I gang med sessioner i ASP

I denne artikel gennemgås en sessioners funktionalitet på forskellig vis. Det giver en hjælpende hånd hvis man vil lave sider der gemmer information om brugeren i kort tid, og f.eks beskytte sine sider med adgangsniveuer, som her er kaldt levels.

Skrevet den **05. Feb 2009** af **nidyahou** I kategorien **Programmering / ASP** | ★☆☆☆☆

Hvad er en session?
kort fortalt er en session en mændge data der følger brugeren rundt på din side. Dette kan blive nyttigt, hvis man skal have sikkerheds informationer med rundt på siden.

Dit første sessions arbejde
Vi skal nu til at arbejde med sessioner for første gang. Vi lægger blødt ud, og laver en session der gemmer information om brugerens browser. Dette kan f.eks. bruges hvis du vil lave browsertilpassede sider.

```
<%
'sessionen browser's værdi sættes nu til brugerens browser
Session("browser") = (Request.ServerVariables("http_user_agent"))
%>
```

Gem brugerinformation
Vi skal nu til at gemme information om en bruger i en session, dette kan bruges hvis du har services, der kræver at man er logget ind.

```
<form action="session.asp" method="post">
Brugernavn<input type="text" name="brugernavn"><br>
Kodeord<input type="password" name="password">
</form>
```

session filen ser nu sådan her ud.

```
<%
'sessionen brugernavn defineres
session("brugernavn") = Request.form("brugernavn")
'sessionen password defineres
session("password") = Request.form("password")
%>
```

Hvis du f.x har et forum hvor man kan oprette tråde kan du jo lave et felt til dette hvor du indsætter brugernavn på forhånd så det ligner det her.

```
<input type="text" name="forfatter" value="<%session("brugernavn")%>" disabled>
```

OBS: Dette er ikke sikkerhedsmæssigt klogt, da man med nogle programmer kan ændre i en HTML kode, derfor bør du bruge session("brugernavn") direkte i koden der bliver indsat i din databasen.

Indholds beskyttelse med sessioner
Nu kan vi altså gemme brugerinformation i en session, men nu begynder vi at sikre indholdet med sessioner. Lav en kolonne i din brugerdatabase der hedder level, og så kan du sætte brugerens session til rs("level"), husk at level er et reserveret ord, så du når du skriver det frit, uden "'er om så skal der stå [level]. Vi laver nu et eksempel på en level sikring.

```
<%
'Sessionen level defineres når brugeren logger på
session("level") = rs("level")

'Nu tjekker vi om besøgeren har level 4 eller højere
If session("level") => 4 then
Response.write "Du er level 4 eller højere, du må se indholdet af siden"
else
Response.Write "Du er ikke level 4 endnu, du må ikke se indholdet af siden"
end if
%>
```

Tjek session ID
Alle brugere på din side får et unikt session id, hvis du får brug for dette i forbindelse med engangsdata kan du tjekke dette med følgende kode:

```
<%
Response.Write Session.SessionID
%>
```

Sessions udløbstid
Når du bruger sessioner til at gemme brugerinformation sklal du være opmærksom på, at de automatisk sletter sig selv efter 20 minutter. Antallet af minutter kan ændres, men vær opmærksom på, at for mange sessioner på en gang belaster systemet.

```
<%
'sessionernes udløbstid tjekkes nu
Response.Write Session.Timeout
'sessionernes udløbstid ændres nu til 40 minutter
Session.Timeout = 40
%>
```

Forlad session
Det kan være en god idé manuelt at slette din session f.eks. i forbindelse med at logge ind på en ny bruger eller en bruger der forlader din side. Vær opmærksom på at sessionen ikke bliver slettet før at alle andre session kommandoer er eksekveret, så placering af linien er ligegyldig, og du kan godt bede om data fra en session på samme side som du sletter sessionen.

```
<%
'Sessionerne slettes
Session.Abandon
%>
```

Opsummering
Hvis vi nu skal prøve at benytte det vi nu har lært kan vi lave et lille eksempel.

```
<%
'sessionen klokken defineres
session("klokken") = Now
'sessionen bruger defineres
session("bruger") = "Andrew"
'sessionen level defineres
session("level") = rs("level")

'sessionernes udløbstid sættes til 30 minutter
Session.Timeout = 30

'Når brugeren har bevæget sig rundt på siden og vil se sin information kan du vise denne således
Response.Write "Dit unikke id er: " & Session.SessionID & ", dit brugernavn er "_
& session("bruger") & ", du loggede ind kl." session("klokken") & ", og du er i level "_
& session("level") & ". Hvis du er inaktiv i " & Session.Timeout & " minutter, så bliver du automatisk
logget af."

'brugeren får adgang efter det level han har
response.redirect "level.asp?level=" & session("level")

'Til sidst logger brugeren ud
Session.Abandon
%>
```

Artiklen rediregeres løbende.

Kommentar af ismand d. 25. Apr 2006 | 1

Kommentar af cyigen d. 30. Mar 2004 | 2

Fin artikel, men synes kun du berører emnet overflødisk. Du uddyber f.eks. ikke hvorfor for mange Session'er belaster systemet, og hvor mange er for mange.

Kommentar af cpufan d. 24. Apr 2004 | 3

Af hastighedsgrunde og for kompatibilitetens skyld bør man uddybe til at kalde variablen for session.contents("etellerandet") . Også fordi det faktisk er den korrekte tilgang til samlingen.

Kommentar af thesurfer d. 07. Jun 2004 | 4

Lille fejl (der skal stå <%=session) ved name="forfatter" value="<%session -- Ser godt ud -- Har ikke læst artiklen igennem -- derfor 50% -- ændres ved gennemlæsning.

Kommentar af morhan d. 15. Apr 2006 | 5

Et eksempel plejer at være noget man kan copy/paste, og så virker det. Jeg synes ikke din sikkerhed er

gennemført.. Hvad er pointen med din session, hvis du overfører level i urlen: level.asp?level=x, kan man få fuld adgang, hvis man sammensætter sin egen url?