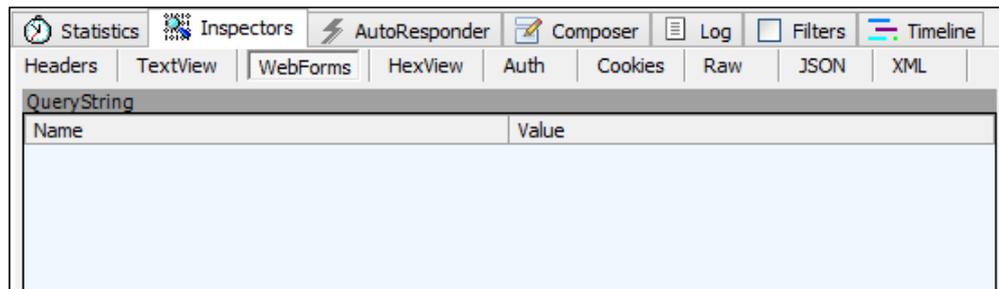
6

Request:

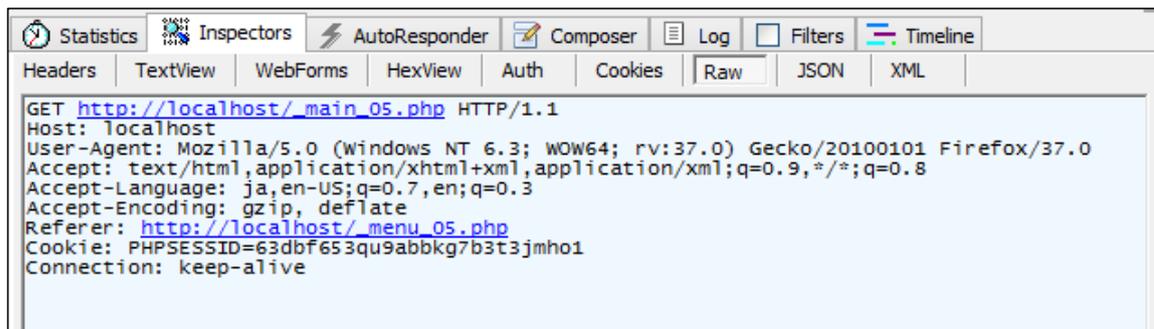
[Headers]



[WebForms]



[Raw]



Response;

[Headers]



The screenshot shows the 'Headers' tab in a web browser's developer tools. The 'Response Headers' section is expanded, displaying the following information:

- HTTP/1.1 200 OK
- Cache**
 - Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
 - Date: Wed, 22 Apr 2015 10:58:46 GMT
 - Expires: Thu, 19 Nov 1981 08:52:00 GMT
 - Pragma: no-cache
- Entity**
 - Content-Length: 812
 - Content-Type: text/html
- Miscellaneous**
 - Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
 - X-Powered-By: PHP/5.5.15
- Transport**
 - Connection: Keep-Alive
 - Keep-Alive: timeout=5, max=100

[WebView]



The screenshot shows the 'WebView' tab in a web browser's developer tools. The rendered HTML content is displayed as follows:

Document is: 812 bytes.

<掲示板>

お名前:

ご意見:

- [ログアウト](#)

[Raw]

```

Get SyntaxView | Transformer | Headers | TextView | ImageView | HexView | WebView | Auth | Caching
Cookies | Raw | JSON | XML
HTTP/1.1 200 OK
Date: Wed, 22 Apr 2015 10:58:46 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
X-Powered-By: PHP/5.5.15
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 812
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<html>
<head>
<meta http-equiv="Content-Type" content="charset=utf-8">
<title>SITE5 TEST</title>
</head>

<body>
<p><掲示板></p>

<form method="POST" action="/_main_05.php">
<p>
お名前:<input type="text" name="site_05"><br><br>
</p>
<p>
ご意見:<br>
<textarea name="contents" rows="8" cols="40"></textarea><br><br>
</p>
<input type="submit" name="bt_05" value="Toukou_suru">
</form>

<ul>
<li><a href="_logout_05.php">ログアウト</a></li>

```

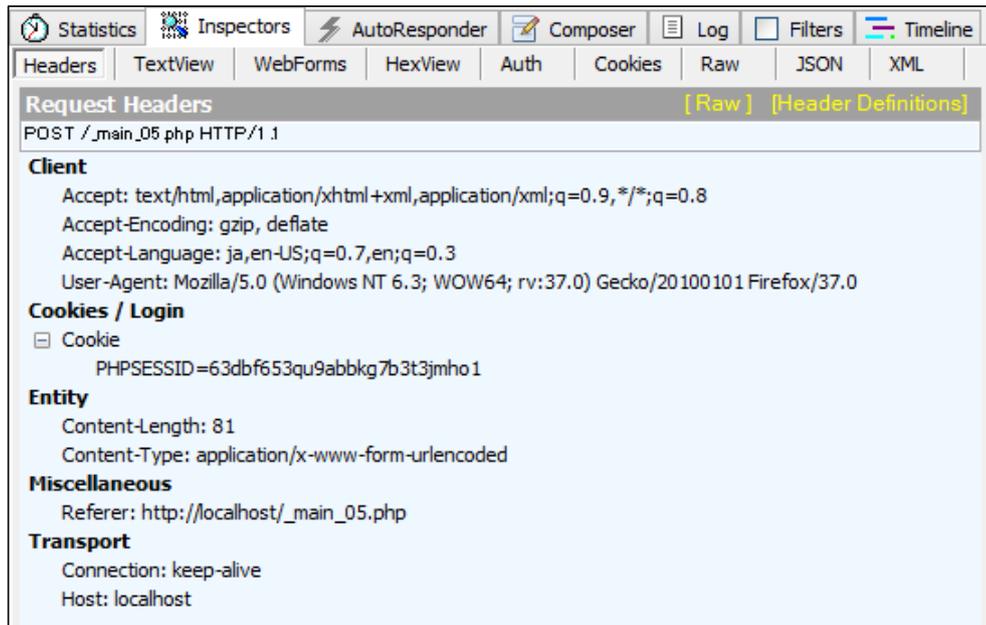
(4) 投稿をしているとき (2回、投稿した)

#	Result	Protocol	Host	URL	Body	Caching	Content-Type
1	200	HTTP	localhost	/_login_05.php	627	no-store, n...	text/html
4	302	HTTP	localhost	/_login_05.php	37	no-store, n...	text/html
5	200	HTTP	localhost	/_menu_05.php	454	no-store, n...	text/html
6	200	HTTP	localhost	/_main_05.php	812	no-store, n...	text/html
7	200	HTTP	localhost	/_main_05.php	909	no-store, n...	text/html
9	200	HTTP	localhost	/_main_05.php	1,035	no-store, n...	text/html
10	200	HTTP	localhost	/_logout_05.php	377	no-store, n...	text/html

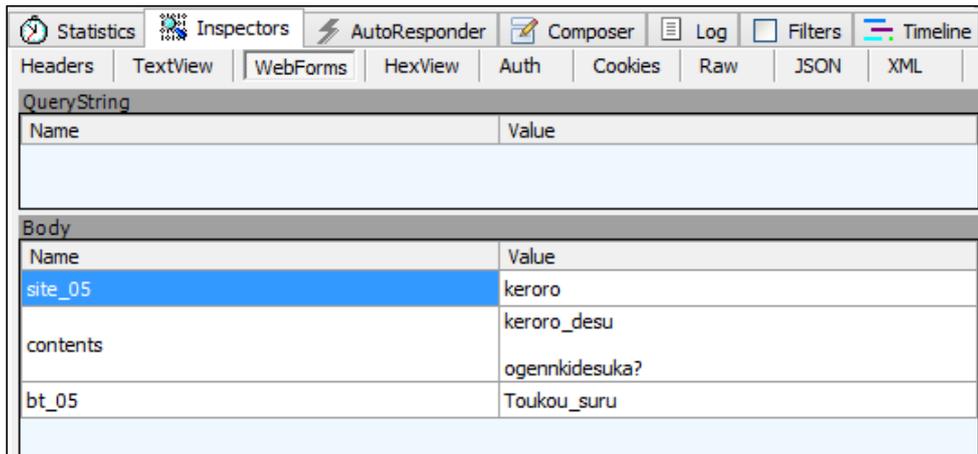
7

Request:

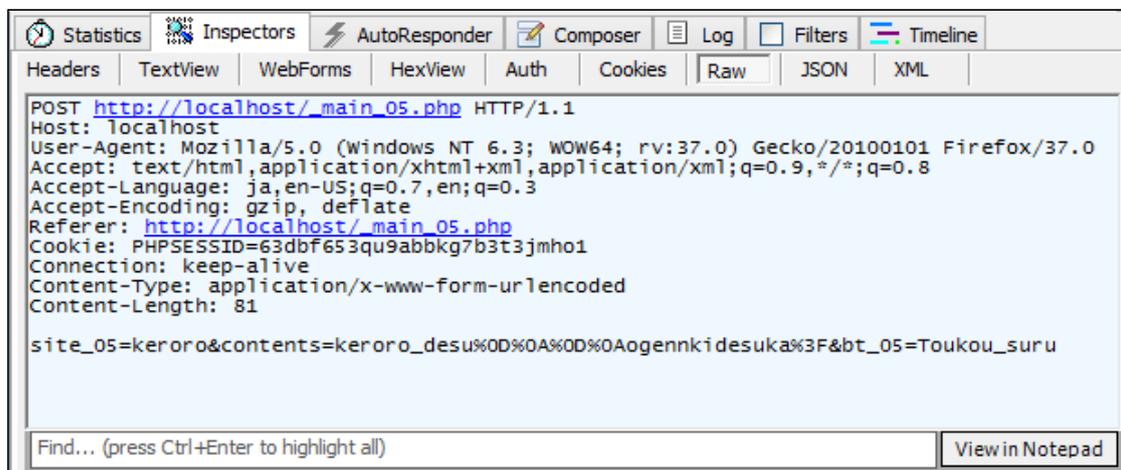
[Headers]



[WebForms]



[Raw]



Response;

[Headers]

Get SyntaxView | Transformer | **Headers** | TextView | ImageView | HexView | WebView

Auth | Caching | Cookies | Raw | JSON | XML

Response Headers [Raw] [Header Definitions]

HTTP/1.1 200 OK

Cache
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Date: Wed, 22 Apr 2015 10:59:19 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache

Entity
Content-Length: 909
Content-Type: text/html

Miscellaneous
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
X-Powered-By: PHP/5.5.15

Transport
Connection: Keep-Alive
Keep-Alive: timeout=5, max=100

[WebView]

Get SyntaxView | Transformer | Headers | TextView | ImageView | HexView | **WebView**

Auth | Caching | Cookies | Raw | JSON | XML

Document is: 909 bytes.

<掲示板>

お名前:

ご意見:

Toukou_suru

- [ログアウト](#)

投稿者: keroro

内容:

keroro_desu
ogennkidesuka?

