

# Sessioner i PHP5

---

Jonas Björk

April 2008

# Licens

Upphovsrätten till ett verk uppkommer automatiskt, utan att man behöver anmäla eller registrera sitt verk. Upphovsrätten är lagstiftad i Sverige och gäller i 70 år från upphovsrättsinnehavarens död. Verk som omfattas av upphovsrätt får inte utan upphovsrättsinnehavarens tillstånd publiceras på en hemsida eller på annat sätt. Verket får heller inte skrivas ut utan upphovsrättsinnehavarens tillstånd.

Istället för att krångla med lagar ger jag Dig, skriftligen i detta dokument, tillstånd att:

- Läs dokumentet.
- Skriva ut dokumentet.
- Kopiera dokumentet och sprida det vidare till dina vänner.
- Publicera ett oförändrat exemplar av dokumentet på din egen webbplats.

Däremot tillåter jag inte att Du till exempel:

- Kopierar innehållet och publicerar det som "ditt".
- Skriver ut och säljer kopior av dokumentet, inte ens till självkostnadspris.
- Tar betalt för kopior av det elektroniska dokumentet.
- Använder dokumentet i undervisning.

Vill du använda dokumentet till något annat än det jag uttryckligen tillåter ovanför ber jag dig ta kontakt med mig på [jonas@jonasbjork.net](mailto:jonas@jonasbjork.net) för att komma överens om hur detta kan genomföras.

PRODUKTER OCH FÖRETAGSNAMN SOM NÄMNS I DETTA DOKUMENT KAN VARA VARUMÄRKEN ELLER REGISTERADE VARUMÄRKEN SOM TILLHÖR RESPEKTIVE ÄGARE SOM ANVÄNDS FÖR ATT IDENTIFIERA DERAS PRODUKTER OCH TJÄNSTER.

UTÖVER VAD SOM UTTRYCKLIGEN SKRIFTLIGEN ÖVERENSKOMMITS ELLER KRÄVS ENLIGT LAG TILLHANDAHÅLLS VERKET I "BEFINTLIG SKICK", UTAN NÅGRA SOM HELST GARANTIER, VARKEN UTTRYCKLIGA ELLER INDIREKTA, UTAN NÅGRA BEGRÄNSNINGAR AVSEENDE GARANTIER AVSEENDE INNEHÅLLET ELLER KORREKTHETEN I VERKET.

©2008 Jonas Björk, [www.jonasbjork.net](http://www.jonasbjork.net)

## Innehållsförteckning

---

Sessioner.....	4
Starta en session.....	4
Lagra data i en session.....	5
Hämta data från en session.....	5
Ta bort variabler från en session.....	5
Avsluta en session.....	6
Exempel.....	6
Visa hur många gånger besökaren varit på sidan.....	6
Ett enkelt lösenordsskydd.....	7
Sessionens namn och id.....	8
Lagen om elektronisk kommunikation.....	8

# Sessioner i PHP5

---

## Sessioner

---

Webbsidor skickas från en webbserver med hjälp av Hyper Text Transfer Protocol (**HTTP**). HTTP är ett förbindelseöst protokoll, vilket innebär att när klienten (webbläsaren) har fått det den efterfrågade från servern (webbservern) dör förbindelsen. Problem som uppstår med detta är att vi inte kan hålla reda på en användare som besöker vår webbplats. Vi kan inte veta om användaren har loggat in på webbplatsen, eller till exempel vilka varor hon lagt i kundkorgen. Varje förbindelse är, från webbserverns sida, helt ny.

Detta kan man lösa med hjälp av kakor (eng *cookies*) eller sessioner (eng *sessions*). Kakor har nackdelen av att de sparar all information hos klienten. Det är inte enkelt, men det går, att manipulera en sådan kaka. Vi skulle till exempel kunna manipulera vår lokala kaka och säga att vi är inloggade som administratör på webbplatsen.

En session lagras på servern och är således betydligt mycket svårare att manipulera. Sessionen skickar en kaka till klienten där den lagrar ett sessions-id, det vill säga vilken session klienten har. Varje gång klienten ansluter till webbplatsen skickas sessions-id:t med i form av en kaka till servern och servern kan därför spåra användaren och hålla koll på vilken användare som är vem. Genom detta kan vi till exempel skapa webbplatser med användarkonton eller kundvagnar.

## Starta en session

---

En session startas första gången vi använder **session\_start()** . **session\_start()** skapar en ny session om det inte redan finns en för användaren, om det finns en session kommer den befintliga att hämtas upp och användas. Detta är det första som skall göras på varje sida i webbapplikationen. Utan **session\_start()** är inte phpskriptet knuten till en session och vi kan därför inte heller använda oss av sessionsvariablerna på sidan.

