

## الفصل التاسع

### بدايتك إلي ال session

ال session هي عبارة عن تقنية للترابط مع المستخدم وهي موجودة ضمن ال PHP4 ولم تكن موجودة ضمن الإصدارات التي قبله بل كان يجب أن تقوم بتركيب مكتبة لكي تستطيع استخدام هذه التقنية، يعتمد فهمنا لل session على فهمنا للكوكيز وكيفية استعمالها ولقد تكلمنا عن الكوكيز بشكل جيد في الفصل السابق، يستخدم ال session لعمل ميكانيكية تواصل بين المستخدم والسيرفر، فلقد قلنا أن ال http لا يوفر لنا ميكانيكية لعمل تواصل، فإذا طلب المستخدم صفحة من السيرفر فإن السيرفر يقوم بإعطائه ما أراد وينتهي عند ذلك فلا يعرف إن كان هو نفس المستخدم أو ليس هو... لأجل ذلك تم إنشاء تقنية ال session لأجل عمل تقنية تواصل بين المستخدم والموقع، فباستطاعتك مثلاً أن تقوم بتحديد عدد زيارات مستخدم معين لصفحتك ليوم واحد أو لأسبوع أو لمدة معينة من الوقت.... أو يمكنك عمل متجر إلكتروني بسيط يستطيع المستخدم شراء عدة أشياء دفعة واحدة من الموقع ويكون على تواصل بينه وبين الموقع عندما يقوم بإضافة مشتري إلى سلة التسوق أو حذف مشتريات.

قبل أن أتكلم عن كيفية استخدام ال Session وإعطاء بعض الأمثلة البسيطة، سأقوم بالتكلم عن كيفية إعداد ال session مع ال PHP.

### إعدادات ال session في ال PHP

لكي تستطيع التعامل مع ال session بشكل جيد يجب عليك أن تتعرف على بعض الإعدادات التي في ملف ال php.ini

عندما تفتح الملف ستجد قسماً خاصاً فيه بال-session هناك حوالى ١٩ إعداد ولكن لن نتطرق إليها كلها بل سنتكلم عن الأساسية والمهمة منها فقط كبداية لنا للتعرف على ال-session وكيفية عمله.

## إعداد طريقة التخزين

```
session.save_handler (files | mm | user)
```

ستجد هذه العبارة مكتوبة في الملف كالتالي: بشكل افتراضي:

```
session.save_handler = files
```

وهذا الإعداد يقوم بتحديد طريقة التخزين لل-session وهناك ثلاثة حالات للتخزين:

١ - التخزين في ملفات عادية على السيرفر:

```
session.save_handler = files
```

٢ - التخزين على ذاكرة السيرفر:

```
session.save_handler = mm
```

٣ - التخزين بطريقة أخرى معرفة ومعينة من قبل المستخدم مثل التخزين في قواعد البيانات وهذا ما سوف نقوم بالتفصيل به بعد الكلام عن قواعد البيانات:

```
session.save_handler = user
```

يجب أن تأخذ في اعتبارك عدد الملفات التي سيقوم ال-session بتخزينها عند استخدامك للأعداد الأول والافتراضي خاصة عندما يكون عدد الزوار بالمئات أو الآلاف.

قد يكون استعمال الذاكرة أسرع ولكن المشكلة أنه من السهل مسح البيانات منها ببساطة.

الطريقة الثالثة قد تكون أكثر الطرق مرونة، ولكنها معقدة وصعبة جداً، وهي تعطيك مرونة لتخزين البيانات في أي وسائط مدعومة من قبل ال-PHP مثل قواعد بيانات mysql و oracle.

الذي افترضه الآن أنك قمت بوضع قيمة هذه الخاصية إلى files .

## إعداد مكان التخزين

```
session.save_path (path/to/directory)
```

هذه الخاصية مفيدة إذا كنت قد ضبطت الإعداد السابق إلى files تقوم هذه الخاصية بتحديد مكان التخزين على السيرفر ومن الأفضل أن تقوم بتحديد مكان التخزين بعيداً عن مجلد السيرفر لكي تمنع تصفح هذه الملفات.

## الإنشاء التلقائي لـ session

```
session.auto_start (0 | 1)
```

هذا الإعداد يقوم بتحديد إذا ما كان الـ session سيتم إنشاؤه تلقائياً عند كل زيارة للموقع أو لأي صفحة من صفحاته بدون إدراج كود الـ session في كل صفحة... وعلى ذلك فإنك تقوم بوضع القيمة إلى (1) إذا أردت ذلك. وعلى افتراض أنك لا تحتاج إلى أن تجعل الـ PHP يقوم بعمل session لكل صفحة تلقائياً ومن غير طلب فستقوم بوضع قيمة هذا الإعداد إلى (0).

