

The background features a grid of light gray squares. Overlaid on this are several wide, colorful rays emanating from a central point on the left. The rays are in shades of orange, red, green, blue, and gray. Faint binary code (0s and 1s) is scattered across the background, particularly in the upper right area.

## Chapter 8

# Cookie และ Session

Theerayut Thongkrau

## ➡ ปัญหาของ Stateless

- HTTP เป็นโปรโตคอลแบบ **Stateless** หรือ **Connectionless** กล่าวคือ request และ response ที่เกิดในแต่ละครั้งมีความเป็นอิสระ ไม่เกี่ยวข้องกัน
- Stateless คือ การส่ง request และรับ response จาก Server และจบการติดต่อทันที หากผู้ใช้คลิกไปยังเว็บหน้าอื่นๆ แม้เว็บหน้านั้นจะอยู่บน Server เดียวกันก็จะเกิดกระบวนการติดต่อใหม่ทุกครั้ง ไม่มีการจดจำสถานะใดๆ
- Request และ Response ในทุกๆ ครั้งที่เกิดไม่สามารถใช้ข้อมูลใดๆ ก่อนหน้านี้ได้

## ➡ เว็บไซต์ที่ต้องการจดจำการติดต่อ

- เว็บไซต์ที่มีการ Sign In เข้าสู่ระบบ เมื่อผู้ใช้ Sign In ผ่านแล้ว เว็บไซต์หน้าอื่นๆ จะต้องรู้ว่าผู้ใช้คือใคร
- เว็บไซต์ที่มีการกำหนดค่าการแสดงผล เช่น เลือกภาษาไทย/อังกฤษ
- เว็บไซต์สินค้า ลูกค้าจะเลือกสินค้าจากเว็บเพจหน้าใดก็ได้เก็บลงตะกร้าสินค้า เพราะการตัดสินใจเลือกสินค้าที่ต้องการในเว็บเพจหน้าเดียวเป็นไปได้ยาก

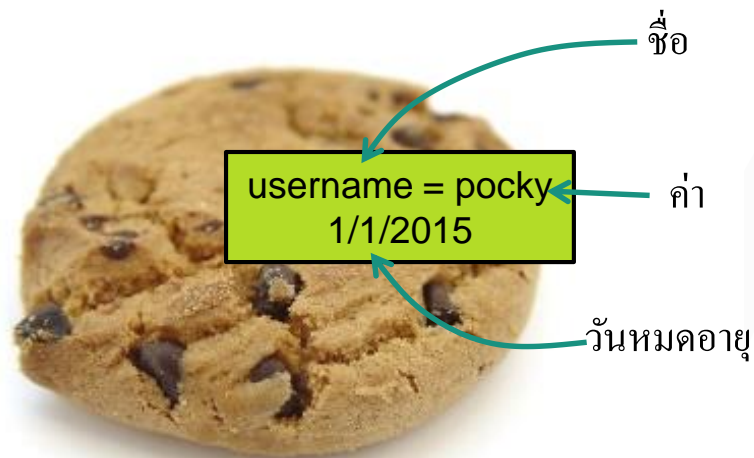
## ➡ Stateful จดจำการติดต่อ

- **Stateful หรือ Connection-oriented** คือ การที่ Server จดจำ Client ที่เข้ามายังเว็บไซต์ เทคนิคที่นำมาใช้ในการจดจำผู้ใช้แต่ละคนประกอบด้วย
  - Cookies คือ ข้อมูลขนาดเล็กที่ถูกส่งมาจาก Server เพื่อเก็บลงในเครื่องผู้ใช้ โดยมี Web Browser เป็นผู้จัดการ
  - Session คือ object ที่ถูกสร้างขึ้นบน Server เพื่อเก็บข้อมูลของผู้ใช้แต่ละคน โดยจะเกิดในระหว่างที่มีการติดต่อกับผู้ใช้ ในระยะเวลาสั้นๆ Web Server จะเป็นผู้จัดการ