[Raw]

```
Get SyntaxView | Transformer | Headers | TextView | ImageView | HexView | WebView | Auth |
Caching | Cookies | Raw | JSON | XML |
HTTP/1.1 200 OK
Date: Wed, 22 Apr 2015 10:59:19 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
X-Powered-By: PHP/5.5.15
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 909
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

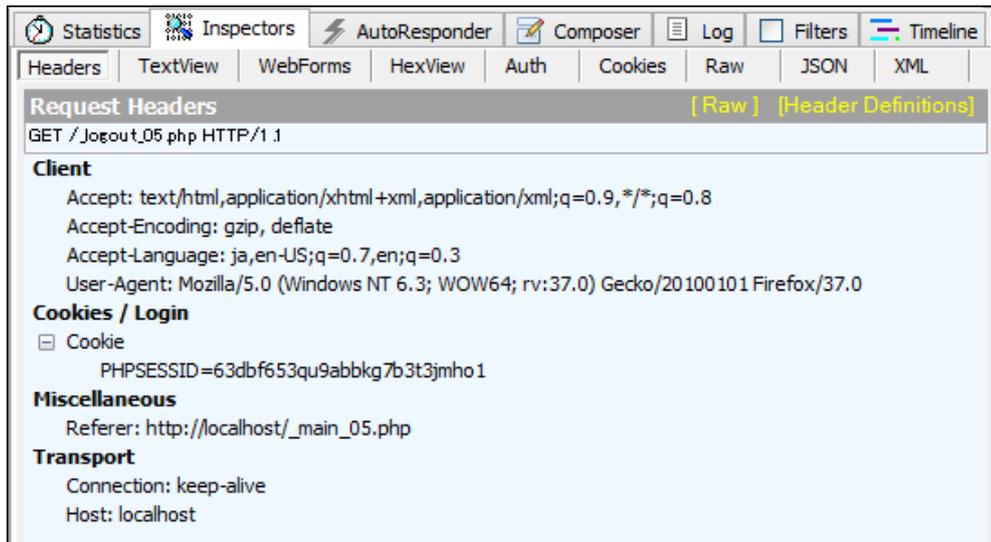
<html>
<head>
<meta http-equiv="Content-Type" content="charset=utf-8">
<title>SITES TEST</title>
</head>
<body>
<p><掲示板></p>
<form method="POST" action="/_main_05.php">
<p>
お名前:<input type="text" name="site_05"><br><br>
</p>
<p>
ご意見:<br>
<textarea name="contents" rows="8" cols="40"></textarea><br><br>
</p>
<input type="submit" name="bt_05" value="Toukou_suru">
</form>
<ul>
<li><a href="_logout_05.php">ログアウト</a></li>
```

※「# 9」は省略

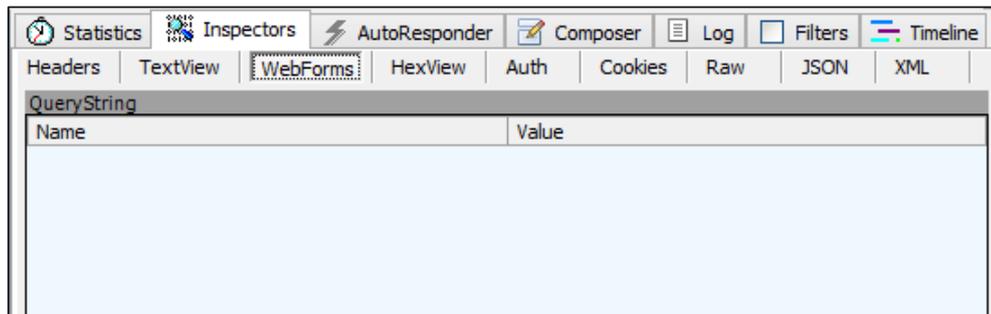
(5) 「ログアウト」を選択したとき

#	Result	Protocol	Host	URL	Body	Caching	Content-Type
1	200	HTTP	localhost	/_login_05.php	627	no-store, n...	text/html
4	302	HTTP	localhost	/_login_05.php	37	no-store, n...	text/html
5	200	HTTP	localhost	/_menu_05.php	454	no-store, n...	text/html
6	200	HTTP	localhost	/_main_05.php	812	no-store, n...	text/html
7	200	HTTP	localhost	/_main_05.php	909	no-store, n...	text/html
9	200	HTTP	localhost	/_main_05.php	1,035	no-store, n...	text/html
10	200	HTTP	localhost	/_logout_05.php	377	no-store, n...	text/html

Request:
[Headers]



[WebForms]



[Raw]



Response;

[Headers]

Get SyntaxView | Transformer | **Headers** | TextView | ImageView | HexView | WebView
Auth | Caching | Cookies | Raw | JSON | XML |

Response Headers [Raw] [Header Definitions]
HTTP/1.1 200 OK

Cache
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Date: Wed, 22 Apr 2015 11:00:14 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache

Cookies / Login
Set-Cookie: PHPSESSID=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=0; path=/
Entity
Content-Length: 377
Content-Type: text/html

Miscellaneous
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
X-Powered-By: PHP/5.5.15

Transport
Connection: Keep-Alive
Keep-Alive: timeout=5, max=100

[WebView]

Get SyntaxView | Transformer | Headers | TextView | ImageView | HexView | **WebView**
Auth | Caching | Cookies | Raw | JSON | XML |

Document is: 377 bytes.

正常にログアウトしました。

- [ログイン画面に戻る](#)
- [いつものホームへ](#)

[Raw]



```
Get SyntaxView | Transformer | Headers | TextView | ImageView | HexView | WebView | Auth | Caching
Cookies | Raw | JSON | XML
HTTP/1.1 200 OK
Date: Wed, 22 Apr 2015 11:00:14 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
X-Powered-By: PHP/5.5.15
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Set-Cookie: PHPSESSID=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=0; path=/
Content-Length: 377
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>サンプルアプリケーション</title>
  </head>
  <body>
    <div>正常にログアウトしました。</div>
    <ul>
      <li><a href="_login_05.php">ログイン画面に戻る</a></li>
      <li><a href="http://www.benri.ne.jp/">いつものホームへ</a></li>
    </ul>
  </body>
</html>
```

Find... (press Ctrl+Enter to highlight all) View in Notepad

※最後に「Set-Cookie:」でセッション ID を削除しています。(セッションを終了)

8. MT4 (MQL4) から「サンプル php コード」をアクセスする

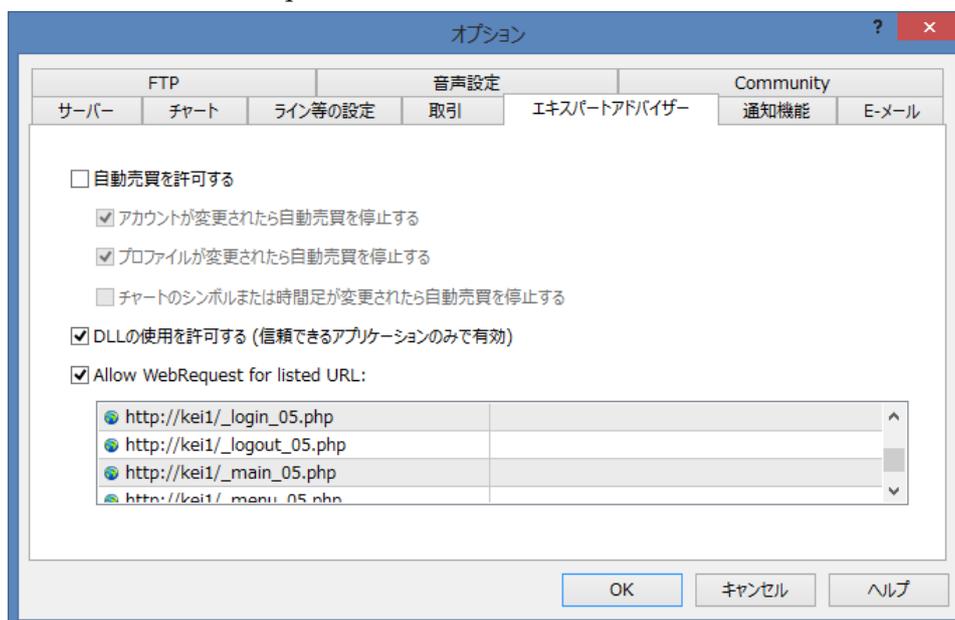
※php コードは「セッション版」を使います。

(1) 準備と確認；

[URLリストへの追加]

- ① [ツール] - [オプション] - [エキスパートアドバイザー] タブを表示させます
- ② [AllowWebRequest for listed URL:] にチェックが入っていることを確認
- ③ 下記のURLを追加していきます
 - ・「http://kei1/_login_05.php」、
 - ・「http://kei1/_main_05.php」、
 - ・「http://kei1/_logout_05.php」

(詳細な手順は、「WebRequest の使い方 (その1)」を参照ください)



※「localhost」ではなくて、「kei1」（マシン名）を使ったのは、小生の環境では WebRequest()による XAMPP サーバーへのアクセスが、「localhost」では Fiddler によるキャプチャーが出来なかったためです。(詳細は2. (2) 参照)

[MQL4 スクリプトのアクセス順]

セッションを利用するため、以下の順番でアクセスします。(セッションの継続・廃棄順番)

[_post_login] ⇒ [_post_main] ⇒ [_post_logout]

[MQL4 スクリプトの機能]

基本機能1；

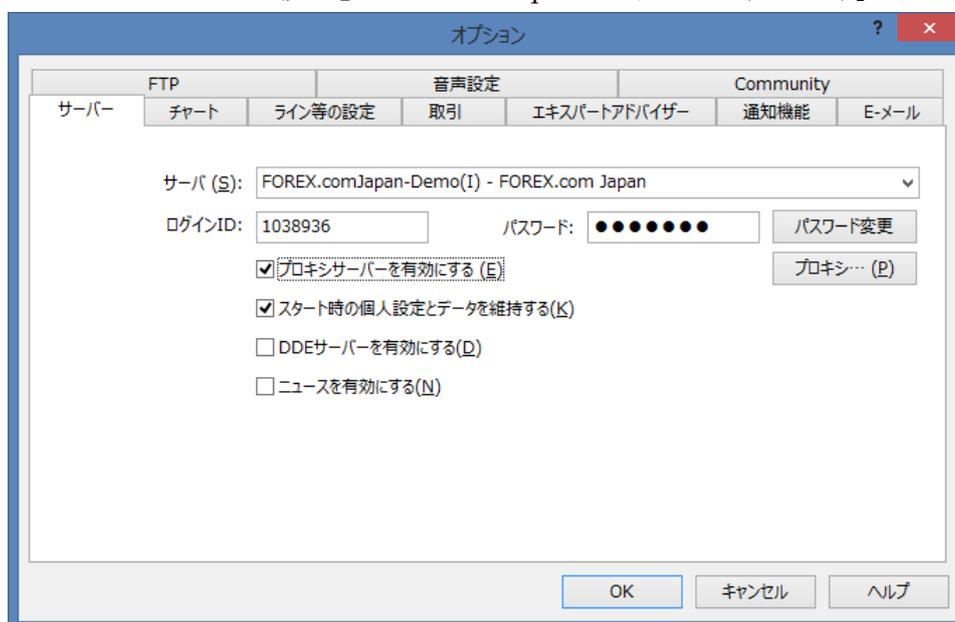
MQL4 スクリプト	アクセスする PHP	HTTP メソッド	POST する内容	セッション
_post_login.mq4	_login_05.php	POST	ID とパスワード	開始する
_post_main.mq4	_main_05.php	POST	投稿内容 (テキスト)	継続する
_post_logout.mq4	_logout_05.php	GET	—	削除する

基本機能2 ; 「リクエスト、レスポンス」ヘッダーの保存とクッキー読出し

MQL4 スクリプト	作成するファイルと記録情報内容		
	Request (リクエスト)	Response (レスポンス)	Response 中の
	メッセージ・ボディ情報	ヘッダー情報	クッキー情報
_post_login.mq4	_post_login_05.htm	_post_login_05.txt	_cookie.txt
_post_main.mq4	_post_main_05.htm	_post_main_05.txt	↑ ; 読出のみ
_post_logout.mq4	_post_logout_05.htm	_post_logout_05.txt	↑ ; 読出のみ

※ 「_post_login.mq4」は Response ヘッダー情報からセッション ID (クッキー) を検出して 「_cookie.txt」 に保存する。(保存先は[MQL4/Files]ホルダ内)
「_post_main」と「_post_logout」は、セッション ID を「_cookie.txt」から読み出して利用する。