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

Det finns ett par undantag till när **session\_start()** inte skall stå först i phpskript. Ett av dem är när vi lagrar objekt i sessionerna. Om vi har valt att lagra ett (eller flera) objekt i en session måste vi inkludera klassdefinitionen innan **session\_start()** ! Detta för att PHP skall förstå hur objektet fungerar.

```
<?php
    include( "MinKlass.class.php" );
```

```
session_start();
?>
```

## Lagra data i en session

---

När vi har en session kan det vara lämpligt att lagra något i den. Vad vi lagrar i sessionen är, för PHP, helt ointressant. Det är vi som vill hålla något *levande* mellan användarens klick på webbplatsen. All data vi skapar i en session är åtkomlig genom variabeln `$_SESSION`. `$_SESSION` är en array och vi lagrar data i den genom att skapa plats för det i arrayen. Det kan till exempel se ut så här:

```
<?php
session_start();
$_SESSION['namn'] = "Jonas";
?>
```

Här skapar vi sessionsvariabeln `$_SESSION['namn']` och tilldelar den värdet *jonas*. Vi lagrar alltså textsträngen *jonas* i variabeln.

## Hämta data från en session

---

Att hämta upp data från sessionsvariablerna är lika enkelt som att ange deras namn. Vi tittar på ett exempel:

```
<?php
session_start();
?>
<p>Ditt namn är <?php echo $_SESSION['namn']; ?>.</p>
```

## Ta bort variabler från en session

---

Vill vi ta bort variabler från vår session använder vi phpfunktionen `unset()`.

```
<?php
session_start();
unset( $_SESSION['namn'] );
?>
```

När vi använder funktionen `unset()` kommer enbart den variabel vi anger tas bort, de andra (om det finns andra) kommer finnas kvar i sessionen. Vill vi istället ta bort alla variabler i sessionen använder vi istället funktionen `session_unset()` som tömmer variabeln `$_SESSION` helt.

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

Notera att `session_unset()` inte avslutar sessionen, det gör funktionen `session_destroy()`.

## Avsluta en session

När vi vill avsluta en session på servern använder vi PHP-funktionen **session\_destroy()**. Det är ganska enkelt och vi ger ett exempel på ett utloggningsförfarande här nedanför.

```
<?php
    session_start();
    session_unset();
    unset( $_COOKIE[session_name()] );
    session_destroy();
    header( 'Location: index.php' );
?>
```

Vi måste alltså öppna sessionen först (**session\_start()**) för att sedan avsluta den med funktionen **session\_destroy()**. Innan vi avslutar sessionen ser vi till att tömma sessionen på variabler (**session\_unset()**) och radera kakan hos klienten (**unset( \$\_COOKIE[session\_name()] )**). När vi har avslutat sessionen skickar vi användaren vidare till en annan sida (**index.php**) som i vårt fall är en startsida för webbapplikationen.

En session lever som standard så länge webbläsaren är igång hos klienten. Livslängden definieras i konfigurationsfilen **php.ini** där den som standard är satt till värdet 0, som betyder *tills webbläsaren stängs*. Vill vi ändra sessioners livslängd kan vi ändra på värdet vid **session.cookie\_lifetime** i filen **php.ini** till det antal sekunder vi vill hålla en session levande.

## Exempel

Här följer några exempel på hur sessioner kan användas praktiskt i PHP.

### Visa hur många gånger besökaren varit på sidan

Det första exemplet vi skall titta på är en enkel räknare som visar hur många gånger användaren besökt vår sida.

```
<?php
    session_start();

    if( !isset( $_SESSION['raknare'] ) ) {
        $_SESSION['raknare'] = 1;
    } else {
        $_SESSION['raknare']++;
    }

?>
<p>Du har besökt denna sida <?php echo $_SESSION['raknare']; ?> gånger.</p>
```

I exemplet använder vi variabeln **\$\_SESSION['raknare']** som en räknare för hur många gånger användaren besökt vår sida. Om det är första gången användaren besöker oss existerar naturligtvis inte **\$\_SESSION['raknare']** och det kontrollerar vi genom programkoden **if(!isset(\$\_SESSION['raknare']))** om det är sant (att den inte finns) skapar vi den och tilldelar den värdet 1 med **\$\_SESSION['raknare'] = 1**

Om vi däremot har en **\$\_SESSION['raknare']** i vår session kommer vi istället öka värdet i den med 1 : **\$\_SESSION['raknare']++** . Att ange ++ efter en variabel är samma sak som att skriva **\$\_SESSION['raknare'] = \$\_SESSION['raknare'] + 1** , det är bara lite kortare att skriva ++.