# ➔ Cookie

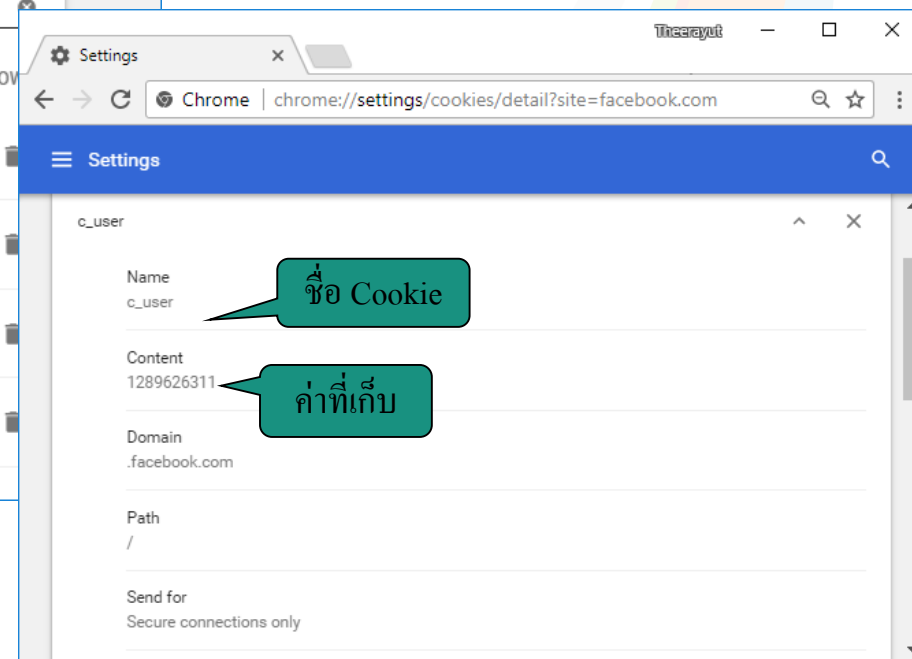
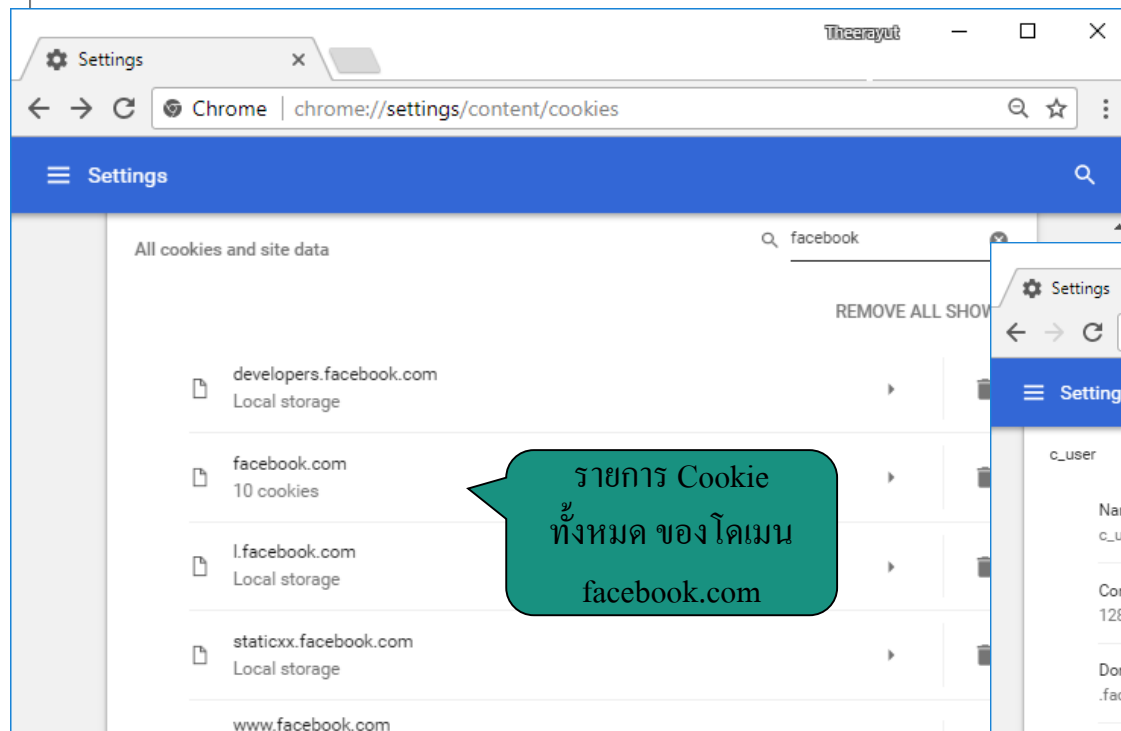
- คุกกี้ คือ ไฟล์ขนาดเล็กที่ Server เขียนบนเครื่องผู้เข้าชมเว็บ
- คุกกี้มักใช้ในการระบุตัวตนของผู้ใช้ โดยใช้ร่วมกับ Session
- เมื่อมีการร้องขอหน้าเว็บเพจจากเครื่องเดิมที่เคยเข้าและเขียนคุกกี้แล้ว จะมีการส่งข้อมูลในคุกกี้ไปด้วย

องค์ประกอบของคุกกี้



# ➡ การดูข้อมูลจาก Cookies ที่เขียนไปแล้ว

1. เปิดโปรแกรม Google Chrome ไปที่เมนู Settings
2. เลือกที่ Advanced หลังจากนั้นเลือกที่ Content settings...
3. หัวข้อ Cookies จะแสดงคุกกี้ทั้งหมด
4. สามารถค้นหาคุกกี้ตามชื่อโดเมนได้โดยพิมพ์ที่ช่อง Search cookies



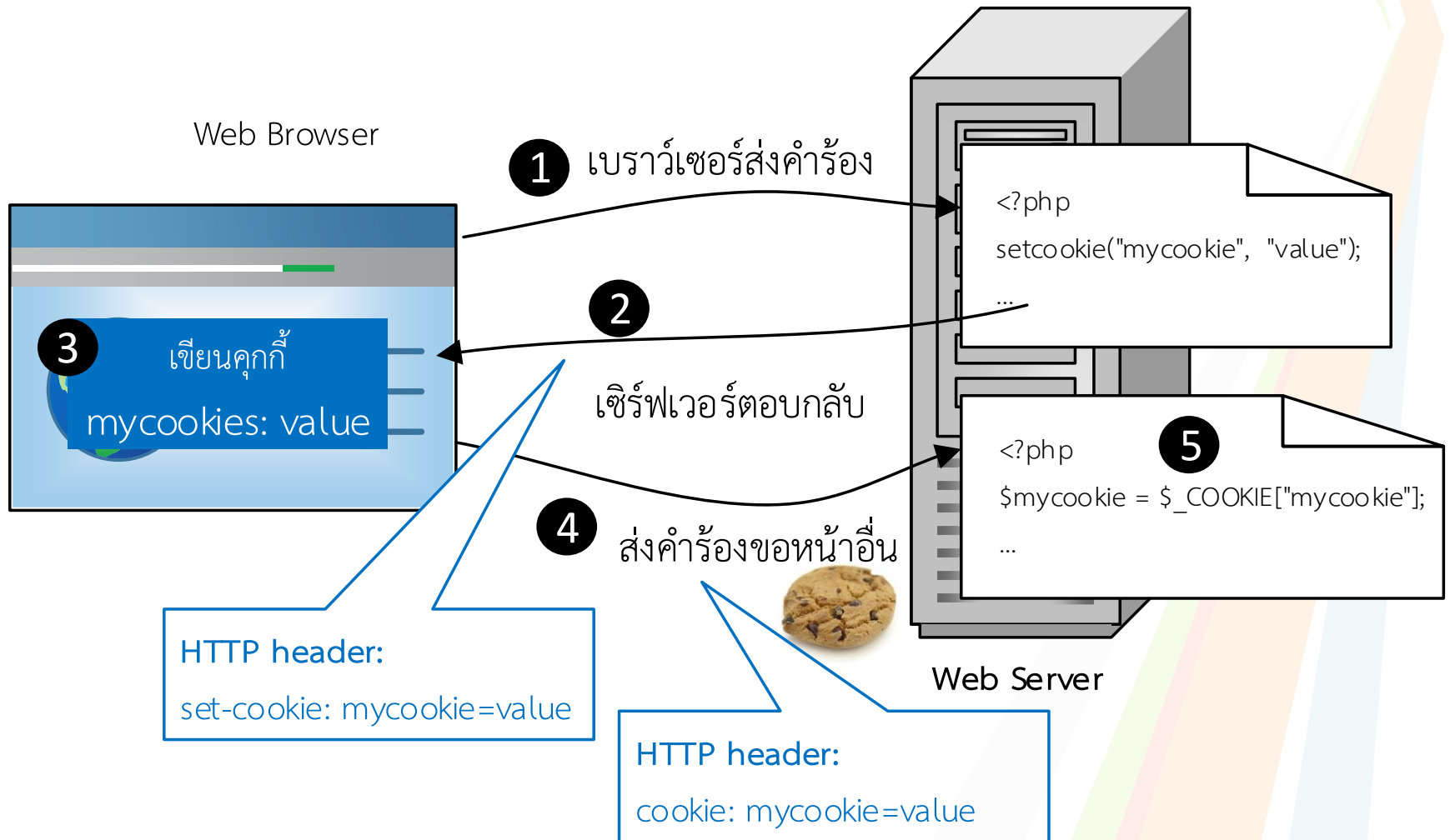
# ➡ การดูข้อมูลจาก Cookies ที่เขียนไปแล้ว

