

Come utilizzare le sessioni ASP

Sessioni

La sessione è utilizzata per memorizzare informazioni sulla *sessione utente*.

Le variabili memorizzate nell'**oggetto Session** mantengono informazioni su un singolo utente, e sono disponibili a tutte le pagine dell'applicazione.

Sessioni

Quando lavoriamo con un'applicazione, la apriamo, effettuiamo dei cambiamenti e poi la chiudiamo.

Una sessione fa più o meno le stesse cose.

Il computer sa chi sei, sa quando inizi una sessione e quando la termini. Ma su internet c'è un problema: il webserver non sa chi sei perché http non mantiene lo stato.

ASP risolve il problema creando un *cookie unico per ogni utente*. Il cookie è inviato al client e contiene informazioni che identificano l'utente. Questa interfaccia è detta **oggetto Session**.

L'oggetto Session è utilizzato per memorizzare o cambiare le informazioni circa la sessione di un singolo utente. Le variabili memorizzate nell'oggetto Session hanno delle informazioni sul singolo utente e sono disponibili per tutte le pagine dell'applicazione.

Informazioni che comunemente sono memorizzate nelle variabili di sessione sono nome, id, e preferenze. Il server crea un oggetto Session per ogni nuovo utente e distrugge l'oggetto Session quando la sessione termina.

Quando inizia una sessione?

Una sessione inizia quando:

- Un nuovo utente richiede un file ASP, ed il file Global.asa include la **procedura *Session_OnStart***
- Un valore viene memorizzato in una **variabile di sessione**
- Un nuovo utente richiede un file ASP, e il file Global.asa utilizza il **tag *<object>*** per istanziare un oggetto con visibilità di sessione

Quando termina una sessione?

Una sessione termina se un utente **non ha richiesto pagine o fatto il refresh** in una pagina dell'applicazione **per un determinato lasso di tempo**. Di default questo tempo è 20 minuti.

Se si vuole cambiare l'intervallo di default, occorre impostare la proprietà Timeout.

L'esempio imposta il timeout a 5 minuti:

```
<% Session.Timeout=5 %>
```

Per terminare una sessione immediatamente si può utilizzare il **metodo *Abandon***:

```
<% Session.Abandon %>
```

Nota: Il problema principale è QUANDO far terminare una sessione. Non sappiamo se la richiesta dell'utente è l'ultima o ne farà ancora, e dobbiamo tenere viva la sessione, tuttavia con le sessioni utilizziamo le risorse del server.

Consiglio: Memorizzare pochi dati nelle variabili di sessione.

Memorizzare e recuperare le variabili di sessione

L'esempio spiega come memorizzare delle variabili di sessione. Imposteremo la variabile di sessione username a "Donald Duck" e la variabile age a "50":

```
<% Session("username")="Donald Duck" Session("age")=50 %>
```

Queste due variabili possono essere recuperate in qualsiasi parte dell'applicazione:

```
Benvenuto <% Response.Write(Session("username")) %>
```

La riga precedente stampa su schermo: "Benvenuto Donald Duck".

Si possono anche memorizzare le preferenze degli utenti, nell'esempio controlliamo se l'utente vuole visualizzare la pagina a bassa o alta risoluzione:

```
<%  
If Session("screenres")="low" Then  
%>  
Versione a bassa risoluzione  
<%Else%>  
Versione ad alta risoluzione  
<%End If%>
```

Eliminare le variabili di sessione

La **collezione Contents** contiene tutte le variabili di sessione.

È possibile rimuovere una variabile di sessione con il *metodo Remove*.

L'esempio rimuove la variabile di sessione "sale" se il valore di "age" è minore di 18:

```
<%  
If Session.Contents("age")<18 then  
    Session.Contents.Remove("sale")  
End If  
%>
```

Per rimuovere tutte le variabili di sessione si può utilizzare il *metodo RemoveAll*:

```
<% Session.Contents.RemoveAll() %>
```

Ciclare il contenuto della collezione Contents

La collezione Contents contiene tutte le variabili di sessione. Si può ciclarne il contenuto per vedere cosa c'è memorizzato:

```
<%  
Session("username")="Donald Duck"  
Session("age")=50  
dim i  
For Each i in Session.Contents  
    Response.Write(i & "<br />")  
Next  
%>
```

Risultato: username
age

Se non conosciamo il numero di elementi nella collection, possiamo utilizzare la *proprietà Count*:

```
<%  
dim i  
dim j  
j=Session.Contents.Count  
Response.Write("Variabili di sessione: " & j & "<br />")  
For i=1 to j  
    Response.Write(Session.Contents(i) & "<br />")  
Next  
>%
```

Risultato: Variabili di sessione: 2
Donald Duck
50

Ciclare la collection StaticObjects

Si può ciclare la **collection StaticObjects**, per vederne i valori:

```
<%  
dim i  
For Each i in Session.StaticObjects  
    Response.Write(i & "<br />")  
Next  
>%
```

Esempi con uso di Jscript

https://www.lukeonweb.net/leggi/144/oggetto_session.asp

Esempi con uso di JavaScript

<http://www.marinsalta.net/asp/corsoasp3.htm>

[Articolo](#), [guida](#), slides con illustrazione degli altri oggetti ASP interni ([.pdf](#))