[MT4 に Fiddler をプロキシに設定] ・ ・ 「WebRequest の使い方 (その 1)」 を参照



設定要旨

[サーバー]; 「localhost:8888」 (または「127.0.0.1 : 8888」) 「HTTP」

※ 「127.0.0.1」 はローカルホストのアドレス、

「8888」 はアクセスするプロキシ (Fiddler) のポート No を示す。

(2) 実行結果 (Fiddler の記録一覧)

① [_post_login] ⇒ ② [_post_main] ⇒ ③ [_post_logout] と実行したとき；

※以下、Fiddler の記録画面のうち、一部だけ掲載しました。(アメンボが適当に選択)

[Fiddler の結果]；

#	Result	Protocol	Host	URL	Body	Caching	Content-Type
1	302	HTTP	kei1	/_login_05.php	37	no-store, n...	text/html
2	200	HTTP	kei1	/_menu_05.php	454	no-store, n...	text/html
3	200	HTTP	kei1	/_main_05.php	658	no-store, n...	text/html
4	200	HTTP	kei1	/_logout_05.php	377	no-store, n...	text/html

['MQL4/Files'] フォルダー内に作成されたファイル]

名前	更新日時	種類	サイズ
_cookie.txt	2015/04/23 18:16	テキスト ドキュメント	1 KB
_post_login_05.htm	2015/04/23 18:16	Firefox HTML D...	1 KB
_post_login_05.txt	2015/04/23 18:16	テキスト ドキュメント	1 KB
_post_logout_05.htm	2015/04/23 18:16	Firefox HTML D...	1 KB
_post_logout_05.txt	2015/04/23 18:16	テキスト ドキュメント	1 KB
_post_main_05.htm	2015/04/23 18:16	Firefox HTML D...	1 KB
_post_main_05.txt	2015/04/23 18:16	テキスト ドキュメント	1 KB

(3) [_post_login.mq4] 実行結果

#	Result	Protocol	Host	URL	Body	Caching	Content-Type
1	302	HTTP	kei1	/_login_05.php	37	no-store, n...	text/html
2	200	HTTP	kei1	/_menu_05.php	454	no-store, n...	text/html
3	200	HTTP	kei1	/_main_05.php	658	no-store, n...	text/html
4	200	HTTP	kei1	/_logout_05.php	377	no-store, n...	text/html

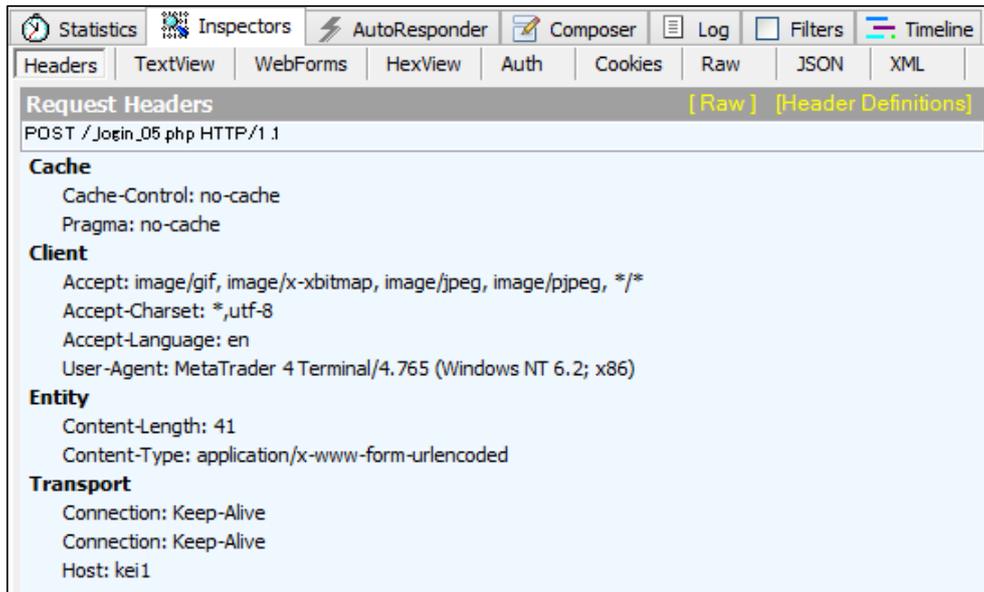
※「赤枠」内が実行・記録された内容です。

1

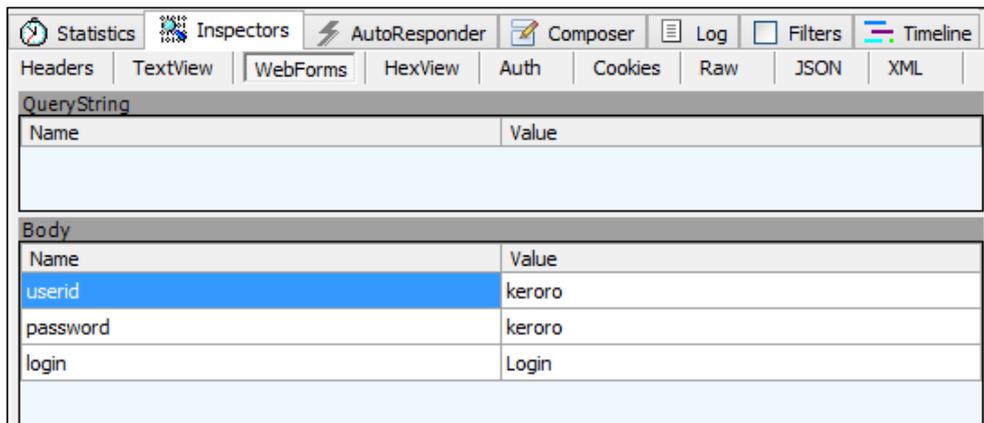
HTTP

Request:

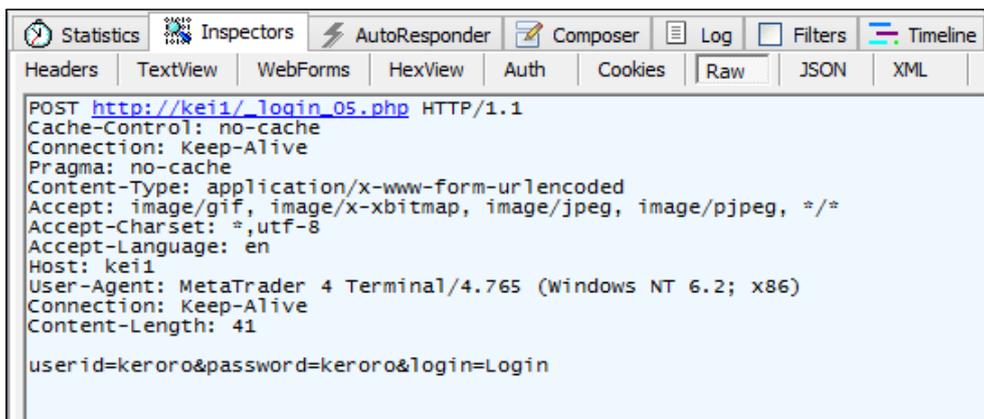
[Headers]



[WebForms]



[Raw]



※ユーザーID=「keroro」、パスワード=「keroro」が読み取れます。

Response:

[Headers]

The screenshot shows the 'Headers' tab in a web browser's developer tool. The 'Response Headers' section is expanded, showing the following information:

- Cache:** Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0; Date: Thu, 23 Apr 2015 09:16:05 GMT; Expires: Thu, 19 Nov 1981 08:52:00 GMT; Pragma: no-cache
- Cookies / Login:** Set-Cookie: PHPSESSID=nunbhurg1093eiahbbmer91hc4; path=/
- Entity:** Content-Length: 37; Content-Type: text/html
- Miscellaneous:** Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15; X-Powered-By: PHP/5.5.15
- Transport:** Connection: Keep-Alive; Keep-Alive: timeout=5, max=100; Location: [_menu_05.php](#)

[WebView]

The screenshot shows the 'WebView' tab in a web browser's developer tool. The document content is displayed as:

Document is: 37 bytes.

チェック: USER_ID= keroro です

[Raw]

The screenshot shows the 'Raw' tab in a web browser's developer tool. The raw response data is displayed as follows:

```
HTTP/1.1 302 Found
Date: Thu, 23 Apr 2015 09:16:05 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
X-Powered-By: PHP/5.5.15
Set-Cookie: PHPSESSID=nunbhurg1093eiahbbmer91hc4; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: _menu_05.php
Content-Length: 37
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

チェック: USER_ID= keroro です
```

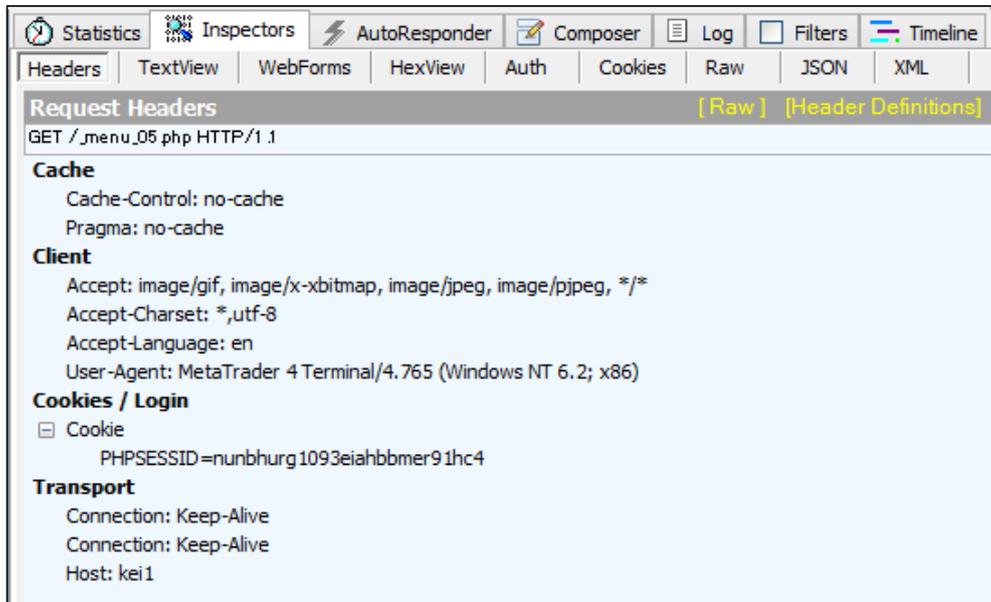
=====

2

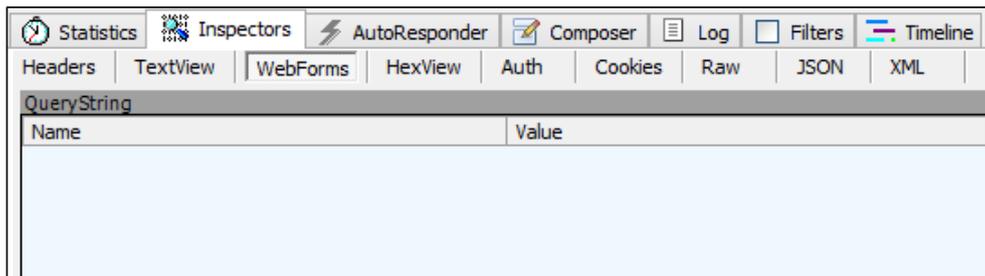
HTTP

Request:

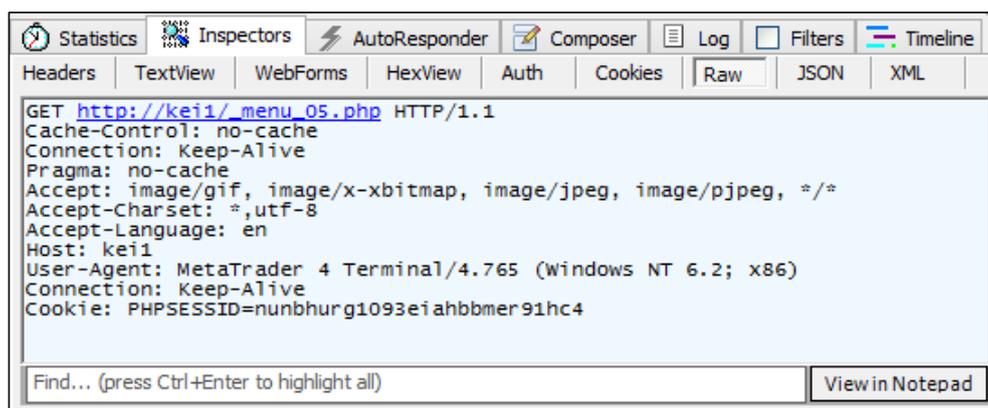
[Headers]



[WebForms]

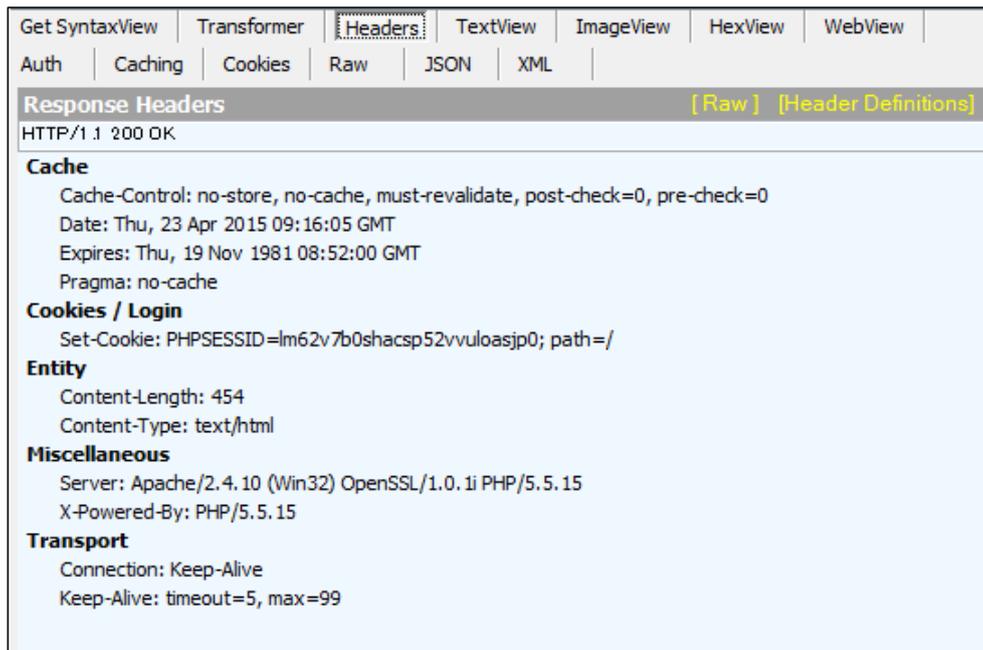


[Raw]



Response:

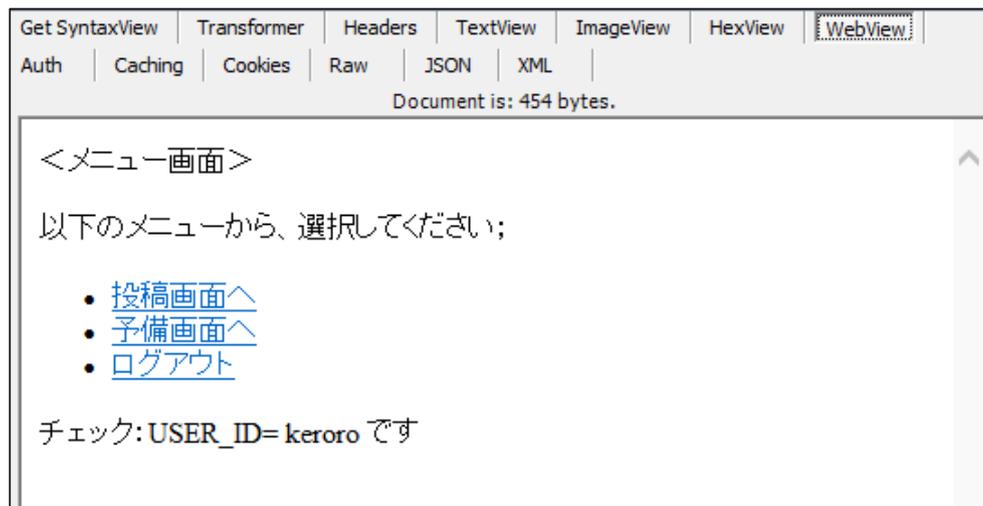
[Headers]



The screenshot shows the 'Headers' tab in a web browser's developer tools. The 'Response Headers' section is expanded, displaying the following information:

- Response Headers** [Raw] [Header Definitions]
HTTP/1.1 200 OK
- Cache**
 - Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
 - Date: Thu, 23 Apr 2015 09:16:05 GMT
 - Expires: Thu, 19 Nov 1981 08:52:00 GMT
 - Pragma: no-cache
- Cookies / Login**
 - Set-Cookie: PHPSESSID=lm62v7b0shacsp52vvuloasjp0; path=/
- Entity**
 - Content-Length: 454
 - Content-Type: text/html
- Miscellaneous**
 - Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
 - X-Powered-By: PHP/5.5.15
- Transport**
 - Connection: Keep-Alive
 - Keep-Alive: timeout=5, max=99

[WebView]



The screenshot shows the 'WebView' tab in a web browser's developer tools. The document content is displayed as follows:

Document is: 454 bytes.

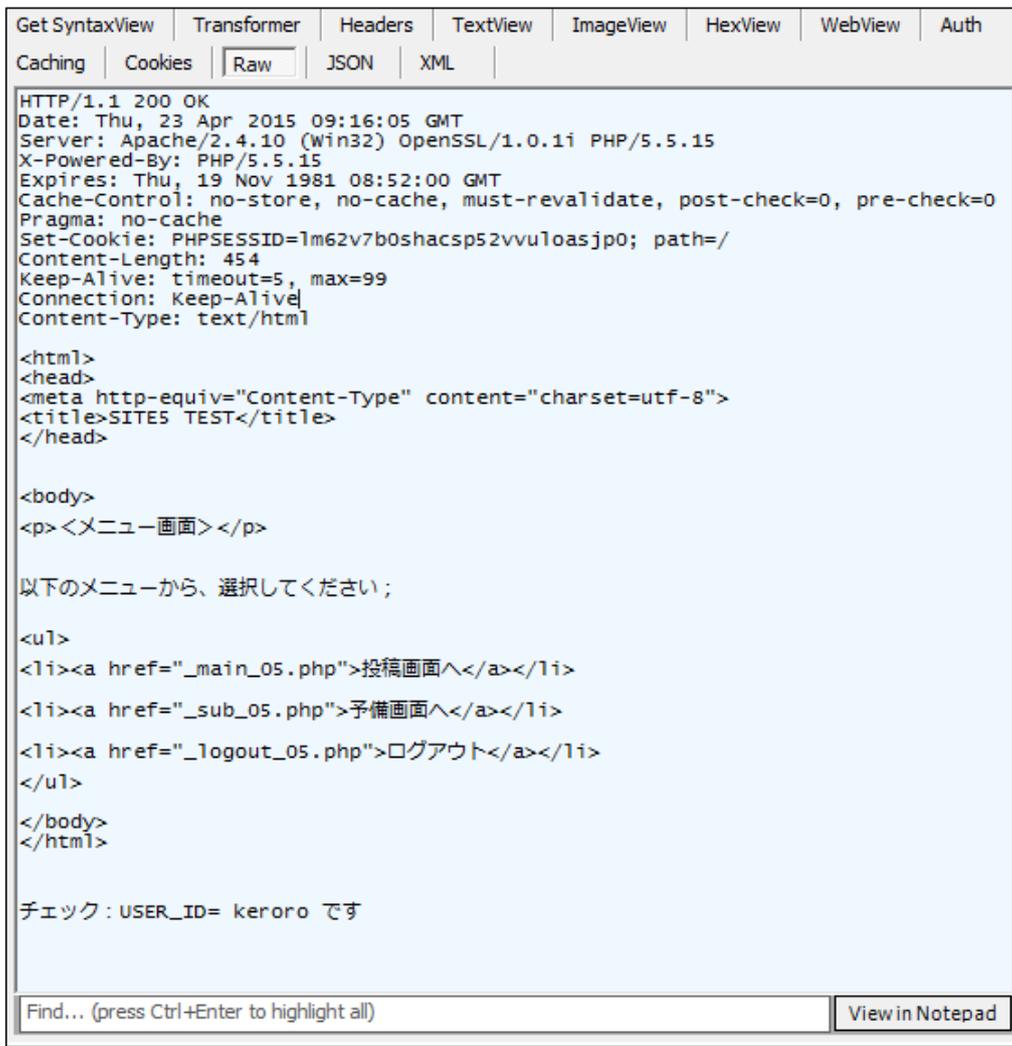
<メニュー画面>

以下のメニューから、選択してください;

- [投稿画面へ](#)
- [予備画面へ](#)
- [ログアウト](#)

チェック: USER_ID= keroro です

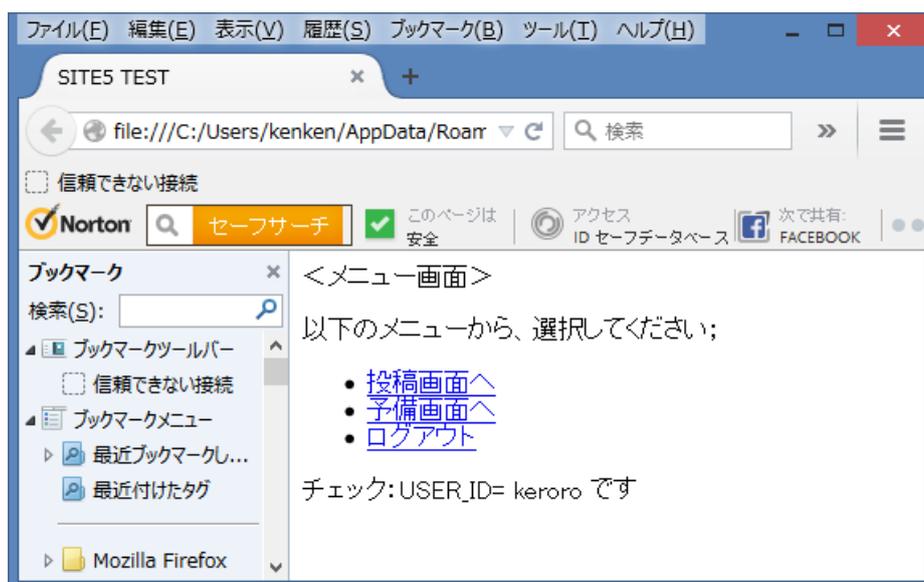
[Raw]