สำหรับมุมมองนักพัฒนาสามารถกดปุ่ม F12 และเลือกที่แท็บ Application และเลือกที่ Cookies จะแสดงรายการคุกกี้ของเว็บไซต์ที่กำลังเปิดอยู่

The screenshot shows a web browser window with the Facebook homepage. The developer tools are open, and the 'Application' tab is selected. Under 'Storage', the 'Cookies' section for 'https://www.facebook.com' is expanded. A table of cookies is displayed. Two callout boxes are present: one pointing to the 'Name' column labeled 'ชื่อ Cookie' and another pointing to the 'Value' column labeled 'ค่าที่เก็บ'.

Name	Value	Domain	Path	Expires / Max-Age	Size	...	Sec...	Sam...
__utma	205758533.2039236764.14...	.gppon...	/	2019-04-02T09:41:54....	63			
act	1508766360719%2F30	.facebo...	/	Session	21		✓	
c_user	1289626311	.facebo...	/	2018-01-21T13:13:09....	16		✓	
datr	McO6WQ7A6nkEhC8CAYla...	.facebo...	/	2019-09-14T17:58:08....	28	✓	✓	
dpr	1	.facebo...	/	2017-10-30T13:36:26....	4		✓	
fr	0LvIRT3euOpK5Pewy.AWW...	.facebo...	/	2018-01-21T13:13:09....	81	✓	✓	
pl	n	facebo	/	2017-12-13T17:58:08	3	✓	✓	

# 👉 วงจรการทำงานของคุกกี้





# ➔ HTTP Request & Response

- Cookies จะเก็บข้อมูลในรูปแบบ name=value ซึ่งถูกส่งระหว่าง server กับ client
- Cookies ใช้เก็บสถานะ หรือข้อมูลบางอย่างที่ Server ต้องการใช้ในครั้งต่อไป หลังจากผู้ใช้ส่ง request มายัง Server อีกครั้ง

```
HTTP/1.1 200 OK
Set-Cookie: username=TomasHirsch
Content-Type: text/html
Content-Length: 397
Date: Wed, 19 Nov 2003 03:25:40 GMT
Server: Apache-Coyote/1.1
Connection: close

<html>
...
</html>
```

**response** จาก Server ครั้งแรกที่ส่งมา  
และต้องการเก็บข้อมูลใน cookies  
ของเครื่องผู้ใช้

```
POST /select/selectBeerTaste2.do HTTP/1.1
Host: www.wickedlysmart.com
User-Agent: Mozilla/5.0
Cookie: username=TomasHirsch
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.1
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
```

**request** ครั้งต่อไป จะแนบข้อมูล  
ใน cookies ที่เคยเก็บในเครื่อง  
ผู้ใช้ ส่งไปยัง server ด้วย

# ➡ การเขียน Cookies ลงบนเครื่องของ Client

- รูปแบบการเขียนคุกกี้

`setcookie("ชื่อคุกกี้", "ค่าที่ต้องการเก็บ", time() + เวลาที่หมดอายุ);`

- `time()` เป็นฟังก์ชันดึงเวลาในปัจจุบัน
- อายุของคุกกี้ มีหน่วยเป็นวินาที

- ตัวอย่าง

`setcookie("username", "tjung", time() + 3600 * 24 * 365);`

1 วันมี 24 ชั่วโมง

1 ชั่วโมงมี 3600 วินาที

จำนวนวัน

- 1 domain สามารถเขียนคุกกี้ได้หลายค่า

## ➡ ตัวอย่าง

```
<html>
<body>
<?php
    setcookie("visit", 0, time() + 3600 * 24 * 1);
?>
</body>
</html>
```

localhost/cookie.php

Welcome to my website! Click here for a tour

Application

Cookies

http://localhost

Name	Value	Domain	Path	Expires / Max-Age	Size
visit	0	localhost	/	2017-10-23T13:31:07.047Z	6

# 👉 ใช้คุกกี้ตรวจสอบการเข้าชมเว็บ

```
<html>
<body>
<?php
```

```
// ถ้าคุกกี้ visit เป็นค่าว่าง ให้สร้างคุกกี้ visit และกำหนดค่าเริ่มต้นเป็น 0
if (empty($_COOKIE["visit"])) {
    setcookie("visit", 0, time() + 3600 * 24);
}
```

```
// ตรวจสอบว่าคุกกี้ชื่อ visit ถูกกำหนดค่าไว้แล้วหรือไม่
// ถ้ายังไม่กำหนด
```