## الـ SID

عندما يقوم الزائر بزيارة صفحتك فإن الـ session يستطيع تتبع هذا الزائر وعدد المرات التي قام فيها الزائر بالدخول لليوم الواحد، يقوم الـ PHP بعمل SID (session identifier) أو رقم معرف تلقائي بشكل افتراضي عندما تقوم بطلب إنشاء session بالزائر، وكل رقم معرف يختلف عن الآخر تماماً، إن رقم المعرف الذي ينشئه الـ PHP شبيه للشكل التالي:

```
fc94ad8b1ee49ef79c713ee98ac1fcc4
```

هناك طريقتان يستطيع بهما الـ PHP متابعة الـ SID للمستخدم:

- 1 - عن طريق المتابعة والتخزين بتسلسل في الكوكيز.
- 2 - عن طريق اتباع رقم المعرف بعنوان الصفحة في الإنترنت.

سنأخذ أمثلة عن كلا الطريقتين:

## ١ - استخدام الكوكيز

بالطبع هذه هي أكثر الطرق شيوعاً للحصول على ترابط بين المستخدم والموقع وهي الأسهل، ولكن يجب أن تضع في اعتبارك أن المستخدم قد يكون قد ألغى أو منع ميزة الكوكيز في المتصفح أو قد يكون متصفحه لا يدعم الكوكيز. خذ في اعتبارك أن بعض المتصفحات لا تسمح بأن يزيد حجم الكوكيز عن ٥ كيلوبايت.

هناك بعض الإعدادات البسيطة في ملف `php.ini` التي يجب معرفة معلومات عنها قبل البدء باستخدام الـ `session` مع الكوكيز:

```
session.use_cookies (0 | 1)
```

هذه الخاصية تحدد ما إذا كان يمكنك استخدام الكوكيز مع الـ `session` أو لا وعند وضع القيمة (٠) فهذا يمنعك من استخدام الكوكيز مع الـ `(session)` وأما إذا كانت قيمته (١) فهذا يسمح باستخدام الكوكيز مع الـ `session`

```
session.name (Default: PHPSESSID)
```

هذا الإعداد يقوم بتحديد اسم الكوكيز الذي سيحتفظ برقم المعرف (SID) والإعداد الافتراضي هو `PHPSESSID` ولن أقوم بتغيير هذا الإعداد لكي تستطيع فهم المثال الذي سأطرحه بعد قليل.

```
session.cookie_lifetime (Default: 0)
```

يقوم هذا الإعداد بتحديد المدة التي سيبقى فيها الكوكيز الذي يحتفظ بقيمة الـ (SID) والإعداد الافتراضي هو صفر، أي أنه سيتم مسح الكوكيز تلقائياً بعد إغلاق المستخدم لنافذة المتصفح مباشرة.

```
session.cookie_path (Default: /)
```

يقوم هذا الإعداد بتحديد مسار دومين يتم تخزين الكوكيز له.. لا تتم بتغيير قيمته ودعه كما هو.

```
session.cookie_domain (Default: null)
```

يقوم هذا الإعداد بتعريف اسم دومين يتم تخزين الكوكيز لصالحه.. والقيمة الافتراضية هي null، لا تتم بتغييرها

ضع في اعتبارك أنه إذا كانت قيمة الأعداد (session.use\_cookies) تساوي واحد فإنه لا داعي لاستدعاء الدالة (set\_cookie) لإعداد الكوكيز بل سيتم إعدادها تلقائياً بواسطة الـ PHP

## ٢ - الإضافة أو الكتابة إلى عنوان الصفحة

إن إضافة عنوان الـ SID إلى عنوان الصفحة يعتبر من الأشياء البشعة جداً رغم أن طريقته سهلة ومفيدة في حالة ما إذا كان الكوكيز غير مدعوم في المتصفح بشكل جيد.

مثال:

```
<a href="configure.php?<?=SID?>">Go to the configuration page</a>
```

بهذه الطريقة نقوم بإضافة المتغير المرجعي SID الذي سيقوم بإعطاء رقم معرف للمستخدم.