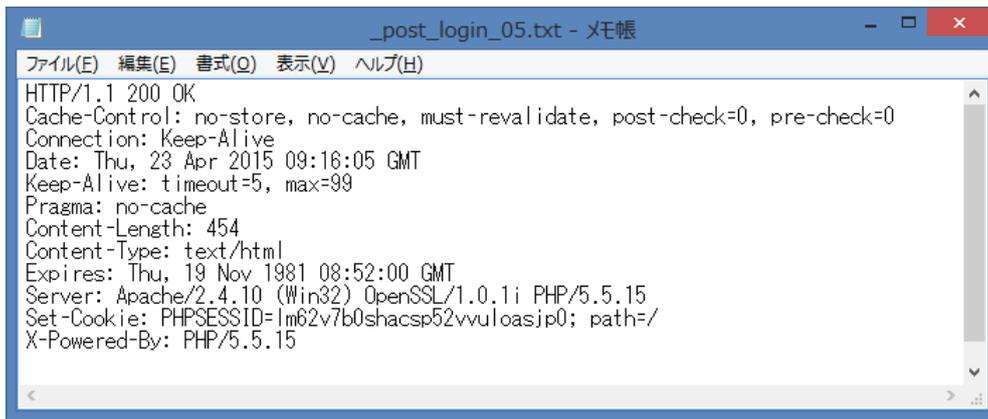
=====

ファイル・・・「MQL4/Files」フォルダに保存される

「_post_login_05.htm」をダブル・クリックするとブラウザが立ち上がり、下記が表示されます



「_post_login_05.txt」

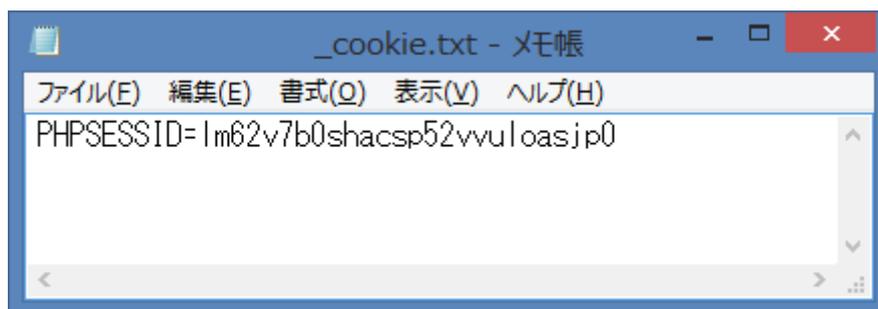


```
_post_login_05.txt - メモ帳
ファイル(E) 編集(E) 書式(O) 表示(V) ヘルプ(H)
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Connection: Keep-Alive
Date: Thu, 23 Apr 2015 09:16:05 GMT
Keep-Alive: timeout=5, max=99
Pragma: no-cache
Content-Length: 454
Content-Type: text/html
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
Set-Cookie: PHPSESSID=lm62v7b0shacsp52vvuloasjp0; path=/
X-Powered-By: PHP/5.5.15
```

内容書出し ;

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Connection: Keep-Alive
Date: Thu, 23 Apr 2015 09:16:05 GMT
Keep-Alive: timeout=5, max=99
Pragma: no-cache
Content-Length: 454
Content-Type: text/html
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
Set-Cookie: PHPSESSID=lm62v7b0shacsp52vvuloasjp0; path=/
X-Powered-By: PHP/5.5.15
```

「_cookie.txt」



```
_cookie.txt - メモ帳
ファイル(E) 編集(E) 書式(O) 表示(V) ヘルプ(H)
PHPSESSID=lm62v7b0shacsp52vvuloasjp0
```

内容書出し ;

```
PHPSESSID=lm62v7b0shacsp52vvuloasjp0
```

(4) [_post_main.mq4] 実行結果

#	Result	Protocol	Host	URL	Body	Caching	Content-Type
1	302	HTTP	kei1	/_login_05.php	37	no-store, n...	text/html
2	200	HTTP	kei1	/_menu_05.php	454	no-store, n...	text/html
3	200	HTTP	kei1	/_main_05.php	658	no-store, n...	text/html
4	200	HTTP	kei1	/_logout_05.php	377	no-store, n...	text/html

※「赤枠」内が実行・記録された内容です。

HTTP

Request:

[Headers]

Statistics Inspectors AutoResponder Composer Log Filters Timeline

Headers TextView WebForms HexView Auth Cookies Raw JSON XML

Request Headers [Raw] [Header Definitions]

POST /_main_05.php HTTP/1.1

Cache
Cache-Control: no-cache
Pragma: no-cache

Client
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
Accept-Charset: *,utf-8
Accept-Language: en
User-Agent: MetaTrader 4 Terminal/4.765 (Windows NT 6.2; x86)

Cookies / Login
 Cookie
PHPSESSID=lm62v7b0shacsp52vvuloasjp0

Entity
Content-Length: 111
Content-Type: application/x-www-form-urlencoded

Miscellaneous
Referer: http://localhost/_menu_05.php

Transport
Connection: Keep-Alive
Connection: Keep-Alive
Host: kei1

[WebForms]

QueryString	
Name	Value
Body	
Name	Value
site_05	tamama
contents	gunnsou_san
	tamama_desu
bt_05	yoroshiku
	onegai_shimasu
	Toukou_suru

[Raw]

```

POST http://kei1/_main_05.php HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
Content-Type: application/x-www-form-urlencoded
Accept: image/gif, image/x-xbitmap, image/jpeg, */*
Accept-Charset: *,utf-8
Accept-Language: en
Cookie: PHPSESSID=1m62v7b0shacsp52vvu1oasjpo
Host: kei1
Referer: http://localhost/_menu_05.php
User-Agent: MetaTrader 4 Terminal/4.765 (Windows NT 6.2; x86)
Connection: Keep-Alive
Content-Length: 111

site_05=tamama&contents=gunnsou_san%0D%0Atamama_desu%0D%0A%0D%0Ayoroshiku%0D%0Aonegai_shimasu&bt_05=Toukou_suru

```

※投稿内容（ボディ部に含まれる）を POST しています、

メッセージ・ボディはURLエンコード済みの下記です。

```
_ms_body="site_05=tamama&contents=gunnsou_san%0D%0Atamama_desu%0D%0A%0D%0Ayoroshiku%0D%0Aonegai_shimasu&bt_05=Toukou_suru"
```

上記はブラウザで手書で投稿した時の「Reques ; レスポンス」ボディをコピーしたものです、URLデコードすると、下記になります。（原文に近くなります）

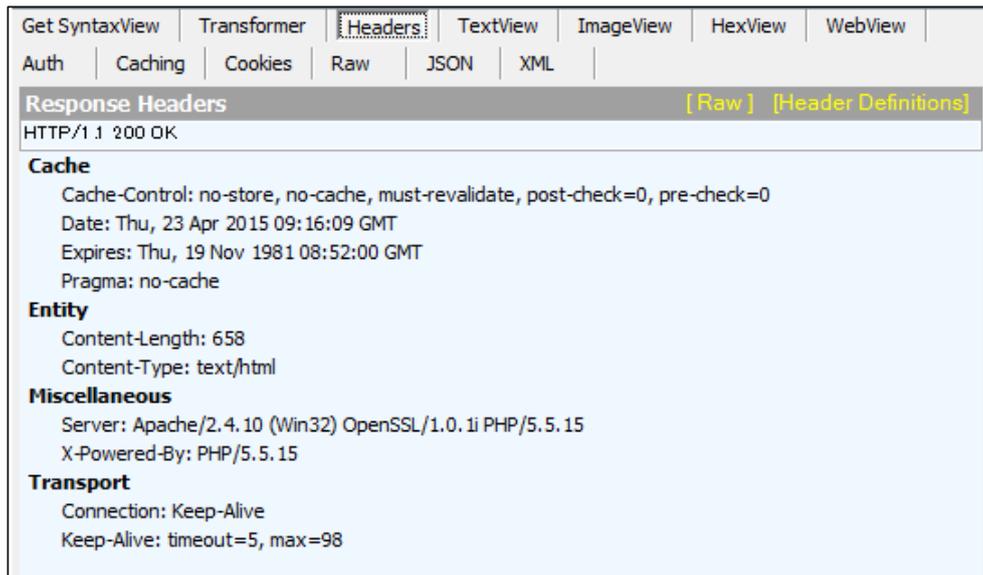
```
_ms_body="site_05=tamama&contents=gunnsou_san
tamama_desu
```

```
yoroshiku
```

```
onegai_shimasu&bt_05=Toukou_suru"
```

Response;

[Headers]



[WebView]



※上記が、ブラウザから手書きで投稿した時のイメージです。

[Raw]



```
HTTP/1.1 200 OK
Date: Thu, 23 Apr 2015 09:16:09 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
X-Powered-By: PHP/5.5.15
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 658
Keep-Alive: timeout=5, max=98
Connection: Keep-Alive
Content-Type: text/html

<html>
<head>
<meta http-equiv="Content-Type" content="charset=utf-8">
<title>SITE5 TEST</title>
</head>

<body>

<p><掲示板></p>

<form method="POST" action="/_main_05.php">
<p>
お名前:<input type="text" name="site_05"><br><br>
</p>
<p>
ご意見:<br>
<textarea name="contents" rows="8" cols="40"></textarea><br><br>
</p>
<input type="submit" name="bt_05" value="Toukou_suru">
</form>

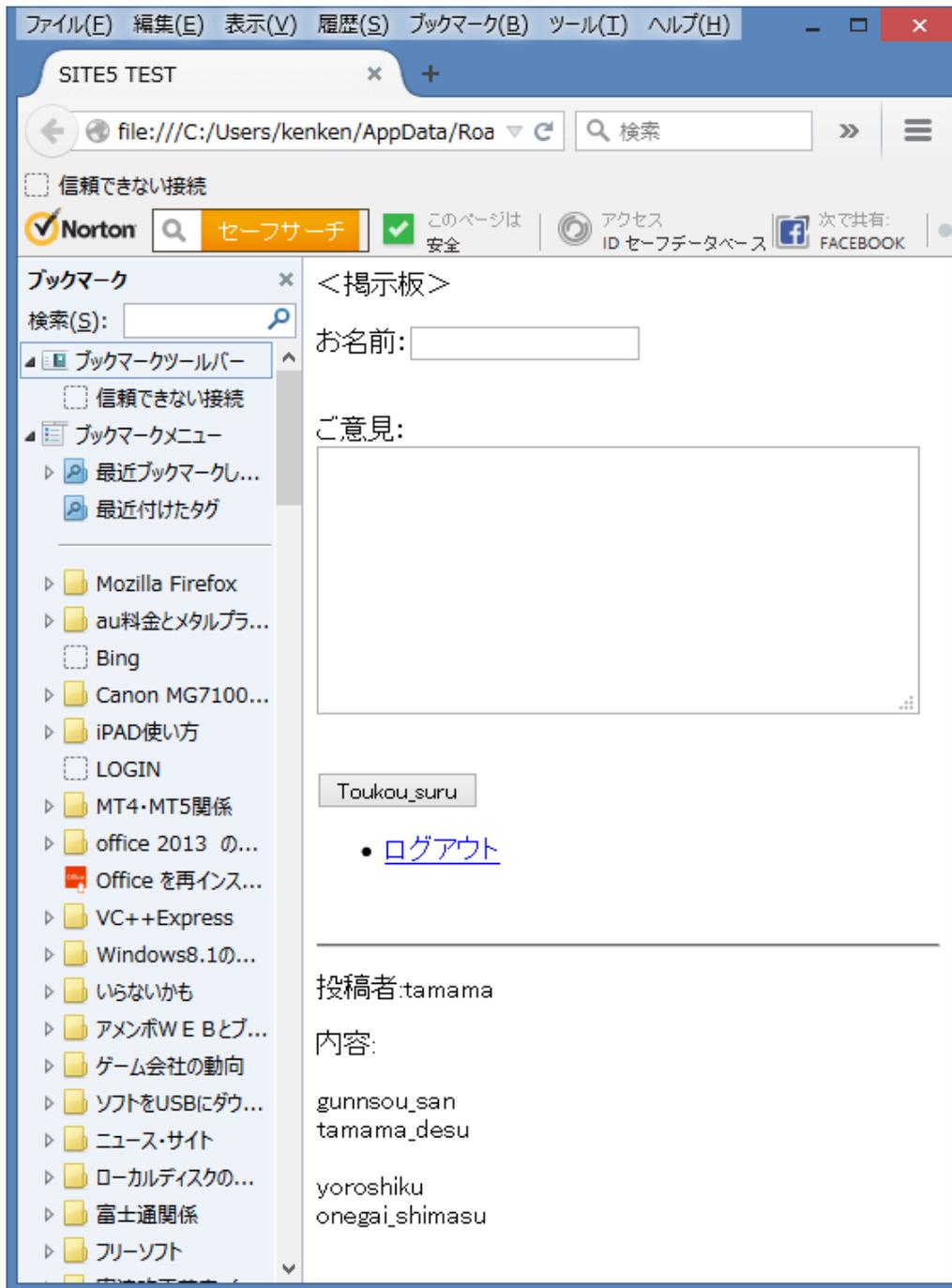
<ul>
<li><a href="_logout_05.php">ログアウト</a></li>
</ul>