## Ett enkelt lösenordsskydd

I det här exemplet kommer vi skapa ett enkelt sätt att lösenordsskydda våra hemsidor. Vi bygger upp applikationen med tre filer: **start.php** som är startsidan, **inne.php** som är den lösenordsskyddade sidan och **loggaut.php** som används för att logga ut användaren. Vi börjar med att titta på **start.php**:

```
<?php
    session_start();

    $error = false;

    if( isset( $_POST['losenordKnapp'] ) ) {
        $l = $_POST['losenord'];
        if( $l == "hemligt" ) {
            $_SESSION['inloggad'] = true;
        } else {
            $error = true;
        }
    }

    if( isset( $_SESSION['inloggad'] ) ) {
        if( $_SESSION['inloggad'] == true ) {
            header( 'Location: inne.php' );
        }
    }
}

?>

<p>Välkommen till min hemliga sida, du måste ange lösenordet för att komma in här!
</p>

<?php
    if($error) {
        echo "<p>Fel lösenord!</p>";
    }
}

?>
<form method="post" action="start.php">
<p>Lösenord: <input type="password" name="losenord" />
<input type="submit" name="losenordKnapp" value="Logga in" />
</p>
</form>
```

Sedan tittar vi på filen **inne.php** som är den sida vi inte kommer åt utan att ange lösenordet:

```
<?php
    session_start();

    if( !isset( $_SESSION['inloggad'] ) ) {
        header( 'Location: start.php' );
    }

    ?>
<p>Välkommen!</p>
<a href="loggaut.php">Logga ut</a>
```

Slutligen är det filen **loggaut.php** som hanterar utloggningen av användaren:

```
<?php
    session_start();
    session_unset();
    unset( $_COOKIE[session_name()] );
    session_destroy();
    header( 'Location: start.php' );
    ?>
```

Om vi vill ha flera sidor som är lösenordsskyddade skiver vi samma kontroll i dem som i filen

inne.php :

```
if( !isset( $_SESSION['inloggad'] ) ) {  
    header( 'Location: start.php' );  
}
```

## Sessionens namn och id

När vi skapar en session med PHP får den ett namn, vanligtvis PHPSESSID. Detta namn kan vi ändra på i konfigurationsfilen för PHP (**php.ini**) om vi vill. Vi letar upp raden i **php.ini** som ser ut så här: **session.name = PHPSESSID** och ändrar PHPSESSID till något annat.

När sessionen skapas får den en unik sträng som skiljer sessionerna mellan. Denna sträng lagras i en kaka hos klienten och vi kan få fram den genom variabeln **\$\_COOKIE[session\_name()]**. Ett exempel på hur det kan se ut i kod:

```
<?php  
    session_start();  
?>  
Sessionsnamn: <?php echo session_name(); ?><br />  
Sessionsid: <?php echo $_COOKIE[session_name()]; ?>
```

När vi kör det här skriptet i webbläsaren kommer vi se något liknande:

```
Sessions-namn: PHPSESSID  
Sessions-id: 7a8tse11mmohcavhjbhlm9akc5
```

Notera att sessionsid kommer vara tom första gången vi besöker sidan. Anledningen till detta är att kakan skickas till klienten *efter* att skriptet körts. Vi har alltså ingen kaka första gången vi kör skriptet och därför kommer inte **\$\_COOKIE[session\_name()]** att innehålla något. Vi uppdaterar webbsidan en gång i webbläsaren så kommer sessions-id att visas.

## Lagen om elektronisk kommunikation

Den 25 juli 2003 trädde Lagen om elektronisk kommunikation i kraft. I lagen står det bland annat att alla som besöker en webbplats som använder kakor skall få information om att webbplatsen använder kakor, vad kakorna används till och hur användaren kan undvika kakor.

Det är alltså viktigt att vi, när vi använder kakor/sessioner informerar användarna om detta!

Så här skriver den svenska linuxgemenskapen linuxportalen.se på sin webbplats:

*linuxportalen.se använder cookies för att du skall kunna göra egna val på sidan, så som att anpassa sidans innehåll. Vill du inte använda cookies, kan du stänga av dem i din webbläsare - se din webbläsares hjälp för hur du gör. Stänger du av cookies kommer vissa av sidans funktioner inte fungera som tänkt.*

Läs mer om detta på Post och Telestyrelsens hemsida: <http://pts.se/sv/Regler/Lagar/Lag-om-elektronisk-kommunikation/Cookies-kakor/>