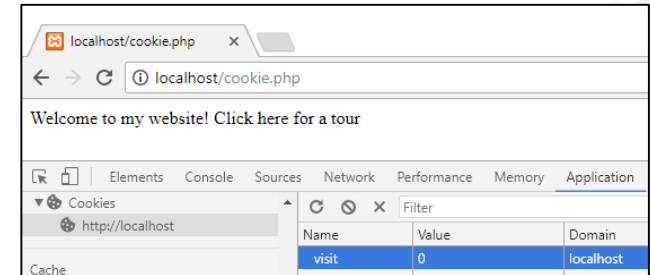
```
if (!isset($_COOKIE["visit"])) {
    echo "Welcome to my website! Click here for a tour";
}
```

```
// ถ้ากำหนดค่าแล้ว จะเพิ่มค่าขึ้น 1 ค่า
} else {
```

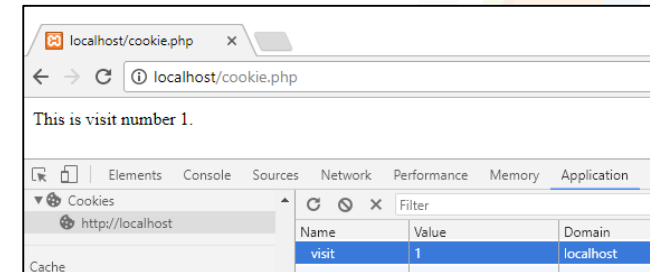
```
    $visit = $_COOKIE["visit"] + 1;
    setcookie("visit", $visit, time() + 3600 * 24);
    echo "This is visit number $visit.";
}
```

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

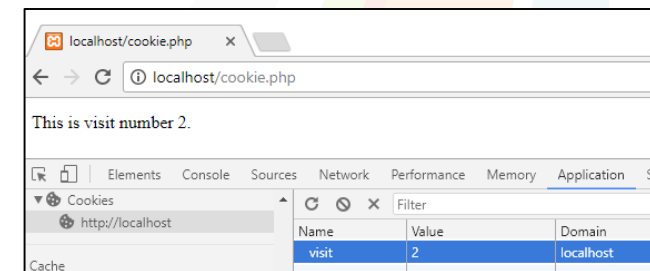
หน้าเว็บที่เข้าครั้งแรก



หน้าเว็บที่เข้าครั้งที่ 2



หน้าเว็บที่เข้าครั้งที่ 3



## ➡ การอ่านข้อมูลจาก Cookies

- ค่าของคุณก็จะถูกส่งมาพร้อมกับคำร้อง และ php จะนำชื่อและค่าเก็บไว้ในอาร์เรย์ชื่อ `$_COOKIE` อัตโนมัติ สามารถอ้างอิงได้ด้วยชื่อคุณก็ เช่น

```
echo $_COOKIE["ชื่อคุณก็"]
```

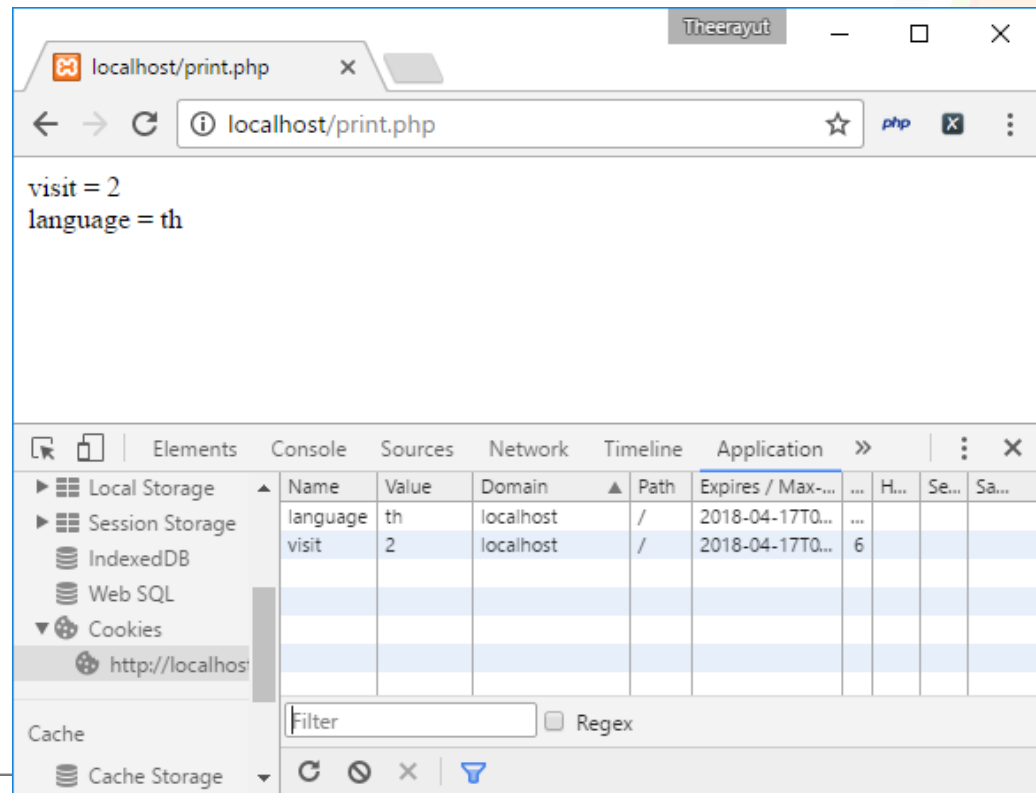
- การลบคุณก็

```
setcookie("ชื่อคุณก็", "", time());
```

กำหนดเป็นค่าว่าง

# ➡ ตัวอย่างการอ่านคุกกี้ทั้งหมด

```
<html>
<body>
<?php
    foreach ($_COOKIE as $i => $value) {
        echo "$i = $value<br>";
    }
?>
</body>
</html>
```



## ➡ กิจกรรม

- จงสร้างไฟล์ select.php สำหรับใช้เขียน Cookies ชื่อ lang ลงบนเครื่องของผู้ใช้ โดยค่าของคุณก็ที่เขียนขึ้นอยู่กับการส่งข้อมูลผ่าน URL ดังนี้
  - คุณก็ lang มีค่าเป็น en เมื่อรัน `http://localhost/select.php?language=en`
  - คุณก็ lang มีค่าเป็น th เมื่อรัน `http://localhost/select.php?language=th`
- สร้างไฟล์ main.php โดยให้อ่านค่าจาก Cookie ชื่อ lang ที่เขียนไปแล้ว โดยเขียนเงื่อนไขตรวจสอบว่าค่าในคุณก็มีค่าเป็นอะไร
  - ถ้าคุณก็ lang มีค่าเป็น en ให้แสดงคำว่า "Welcome"
  - ถ้าคุณก็ lang มีค่าเป็น th ให้แสดงคำว่า "ยินดีต้อนรับ"