</body>
</html>

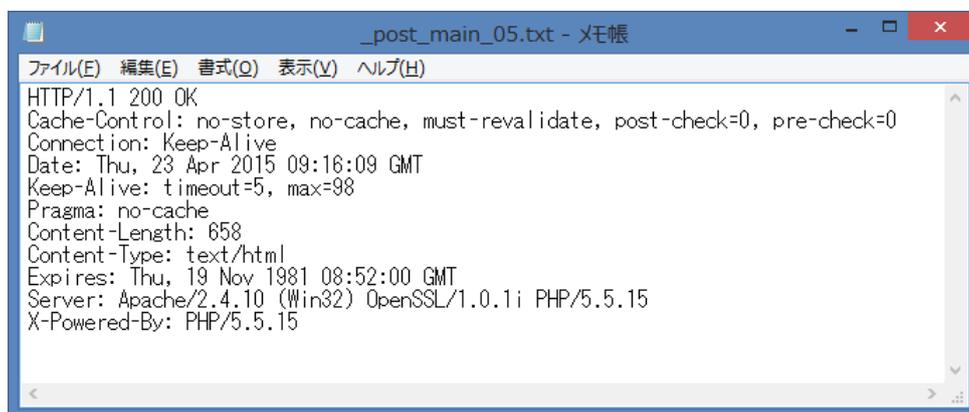
<hr>
<p>投稿者:tamama</p>
<p>内容:</p>
<p>gunnsou_san<br />
tamama_desu<br />
<br />
yoroshiku<br />
onegai_shimasu</p>
```

=====

ファイル・・・「MQL4/Files」フォルダに保存される
「_post_main_05.htm」・・・ダブルクリックする



「_post_main_05.txt」



内容書出し；

HTTP/1.1 200 OK

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

Connection: Keep-Alive

Date: Thu, 23 Apr 2015 09:16:09 GMT

Keep-Alive: timeout=5, max=98

Pragma: no-cache

Content-Length: 658

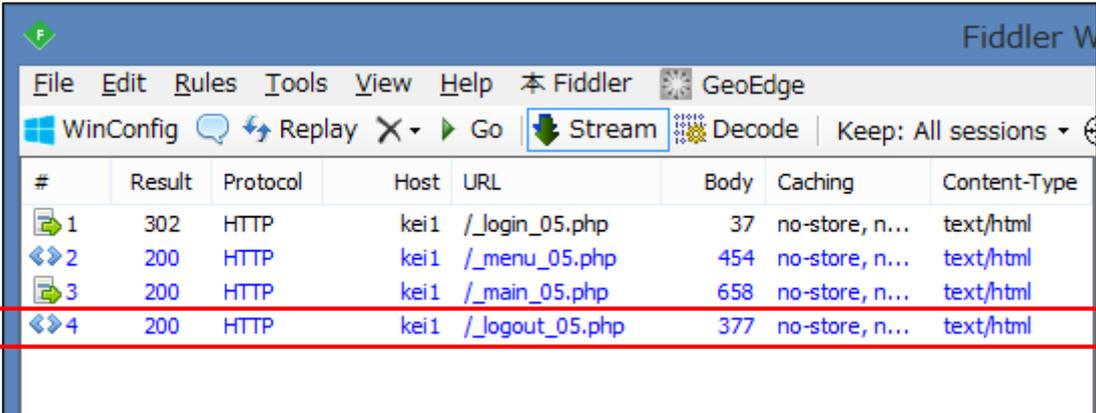
Content-Type: text/html

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15

X-Powered-By: PHP/5.5.15

(5) [_post_logout.mq4] 実行結果



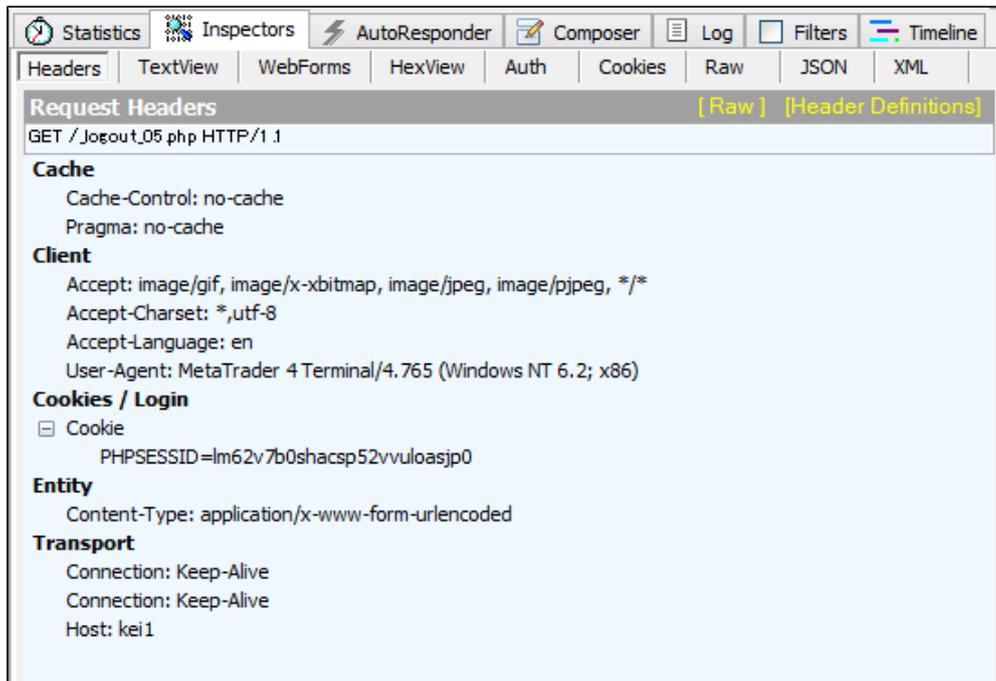
#	Result	Protocol	Host	URL	Body	Caching	Content-Type
1	302	HTTP	kei1	/_login_05.php	37	no-store, n...	text/html
2	200	HTTP	kei1	/_menu_05.php	454	no-store, n...	text/html
3	200	HTTP	kei1	/_main_05.php	658	no-store, n...	text/html
4	200	HTTP	kei1	/_logout_05.php	377	no-store, n...	text/html

※「赤枠」内が実行・記録された内容です。

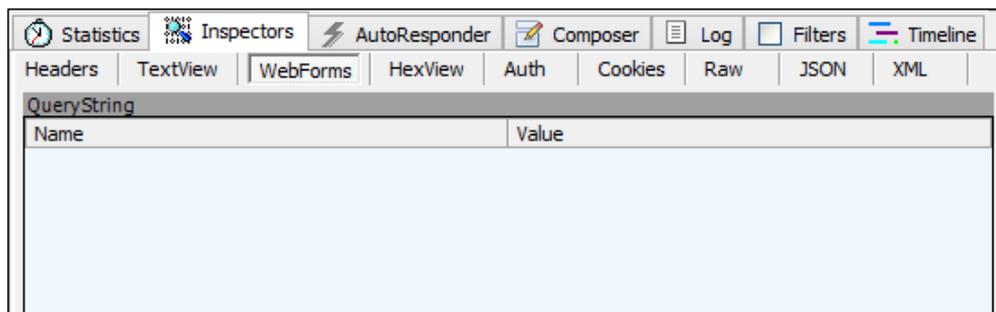
HTTP

Request:

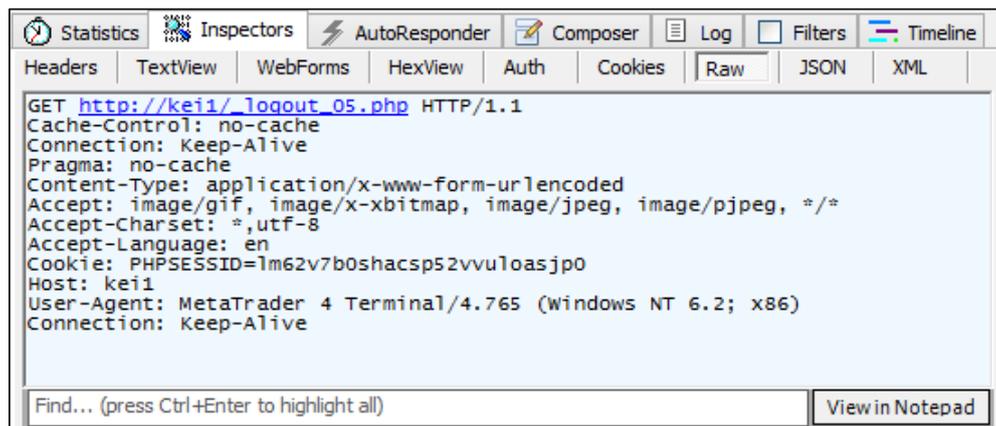
[Headers]



[WebForms]

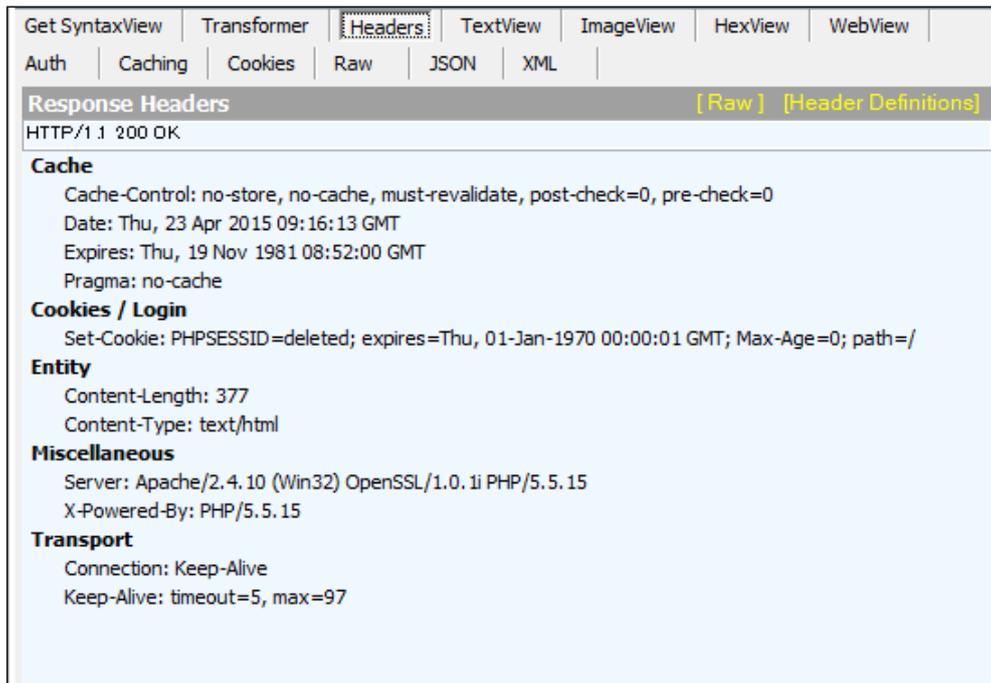


[Raw]

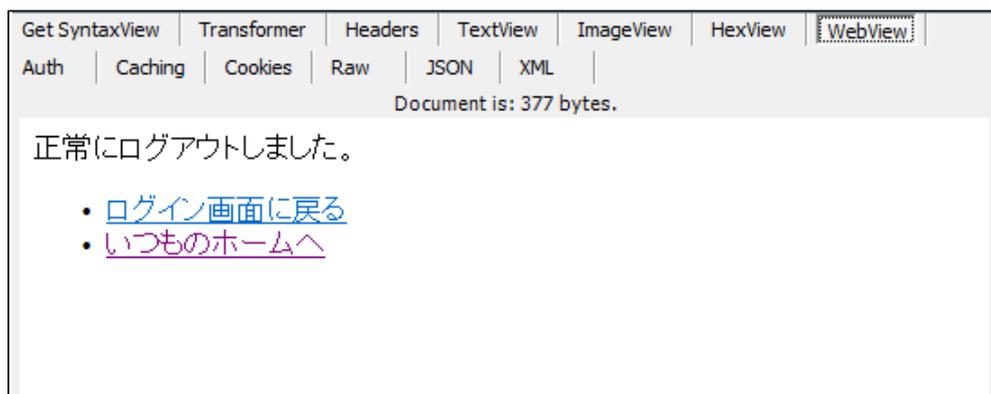


Response:

[Headers]



[WebView]



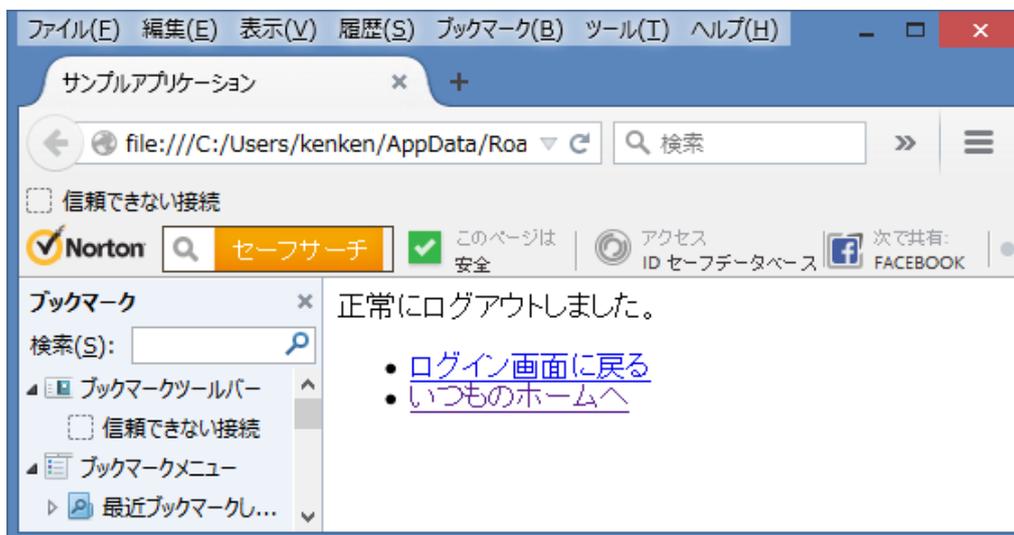
[Raw]



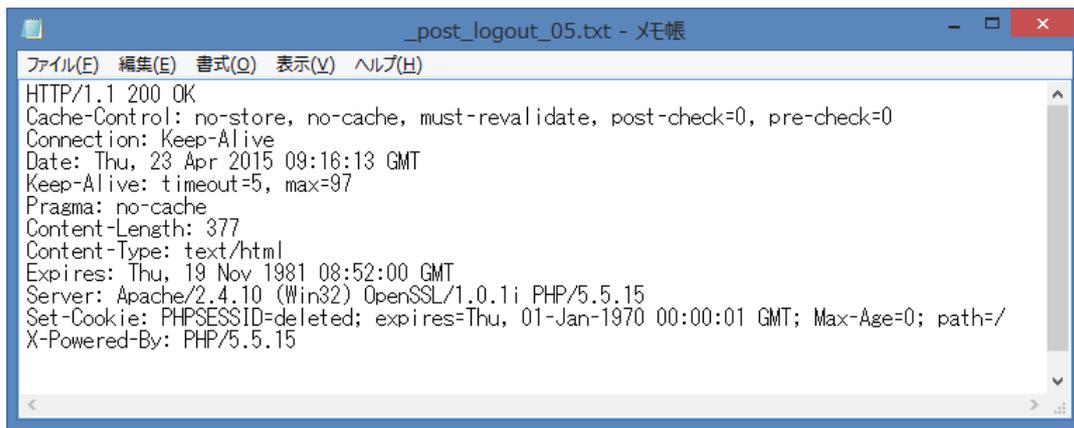
※ 「Set-Cookie: PHPSESSID=deleted;」 に注目、セッションが削除されて終了します。

=====
ファイル・・・「MQL4/Files」フォルダに保存される

「_post_logout_05.htm」・・・ダブルクリックする



「_post_logout_05.txt」



The screenshot shows a Notepad window titled "_post_logout_05.txt - メモ帳". The menu bar includes "ファイル(F)", "編集(E)", "書式(O)", "表示(V)", and "ヘルプ(H)". The text content is as follows:

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Connection: Keep-Alive
Date: Thu, 23 Apr 2015 09:16:13 GMT
Keep-Alive: timeout=5, max=97
Pragma: no-cache
Content-Length: 377
Content-Type: text/html
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
Set-Cookie: PHPSESSID=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=0; path=/
X-Powered-By: PHP/5.5.15
```