## منابة الـ session

لقد أخذنا حتى الآن معلومات جعلنا ندخل عالم البرامج المسيرة بالـ session بدون خوف، سأبدأ الآن بطرح بعض الأمثلة البسيطة التي تثبت لديك بعض المفاهيم الأساسية في الـ session... سأشرح في هذا المثال كيفية إنشاء الـ SID وتخزينه لاستعماله لاحقاً، وخلاصة السيناريو للصفحة أننا نريد من المستخدم أن يفهم أنه يستطيع تخصيص لون الخلفية الذي يريد أن يشاهد به صفحات موقعنا... سأقوم

بتخزين قيمة مبدئية في المتغير الذي يقوم بتحديد لون الصفحة، أنا أفترض طبعاً أن المتصفح يدعم الكوكيز:

## سكربت يقوم بإنشاء وتسجيل متغير session

```
<?
session_start();
session_register("zx");
session_register("co");
$zx=10;
$co++;
echo "مرحباً بك في موقعنا أيها الزائر الكريم";
"echo عدد زيارتك لهذه الصفحة; $co ".
echo "<br>";
"<a href="php2.php">echo 'الصفحة الثانية';
?>
```

أقصد بالجلسة هي الـ (session) وإن كانت الترجمة غير صحيحة ولكن فقط نأخذه كمصطلح.

متغير الجلسة هو الـ (session-variable) أو متغير الـ session أو سمه ما شئت.

### الشرح

يقوم هذا السكربت في البداية بإنشاء متغير اسمه (ZX) ومتغير اسمه (CO) وقمنا بإعطاء القيمة (١٠) للمتغير (ZX) وقمنا بزيادة القيمة الموجودة (وهي الصفر) في (CO) بواحد وكتبنا مرحباً بك أيها الزائر الكريم في موقعنا، ثم قلنا له إن عدد زيارتك لهذه الصفحة هي قيمة المتغير (CO) ثم أعطيناه رابط للصفحة الثانية. في الواقع إن هذه المتغيرات وقيمها يتم الاحتفاظ بها في كوكيز له اسم خاص قمنا بتحديد سابعاً من ملف PHP.ini، وهذا الكوكيز يحتفظ بقيمة الـ SID لـ session.

نحن لا نقوم بإخبار الـ PHP أين سيحتفظ بقيمة المتغيرات لأننا بدأنا بكلمة الـ:

```
session_start();
```

وعلى هذا فإن الـ PHP سيفهم أنه سيقوم بتخزين القيمة في الكوكيز الخاص بالـ session.

قمنا بجعل المتغير CO كعداد بسيط لعدد المرات التي سوف نقوم بها بزيارة الصفحة فعند عمل تحديث للصفحة سيتم زيادة العداد بمقدار واحد

```
$c++;
```

وطبعاً قبل زيادة العداد بقيمة واحد فإنه يتم حساب القيمة السابقة للمتغير عند إنشائه تلقائياً... ومن ثم تتم الزيادة وبعد ذلك طباعة القيمة.

### كتابة رقم الـ SID

اكتب الآن الكود التالي واحفظه باسم php2.php

```
<?
session_start();
echo $PHPSESSID."<br>";
echo $zx;
?>
```

في هذه الصفحة نقوم بطباعة قيمة الـ SID وذلك بطباعة قيمة المتغير \$PHPSESSID (الذي هو اسم الكوكيز الخاص بالـ session).

بعد ذلك قمنا في النهاية بطباعة قيمة المتغير \$zx لكي ألفت نظرك بأن الكوكيز ما زال يحتفظ بها ولم يفقدها لأننا قد حددنا الإعداد في ملف الـ php.ini الخاص بوقت الكوكيز الـ 3600 أي لمدة ساعة ثم بعد تلك الساعة سيتم مسح الكوكيز ولن يمكنك من استرجاع قيمة أي متغير:

```
session.cookie_lifetime = 3600
```

واضف إلى معلوماتك أنه لا يمكنك قراءة القيم للكوكيز الخاص بالـ session إلا عن طريق إضافة الأمر

```
session_start();
```

يجب أن تبدأ بهذا الأمر دائماً إذا أردت قراءة قيم المتغيرات التي يحتفظ بها الكوكيز الخاص بالـ session.

## session

كل ما عليك فعله هو استخدام هذه الدالة:

```
session_unregister(variable name);
```

تقوم بوضع اسم المتغير في مكان الـ (variable name)

مثال:

```
session_unregister("brn");
```

سيقوم هذا الأمر بمسح المتغير (brn) من الكوكيز الخاص بالـ (session)

قراءة قيم المتغيرات في الكوكيز الخاصة بالـ session

كل ما عليك فعله هو استخدام الدالة:

```
session_encode();
```

مثال:

```
<?
session_start();
session_register("bgcolor");
session_register("name");
session_register("email");
$bgcolor = "#8080ff";
$name = "alfareees almolthem";
$email = "php@php.com";
$e = session_encode();
print "The encoded string is: $e";
?>
```

بهذا السكريبت نكون قد أنهينا فصلنا عن مقدمة بسيطة لـ session. هذه مجرد مقدمة ولكي نستطيع أن نتعمق بالـ session فيجب علينا أن نتعلم شيئاً عن قواعد البيانات.