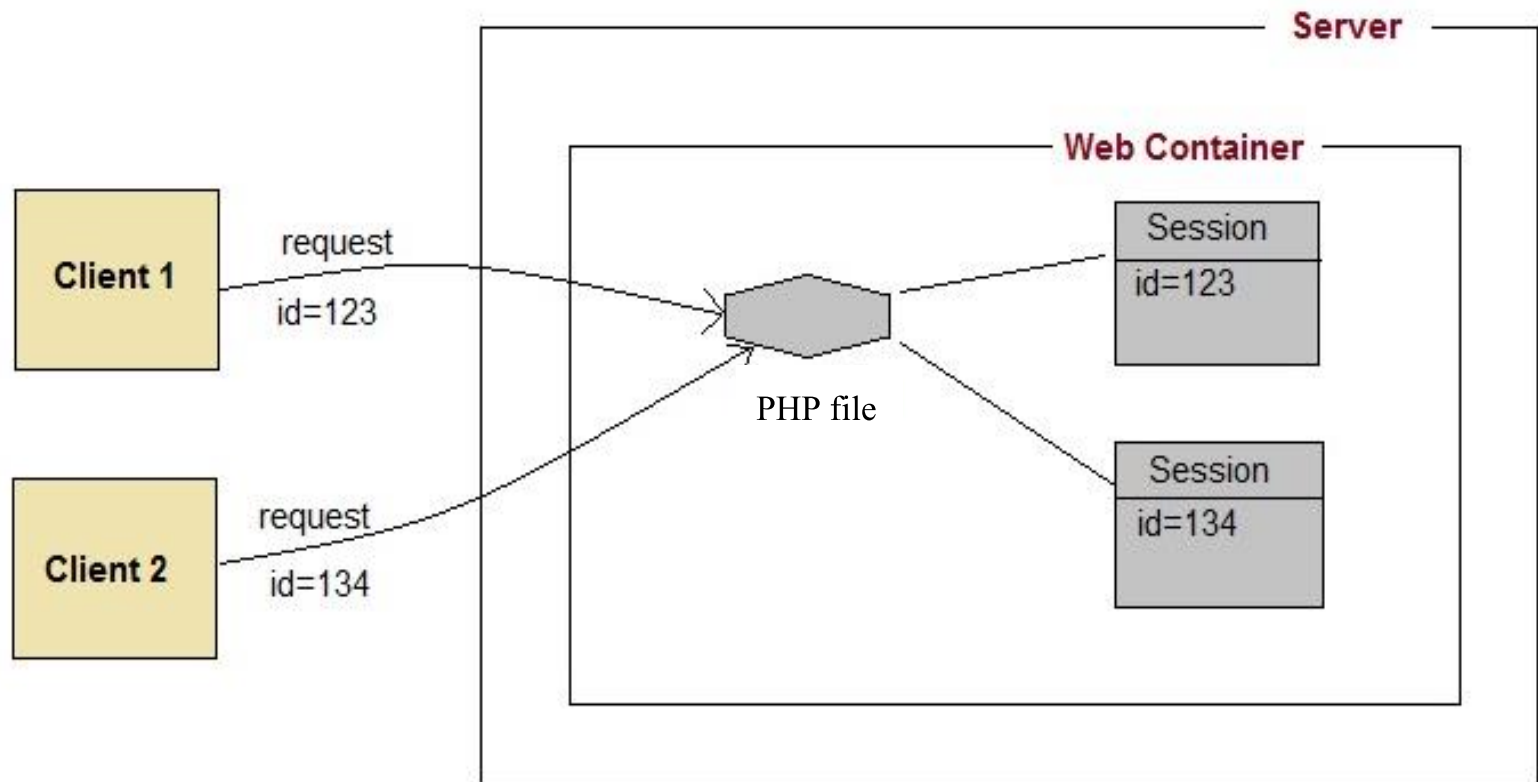
## Session

- การเก็บข้อมูลใน Cookie มีขนาดจำกัด คือไม่เกิน 4096 bytes เว็บไซต์ที่ต้องการเก็บข้อมูลมากกว่า เช่น ตะกร้าสินค้า อาจเก็บไม่พอ จึงใช้ Session เก็บข้อมูลแทน
- Session คือ object ที่สร้างขึ้นบน Server สำหรับแต่ละ client ที่ส่ง request เข้ามายัง server เพื่อเก็บข้อมูลบางอย่าง
- 1 client ต่อ 1 session object
- หาก client เดิมติดต่อมายัง server จะดึง session object เดิมขึ้นมา โดยไม่มีการสร้าง object ใหม่
- Session object จะสามารถถูกดึงออกมาใช้ในเว็บเพจหน้าใดก็ได้ (ที่เขียนด้วย PHP)



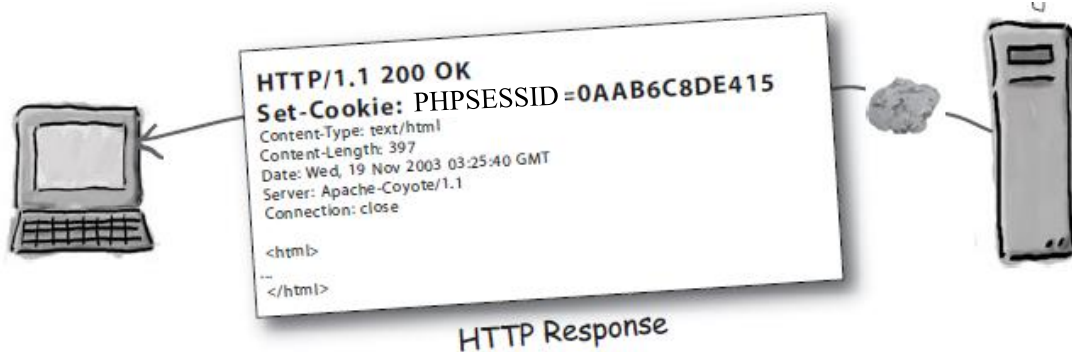
# ➔ Session

- เมื่อ client ส่ง request มายัง server ในครั้งแรก web server จะสร้าง Session ID ที่ไม่ซ้ำกันขึ้นมาเพื่อเป็นตัวแทนของ client นั้น