内容書出し ;

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Connection: Keep-Alive
Date: Thu, 23 Apr 2015 09:16:13 GMT
Keep-Alive: timeout=5, max=97
Pragma: no-cache
Content-Length: 377
Content-Type: text/html
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15
Set-Cookie: PHPSESSID=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=0; path=/
X-Powered-By: PHP/5.5.15
```

9. スクリプト・サンプル；コード詳細

※PHP コード内容の解説はいたしません、ご容赦！（MQL4 とは直接には関係しないため）

(1) PHPコード；セッション版

_login_05.php

```

<?php
// 「undefined index」 警告を消すために以下を挿入
error_reporting(E_ALL ^ E_NOTICE);

session_start();
// エラーメッセージ
$errorMessage = "";
// 画面に表示するため特殊文字をエスケープする
$viewUserId = htmlspecialchars($_POST["userid"], ENT_QUOTES);

// ログインボタンが押された場合
if (isset($_POST["login"])) {

    // 認証成功
    if ($_POST["userid"] == "keroro" && $_POST["password"] == "keroro") {
        // セッション ID を新規に発行する
        //session_regenerate_id(TRUE);//①
        $_SESSION["USERID"] = $_POST["userid"];
        //チェック用-----
        echo 'チェック：USER_ID= ' . $_SESSION["USERID"] . ' です';
        //-----
        header("Location: _menu_05.php");
        exit;
    }
    else {
        $errorMessage = "ユーザ ID、又はパスワードに誤りがあります。";
    }
}

?>
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>SITE5 TEST</title>
</head>
<body>
<form id="loginForm" name="loginForm" action="<?php print($_SERVER['PHP_SELF']) ?>"
method="POST">
    <fieldset>
    <legend>ログインフォーム</legend>
    <div><?php echo $errorMessage ?></div>
    <label for="userid">ユーザ ID</label><input type="text" id="userid" name="userid"
value=" Input ID ">
    <br>
    <label for="password">パスワード</label><input type="password" id="password"
name="password" value="">
    <br>
    <label></label><input type="submit" id="login" name="login" value="Login">
    </fieldset>
</form>
</body>
</html>

```

_menu_05.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="charset=utf-8">
<title>SITE5 TEST</title>
</head>
```

```
<?php error_reporting(E_ALL & ~E_NOTICE); ?>
```

```
<body>
<p><メニュー画面></p>
```

以下のメニューから、選択してください；

```
<ul>
<li><a href="_main_05.php">投稿画面へ</a></li>
<li><a href="_sub_05.php">予備画面へ</a></li>
<li><a href="_logout_05.php">ログアウト</a></li>
</ul>
```

```
</body>
</html>
```

```
<?php
//
session_start();
// セッション再発行
session_regenerate_id(TRUE);//①

//session_name("tamama");//課題=何故？NG

// ログイン状態のチェック
if (!isset($_SESSION["USERID"])) {
    header("Location: _no_05.php");
    exit;
}
//チェック用-----
echo 'チェック：USER_ID= ' . $_SESSION["USERID"]. ' です';
//-----
?>
```

main_05.php

```

<html>

<head>
<meta http-equiv="Content-Type" content="charset=utf-8">
<title>SITE5 TEST</title>
</head>

<?php error_reporting(E_ALL & ~E_NOTICE); ?>

<body>

<p><掲示板></p>

<form method="POST" action="<?php print($_SERVER['PHP_SELF']) ?>">
<p>
お名前: <input type="text" name="site_05"><br><br>
</p>
<p>
ご意見: <br>
<textarea name="contents" rows="8" cols="40"></textarea><br><br>
</p>
<input type="submit" name="bt_05" value="Toukou_suru">
</form>

<ul>
<li><a href="_logout_05.php">ログアウト</a></li>
</ul>

</body>

</html>

<?php
session_start();

// ログイン状態のチェック
if (!isset($_SESSION["USERID"])) {
    header("Location: _no_05.php");
    exit;
}

//--- 掲示板に入力したデータの処理
$file_name='site_05.txt';
if(!file_exists($file_name)){
    // ファイル作成
    touch($file_name);
}

//--- メイン・ルーチン
if($_SERVER["REQUEST_METHOD"] == "POST"){
    writeData();
}

readData();

//--- 関数類

```

```

function readData() {
    $site_05_file = 'site_05.txt';

    $fp = fopen($site_05_file, 'rb');

    if ($fp) {
        if (flock($fp, LOCK_SH)) {
            while (!feof($fp)) {
                $buffer = fgets($fp);
                print($buffer);
            }

            flock($fp, LOCK_UN);
        } else {
            print('ファイルロックに失敗しました');
        }
    }

    fclose($fp);
}

//
function writeData() {
    $site_05 = $_POST['site_05'];
    $contents = $_POST['contents'];
    $contents = nl2br($contents);

    $data = "<hr>¥r¥n";
    $data = $data. "<p>投稿者:". $site_05. "</p>¥r¥n";
    $data = $data. "<p>内容:</p>¥r¥n";
    $data = $data. "<p>". $contents. "</p>¥r¥n";

    $site_05_file = 'site_05.txt';

    $fp = fopen($site_05_file, 'ab');

    if ($fp) {
        if (flock($fp, LOCK_EX)) {
            if (fwrite($fp, $data) === FALSE) {
                print('ファイル書き込みに失敗しました');
            }
            flock($fp, LOCK_UN);
        } else {
            print('ファイルロックに失敗しました');
        }
    }

    fclose($fp);
}

?>

```

※以下の内容だけは補足しておきます。

「投稿した内容」は、XAMMP 直下の htdocs フォルダ内に生成した、「site_05.txt」ファイル中に書き込まれます。(追記されていきます)

再度「投稿する」と、そのファイル内容を読み出してブラウザ上に表示していきます。

(書き込み内容をクリアするには、ファイルを削除するか、内容を消してください)

`_sub_05.php`

```
<html>
<head>
<meta http-equiv="Content-Type" content="charset=utf-8">
<title>SITE5 TEST</title>
</head>

<?php error_reporting(E_ALL & ~E_NOTICE); ?>

<body>
<p><予備画面></p>

ただ今、工事中です。<br>
下記から選択してください。<br>

<ul>
<li><a href="_main_05.php">投稿画面へ</a></li>
<li><a href="_logout_05.php">ログアウト</a></li>
</ul>

</body>
</html>

<?php
session_start();

// ログイン状態のチェック
if (!isset($_SESSION["USERID"])) {
    header("Location: _no_05.php");
    exit;
}

?>
```