# ➔ Session ID

- Session ID หรือรหัส Session ในภาษา PHP จะถูกเก็บไว้ใน Cookie ชื่อ PHPSESSID บนเครื่องของ client อัตโนมัติ



response ครั้งแรก จะส่งรหัส Session มาเขียนลงบน Cookie โดยที่นักพัฒนาไม่ต้องใช้คำสั่งเขียนข้อมูลใน cookies เอง



request ครั้งต่อไป รหัส Session ใน Cookie จะถูกแนบมากับ request ด้วย

## ➡ คำสั่งในการสร้าง Session Object

- เมื่อต้องการสร้าง Session จะใช้ฟังก์ชัน ดังนี้

`<?php session_start(); ?>` ← เรียกฟังก์ชัน ก่อนอ่านหรือเขียนข้อมูลลง session

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

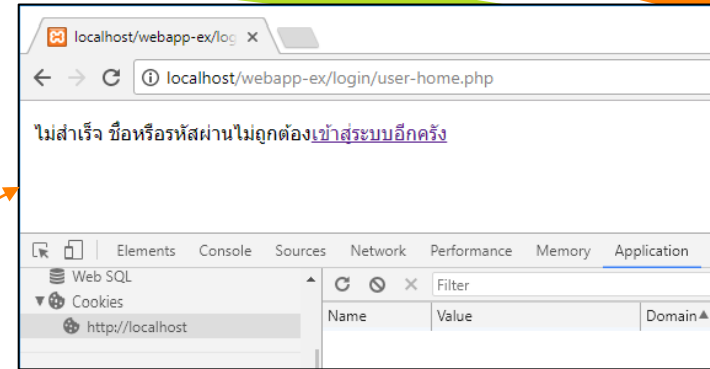
- เว็บไซต์ใดที่ใช้คำสั่งนี้ หมายถึง จะมีการสร้าง Session ID กรณีที่ยังไม่เคยสร้าง แต่ถ้าเคยสร้างแล้วจะใช้ Session ID เดิม หลังจากนั้นจะสามารถอ่านและเขียนข้อมูลบนตัวแปร session ได้

# ➡ คำสั่งในการสร้าง Session Object

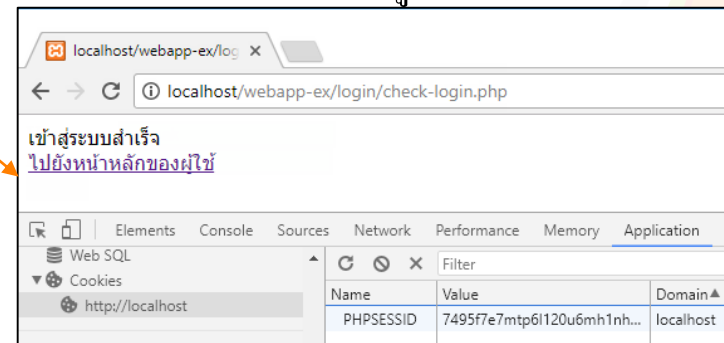
- เมื่อใช้เมธอด `session_start()`; หากเป็นการเรียกครั้งแรกจะเกิดการทำงานอัตโนมัติดังนี้
  - Web Server จะสร้าง Session ID ที่ไม่ซ้ำกันขึ้นมา
  - Web Server จะสร้าง Cookie object และเก็บค่า Session ID เก็บใน Cookie เพื่อส่งไปกับ response
  - Web Server จะสร้าง Session object สำหรับ client นั้น
- หากเคยเรียกเมธอด `session_start()`; แล้ว จะค้นหา Session ID จาก Cookie ที่มากับ request และดึงเอา Session object เดิมขึ้นมา
- การทำงานที่เกี่ยวกับ Cookie จะอยู่เบื้องหลังทั้งหมด นักพัฒนาไม่ต้องจัดการเอง

# ระบบ Login

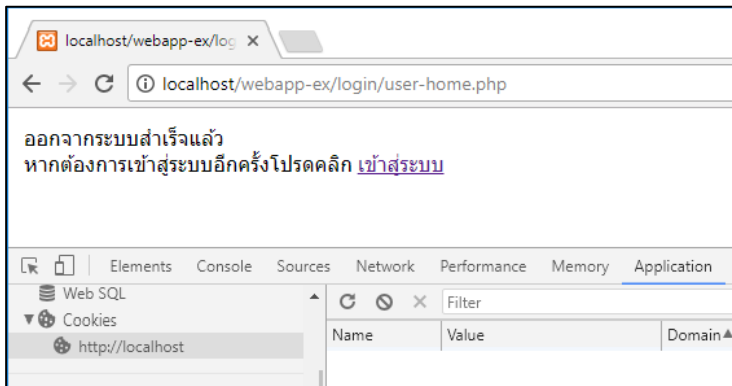
กรอก username และ password ในแบบฟอร์ม HTML



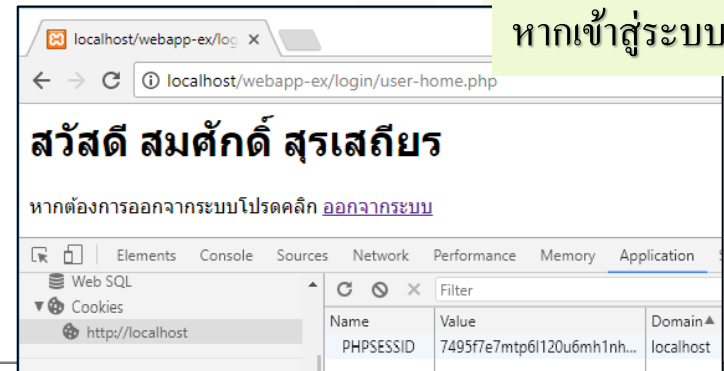
หน้าจอหากเข้าสู่ระบบสำเร็จ



หากผู้ใช้คลิกออกจากระบบ จะหน้าจอดังนี้



หน้าจอที่ผู้ใช้จะเห็นได้ หากเข้าสู่ระบบสำเร็จ



# ➡ สร้างฟอร์มสำหรับ Login

ไฟล์ login-form.php

```
<html>
```

```
<body>
```

```
<form action="check-login.php" method="POST">
```

```
  Username: <input type="text" name="username"><br>
```