_logout_05.php

```

<?php
session_start();

if (isset($_SESSION["USERID"])) {
    $errorMessage = "正常にログアウトしました。";
}
else {
    $errorMessage = "セッションがタイムアウトしました。(異常終了)";
}
// セッション変数のクリア
$_SESSION = array();
// クッキーの破棄
if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000,
        $params["path"], $params["domain"],
        $params["secure"], $params["httponly"]
    );
}
// セッションクリア
@session_destroy();
?>

<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>サンプルアプリケーション</title>
  </head>
  <body>
    <div><?php echo $errorMessage; ?></div>
    <ul>
      <li><a href="_login_05.php">ログイン画面に戻る</a></li>
      <li><a href="http://www.benri.ne.jp/">いつものホームへ</a></li>
    </ul>
  </body>
</html>

```

_no_05.php

```

<?php
    //空っぽ
?>

<!doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content=" charset=utf-8">
<title>SITE5 TEST</title>
</head>
  <body>
    <br>これは、不正アクセスです！<br>
    <br>ログインから、やり直してください。<br>
  </body>
</html>

```

(2) PHPコード (セッション無し版)

※ 「セッション版」のセッション関係のコードをコメント・アウトしただけのものが、「セッション無し版」です、詳細は実際のコードを参照してください。

例； _menu_05.php の「セッション版」と「セッション無し版」を比較します

_menu_05.php の「セッション版」；

```
<html>
<head>
<meta http-equiv="Content-Type" content="charset=utf-8">
<title>SITE5 TEST</title>
</head>
```

```
<?php error_reporting(E_ALL & ~E_NOTICE); ?>
```

```
<body>
<p><メニュー画面></p>
```

以下のメニューから、選択してください；

```
<ul>
<li><a href="_main_05.php">投稿画面へ</a></li>
<li><a href="_sub_05.php">予備画面へ</a></li>
<li><a href="_logout_05.php">ログアウト</a></li>
</ul>
```

```
</body>
</html>
```

```
<?php
//
session_start();
// セッション再発行
session_regenerate_id(TRUE); //①

//session_name("tamama");//課題=何故？NG

// ログイン状態のチェック
if (!isset($_SESSION["USERID"])) {
    header("Location: _no_05.php");
    exit;
}
//チェック用-----
echo 'チェック：USER_ID= ' . $_SESSION["USERID"] . ' です';
//-----
?>
```

_menu_05.php の「セッション無し版」;

```
<html>
<head>
<meta http-equiv="Content-Type" content="charset=utf-8">
<title>SITE5 TEST</title>
</head>
```

```
<?php error_reporting(E_ALL & ~E_NOTICE); ?>
```

```
<body>
<p><メニュー画面></p>
```

以下のメニューから、選択してください;

```
<ul>
<li><a href="_main_05.php">投稿画面へ</a></li>
<li><a href="_sub_05.php">予備画面へ</a></li>
<li><a href="_logout_05.php">ログアウト</a></li>
</ul>
```

```
</body>
</html>
```

```
<?php
//
////session_start();////
// セッション再発行
////session_regenerate_id(TRUE);//①////

//session_name("tamama");//課題=何故? NG

// ログイン状態のチェック
////if (!isset($_SESSION["USERID"])) {
//     ////header("Location: _no_05.php");
//     ////exit;
// }
//チェック用-----
//echo 'チェック : USER_ID= ' . $_SESSION["USERID"]. ' です';
//-----
?>
```

(3) MQL 4 コード

試してみる場合は、「**keil**」部分を読者PCのマシン名に変更してください。

(詳細は、「2. (2)」を参照ください)

※コード内容の解説は省略いたします、読者にて解析してみてください、

小生はバージョン2 (引数7個) の **WebRequest()** サンプル・コードを参考に作成しました。

(それほど難しいテクニックは含まれていません)

_post_login.mq4

```
//+-----+
//|                                     _post_login.mq4 |
//|                                     amenbo         |
//|                                     泉の森の弁財天池 |
//+-----+
#property copyright "amenbo"
#property link      "泉の森の弁財天池"
#property version   "1.00"
#property strict

//+-----+
//| Script program start function |
//+-----+
void OnStart()
{
    int  res_pos;    // To receive the operation execution result
    char post[];    // Data array to send POST requests
    char result[];  // Data array for Response data
    string _ms_body="userid=keroro&password=keroro&login>Login"; //_login_05. php
    string cookie=NULL, headers;
//-----
    string local_url="http://keil/_login_05. php";
//--- reset last error
    ResetLastError();
//--- load html page from localhost/_start.html
    int timeout=5000; //--- timeout less than 1000 (1 sec.)
//--- set referer
    //string referer="fukutoku.benzaiten. ike";
//--- post データを準備する
    int _size=StringToCharArray(_ms_body, post, 0, WHOLE_ARRAY, CP_UTF8);
    int _count=ArrayResize(post, _size-1);
//-----
//

res_pos=WebRequest("POST", local_url, cookie, NULL, timeout, post, ArraySize(post), result, headers);
//
//-----レスポンス結果をファイルに記録する-----
    PrintFormat("POST 成功, size =%d bytes.", ArraySize(result));
//--- メッセージ・ボディーを result[] に記録
    int filehandle1=FileOpen("_post_login_05. htm", FILE_WRITE|FILE_BIN);
//--- check
    if(filehandle1!=INVALID_HANDLE)
    {
//--- メッセージ・ボディーを result[] に記録
        FileWriteArray(filehandle1, result, 0, ArraySize(result));
//--- close file
    }
}
```

```

    FileClose(filehandle1);
}
else Print("Error in File1_Open. Error code=", GetLastError());
//--- ヘッダーを headers に記録
int filehandle2=FileOpen("_post_login_05.txt", FILE_WRITE|FILE_TXT);
//---
if(filehandle2!=INVALID_HANDLE)
{
    //--- write response headers to file
    FileWriteString(filehandle2, headers);
    //--- close file
    FileClose(filehandle2);
}
else Print("Error in File2_Open. Error code=", GetLastError());
//
//-----セッションID (クッキー) をファイルに保存する-----
int filehandle3=FileOpen("_post_login_05.txt", FILE_READ|FILE_TXT);
//---
if(filehandle3!=INVALID_HANDLE)
{
    int res;
    string str, _cookie, _cookie_set;
    //
    while(!FileIsEnding(filehandle3))
    {
        str=FileReadString(filehandle3);
        res=StringFind(str, "Set-Cookie: PHPSESSID=");
        //
        if(res>=0)//①
        {
            PrintFormat(str);
            _cookie=StringSubstr(str, res+12);
            _cookie_set=StringSubstr(_cookie, 0, StringFind(_cookie, ";"));
            //
            PrintFormat(_cookie_set);
            //
            int filehandle4=FileOpen("_cookie.txt", FILE_WRITE|FILE_TXT);
            FileWriteString(filehandle4, _cookie_set);
            FileClose(filehandle4);
        }
    }
    //
}
//
//
FileClose(filehandle3);
}
//-----
PlaySound("alert2");
}
//=====

```

_post_main.mq4

```

//+-----+
//|                                     _post_main.mq4 |
//|                                     amenbo       |
//|                                     泉の森の弁財天池 |
//+-----+
#property copyright "amenbo"
#property link      "泉の森の弁財天池"
#property version   "1.00"
#property strict
//+-----+
//| Script program start function |
//+-----+
void OnStart()
{
    int  res_pos;    // To receive the operation execution result
    char post[];    // Data array to send POST requests
    char result[];  // Data array for Response data
    string
_ms_body="site_05=tamama&contents=gunnsou_san%0D%0A%0D%0A%0D%0Ayoroshiku%0D%
0Aonegai_shimasu&bt_05=Toukou_suru";
    string cookie,headers;
//-----
    string local_url="http://keil/_main_05.php";
//--- reset last error
    ResetLastError();
//--- load html page from localhost/_start.html
    int timeout=5000; //--- timeout less than 1000 (1 sec.)
//--- set referer
    string referer="http://localhost/_menu_05.php";//一応書いてみた
//--- post データを準備する
    int _size=StringToArray(_ms_body, post, 0, WHOLE_ARRAY, CP_UTF8);
    int _count=ArrayResize(post, _size-1);
//-----セッションを継続するためにセッション I D (クッキー) を読み出す-----
    int filehandle3=FileOpen("_cookie.txt",FILE_READ|FILE_TXT);
    cookie=FileReadString(filehandle3);
    FileClose(filehandle3);
//-----
//

res_pos=WebRequest("POST", local_url, cookie, referer, timeout, post, ArraySize(post), result,
headers);
//
//-----レスポンス結果をファイルに記録する-----
    PrintFormat("POST 成功, size =%d bytes.", ArraySize(result));
//--- メッセージ・ボディを result[] に記録
    int filehandle1=FileOpen("_post_main_05.htm",FILE_WRITE|FILE_BIN);
//--- check
    if(filehandle1!=INVALID_HANDLE)
    {
        //--- メッセージ・ボディを result[] に記録
        FileWriteArray(filehandle1, result, 0, ArraySize(result));
//--- close file
        FileClose(filehandle1);
    }
    else Print("Error in File1_Open. Error code=", GetLastError());
//--- ヘッダーを headers に記録
    int filehandle2=FileOpen("_post_main_05.txt",FILE_WRITE|FILE_TXT);

```

```
//---
if(filehandle2!=INVALID_HANDLE)
{
    //--- write response headers to file
    FileWriteString(filehandle2,headers);
    //--- close file
    FileClose(filehandle2);
}
else Print("Error in File2_Open. Error code=",GetLastError());
//
//-----
    PlaySound("alert");
}
//=====
```

_post_logout.mq4

```

//+-----+
//|                                     _post_logout.mq4 |
//|                                     amenbo          |
//|                                     泉の森の弁財天池 |
//+-----+
#property copyright "amenbo"
#property link      "泉の森の弁財天池"
#property version   "1.00"
#property strict

//+-----+
//| Script program start function |
//+-----+
// セッションを正常に終了させるために、呼出す必要がある。
//
void OnStart()
{
    int  res_get;    // To receive the operation execution result
    char get[];     // Data array to send GET requests
    char result[];  // Data array for Response data
    string cookie,headers;

//-----
    string local_url="http://keil/_logout_05.php";
//--- reset last error
    ResetLastError();
//--- load html page from localhost/_start.html
    int timeout=5000; //--- timeout less than 1000 (1 sec.)
//--- set referer
    //string referer="fukutoku.benzaiten.ike";
//-----セッションを継続するためにセッション I D (クッキー) を読み出す-----
    int filehandle1=FileOpen("_cookie.txt",FILE_READ|FILE_TXT);
    cookie=FileReadString(filehandle1);
    FileClose(filehandle1);
//

res_get=WebRequest("GET", local_url, cookie, NULL, timeout, get, ArraySize(get), result, headers);
//
//-----レスポンス結果をファイルに記録する-----
    PrintFormat("GET 成功, size =%d bytes.", ArraySize(result));
//--- メッセージ・ボディを result[] に記録
    int filehandle2=FileOpen("_post_logout_05.htm",FILE_WRITE|FILE_BIN);
//--- check
    if(filehandle2!=INVALID_HANDLE)
    {
        //--- メッセージ・ボディを result[] に記録
        FileWriteArray(filehandle2, result, 0, ArraySize(result));
//--- close file
        FileClose(filehandle2);
    }
    else Print("Error in File1_Open. Error code=", GetLastError());
//--- ヘッダーを headers に記録
    int filehandle3=FileOpen("_post_logout_05.txt",FILE_WRITE|FILE_TXT);
//---
    if(filehandle3!=INVALID_HANDLE)
    {
        //--- write response headers to file
        FileWriteString(filehandle3, headers);

```

```
//--- close file
FileClose(filehandle3);
}
else Print("Error in File2_Open. Error code=", GetLastError());
//
//---
PlaySound("alert2");
}
//=====
```

以 上