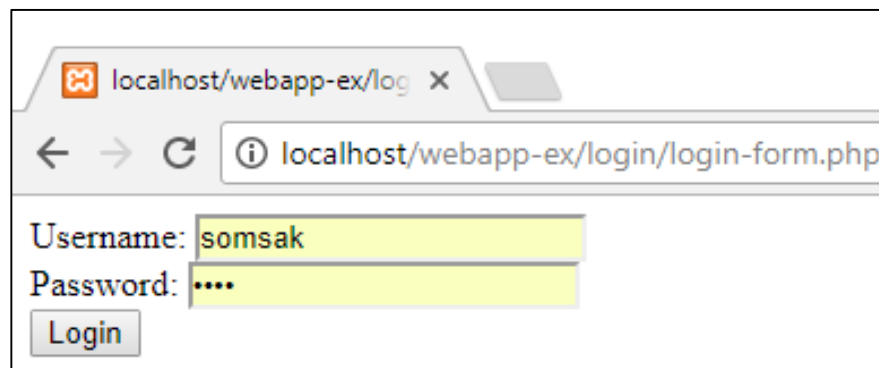
```
  Password: <input type="password" name="password"><br>
```

```
  <input type="submit" value="Login">
```

```
</form>
```

```
</body>
```

```
</html>
```



A screenshot of a web browser window. The address bar shows the URL 'localhost/webapp-ex/login/login-form.php'. The page content displays a login form with two input fields: 'Username:' containing the text 'somsak' and 'Password:' containing four dots. Below the password field is a 'Login' button.



# ตรวจสอบ username/password

ไฟล์ check-login.php

```
<?php
include "connect.php";
session_start();

$stmt = $pdo->prepare("SELECT * FROM member WHERE username = ? AND password = ?");
$stmt->bindParam(1, $_POST["username"]);
$stmt->bindParam(2, $_POST["password"]);
$stmt->execute();
$row = $stmt->fetch();
```

```
// หาก username และ password ตรงกัน จะมีข้อมูลในตัวแปร $row
if (!empty($row)) {
```

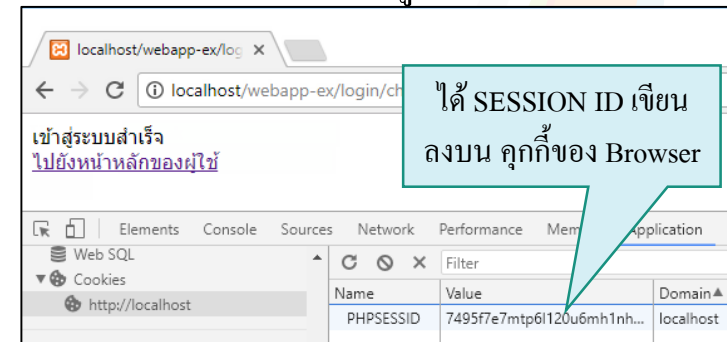
```
    // เปลี่ยน session id ป้องกัน Session Fixation
    session_regenerate_id();
```

```
    // นำข้อมูลผู้ใช้จากฐานข้อมูลเขียนลง session 2 ค่า
    $_SESSION["fullname"] = $row["name"];
    $_SESSION["username"] = $row["username"];
```

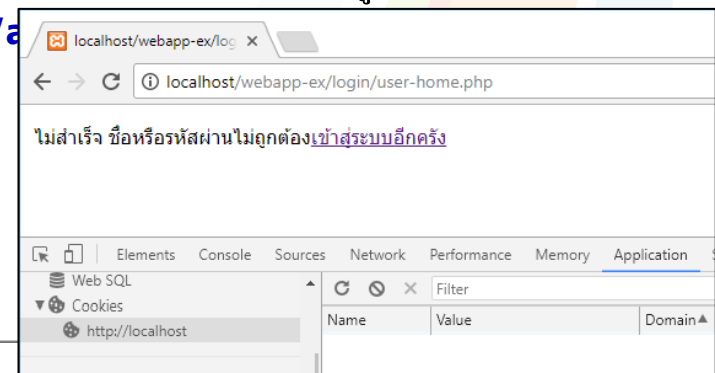
```
    // แสดง link เพื่อไปยังหน้าต่อไปหลังจากตรวจสอบสำเร็จแล้ว
    echo "เข้าสู่ระบบสำเร็จ<br>";
    echo "<a href='user-home.php'>ไปยังหน้าหลักของผู้ใช้</a>";
```

```
// กรณี username และ password ไม่ตรงกัน
} else {
    echo "ไม่สำเร็จ ชื่อหรือรหัสผ่านไม่ถูกต้อง";
    echo "<a href='login-form.php'>เข้าสู่ระบบอีกครั้ง</a>";
}
?>
```

หน้าจอหากเข้าสู่ระบบสำเร็จ



หน้าจอหากเข้าสู่ระบบไม่สำเร็จ



# ➡ นำ session ชื่อผู้ใช้ไปใช้ในหน้าอื่นๆ

ไฟล์ user-home.php

```
<?php session_start(); ?>
```

ประกาศว่าหน้านี้จะใช้ตัวแปร session

```
<html>
```

```
<body>
```

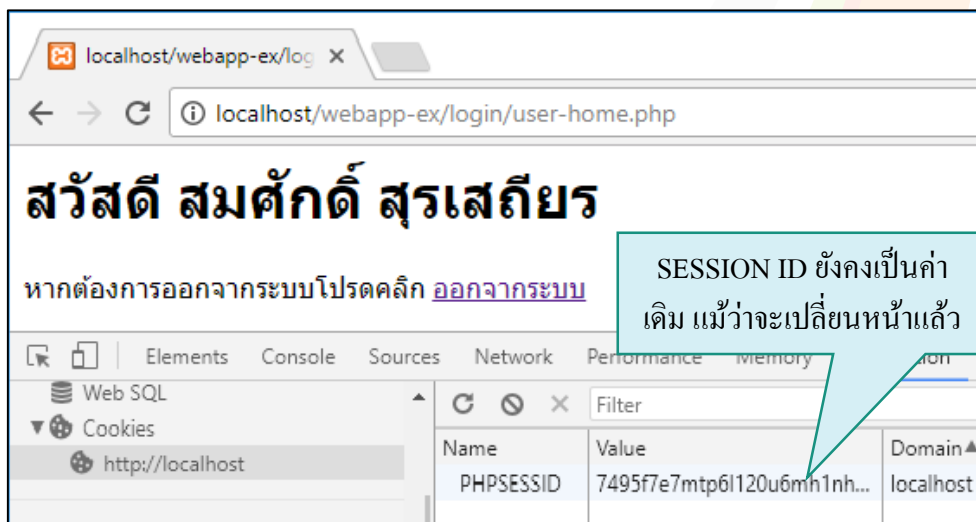
ดึงค่าจาก Session มาแสดง

```
<h1>สวัสดี <?=$_SESSION["fullname"?></h1>
```

```
หากต้องการออกจากระบบโปรดคลิก <a href='logout.php'>ออกจากระบบ</a>
```

```
</body>
```

```
</html>
```





# ➡ การป้องกันการเข้าหน้าเว็บ

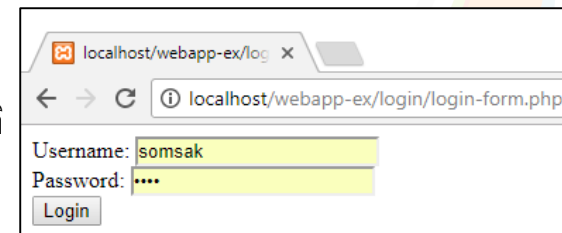
ไฟล์ product-list.php

```
<?php
include "connect.php";
session_start();
// ตรวจสอบว่ามีชื่อใน session หรือไม่ หากไม่มีให้ไปหน้า login อัตโนมัติ
if (empty($_SESSION["username"])) {
    header("location: login-form.php");
}
?>
```

ไปยังหน้า login อัตโนมัติ

```
<html>
<head><meta charset="utf-8"></head>
<body>
```

```
<?php
$stmt = $pdo->prepare("SELECT * FROM product");
$stmt->execute();
while ($row = $stmt->fetch()) {
    echo "ชื่อสินค้า: " . $row ["pname"] . "<br>";
    echo "ราคา: " . $row ["price"] . " บาท <br>";
    echo "<hr>\n";
}
?>
</body></html>
```



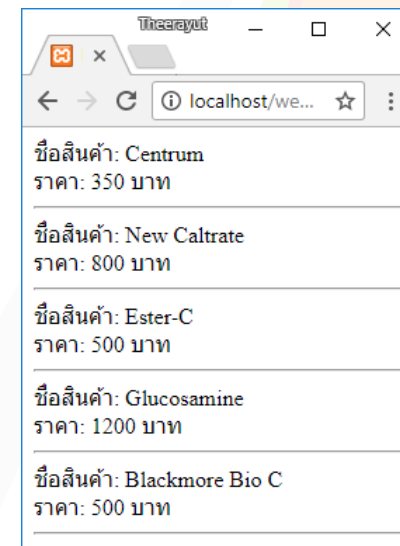
localhost/webapp-ex/log x

localhost/webapp-ex/login/login-form.php

Username: somsak

Password: ....

Login



ชื่อสินค้า: Centrum	ราคา: 350 บาท
ชื่อสินค้า: New Caltrate	ราคา: 800 บาท
ชื่อสินค้า: Ester-C	ราคา: 500 บาท
ชื่อสินค้า: Glucosamine	ราคา: 1200 บาท
ชื่อสินค้า: Blackmore Bio C	ราคา: 500 บาท

การมีค่า username ใน session แล้วให้ทำงานโค้ดส่วนที่เหลือได้

## ➡ การ Logout

- การ Logout คือ การทำลาย session ที่สร้างไว้

<?php

```
session_start();
```

ไฟล์ logout.php

```
// ลบ session id ในคุกกี้เครื่องผู้ใช้
```

```
$params = session_get_cookie_params();
```

```
setcookie(session_name(), "", time() - 42000,
```

```
    $params["path"], $params["domain"],
```

```
    $params["secure"], $params["httponly"]
```

```
);
```

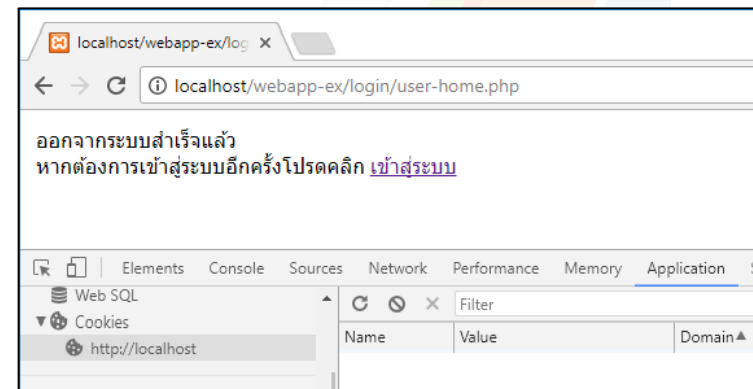
```
session_destroy(); // ทำลาย session
```

>

ออกจากระบบสำเร็จแล้ว<br>

หากต้องการเข้าสู่ระบบอีกครั้งโปรดคลิก

<a href='login-form.php'>เข้าสู่ระบบ</a>



## ➡ ช่วงอายุของ Session

- การกำหนดอายุของ Session แต่ละ Server จะแตกต่างกันไป แต่ใน Apache Web Server จะกำหนดในไฟล์ php.ini

`session.cache_expire = 180`

หน่วยเป็น นาที