Corso basi di dati
Gli oggetti interni ASP

Gianluca Di Tomassi
Email: ditomass@dia.uniroma3.it

Università di Roma Tre

Cosa sono gli oggetti ?

Gli oggetti sono costituiti dagli elementi che li descrivono e dalle operazioni che possono essere effettuate utilizzandoli.

- Gli elementi che descrivono gli oggetti sono denominati *proprietà*
- Le operazioni che possono essere effettuate utilizzandoli sono denominate *metodi*

Oggetti direttamente disponibili in ASP (1)

ASP mette a disposizione per la gestione dell'output, dell'input e delle operazioni che il client vuole o deve compiere sul server diversi oggetti predefiniti.

Application: utilizzato per condividere informazioni tra numerosi client che esplorano lo stesso gruppo di pagine

Session: informazioni sul singolo utente che accede a un'applicazione

Request: utilizzato per recuperare informazioni passate dal client al server dal browser

Response: utilizzato per inviare manda al client il codice HTML risultante

Oggetti direttamente disponibili in ASP (2)

Server: mette a disposizione alcune funzionalità del server

ObjectContext: permette di gestire le transazioni (collega la pagine ASP e Microsoft Transaction Server)

ASPError: consente di ottenere informazioni relative agli errori degli script all'interno delle pagine

Ogni oggetto, ha una collezione di metodi, proprietà ed eventi

Collection

Rappresenta un insieme di coppie di nomi o valori

La stringa d'interrogazione:

"?nome=Gianluca&cognome=DiTomassi"

Contiene due coppie di nomi e valori, la prima coppia ha il nome "nome" e il valore "Gianluca". La seconda ha il nome "cognome" e il valore "Di Tomassi"

Questi dati (vedremo) vengono memorizzati nella collection QueryString dell'oggetto Request

Altri oggetti utili

Altri oggetti utili nella gestione delle applicazioni:

Oggetto ADO:	Gestisce le collezioni della tecnologia ADO. Nel seguito lo useremo per le operazioni con i database
Oggetto BrowserCap:	Permette di riconoscere il browser e le impostazioni (ad esempio la risoluzione,il numero di colori etc) che il visitatore sta utilizzando, adattando il sito al software dell'utente
Oggetto FileSystem:	Permette di manipolare files e directories

L'oggetto Response

Consente di inviare informazioni al browser e di controllare il modo in cui vengono inviate.

Sintassi oggetto Response:

Response.*collection*/*property*/*metodo*

➢Le ***collection*** dell'oggetto Response sono:

***Cookies*:** Si usano per creare i cookie

➢Le ***property*** per l'oggetto Response sono:

***CacheControl*:** determina se il proxy riesce a mantenere in cache il contenuto delle pagine ASP

***CharSet*:** Specifica il set di caratteri utilizzato

***Expires*:** Specifica il tempo che deve trascorrere prima che una pagina presente nella cache del browser venga rimossa

Esempi:

`<% Response.Expires = numero %>` la pagina scade dopo *numero* minuti

`<% Response.Expires = -1500 %>` la pagina scade immediatamente

***ExpiresAbsolute*:** Specifica la data e l'ora quando la pagina deve essere cancellata dalla cache del browser

```
<% Response.ExpiresAbsolute = Data Time %>
```

Se *Data* viene omessa, si utilizza la data corrente

Se *Time* viene omessa, la pagina memorizzata scade alle 24:00

Esempio: `<%Response.ExpiresAbsolute = #2 giu 2001 18:16:00#%>`

ContentType: E' una stringa che descrive i contenuti supportati dal documento

Esempio: `<% Response.ContentType = "text/plain"  %>`

Buffer: Gestisce la bufferizzazione del codice da inviare all'utente. (occorre settarlo dopo l'Option Explicit)

- Per default è settato a FALSE
- Se viene settato a TRUE tutti gli script saranno processati prima che qualsiasi cosa venga inviata al browser

Esempio: `<% Response.Buffer = True %>`

➢I ***metodi*** dell'oggetto Response sono:

AddHeader: pone il valore tra i tag <Header> ad un valore *value*

Write: invia info all'utente (scrive una stringa in output). La stringa scritta non può contenere il tag "%>", se è necessario scriverlo utilizzare "%\>"

`<% Response.Write(expr) %>` è equivalente a `<%=expr%>`

Esempi di utilizzo:

```
Response.Write("<IMG SRC='banner.gif'>")
Response.Write(Server.HTMLEncode("<IMG SRC='banner.gif'>"))
```

Redirect: permette di reindirizzare il browser ad un altro URL

- In questo modo è possibile fare dei controlli sulla navigazione degli utenti e pilotarla
- E' possibile effettuare la Redirect solo se il contentuto del file non è già stato inviato al browser oppure utilizzando il buffering

Esempi: `<% Response.Redirect "homepage.asp" %>`

`<% Response.Redirect = "http://www.dia.uniroma3.it" %>`

AppendToLog: Aggiunge una stringa al log del Web server per una interrogazione

Clear: cancella qualsiasi codice HTML nel buffer di trasmissione

End: Viene interrotta l'esecuzione della pagina ASP e restituisce il risultato corrente (cioè se il buffering è attivato e sono presenti dati nel buffer, questi vengono inviati)

Flush: Invia immediatamente il contenuto del buffer di trasmissione anche se si sta bufferizzando (utile quando si desidera inviare la maggior parte possibile dell'output al client.

NOTA: Response.Clear e Response.Flush producono un messaggio di errore quando il buffering viene disattivato

L'oggetto Request

Consente di accedere ai dati che il client ha inviato quando ha richiesto la pagina corrente

Sintassi oggetto Request:

Request.*collection*/*property*/metodo (*Variabile*)

Le ***collection*** dell'oggetto Request sono:

ClientCertificate: I valori dei campi memorizzati nei certificati del client, che vengono mandati durante una richiesta via HTTP

Cookies: Si usa per determinare il valore dei cookies

Form: contiene tutte le informazioni che un utente inserisce in una form tramite il metodo POST

Esempio:

```
<form action= "send.asp" method="post">
    <p> Sito preferito: <select nome= "sito"> <option>
    ...
    Il tuo sito preferito è <% Request.Form("sito") %>
```

QueryString: Contiene tutte le informazioni passate come parametro dopo ? nell'URL (quindi passate con metodo GET)

Esempio:

```
<a href= "saluto.asp?nome=Gianluca&eta=27">
   Ciao <%= Request.QueryString("nome") %>
   hai  <%= Request.QueryString ("eta") %> anni
```


ServerVariables: Fornisce informazioni ricavate dall'intestazione http che poi vengono trattate come variabili d'ambiente del Web server

Esempio: `<% Request.ServerVariables(nomeVarAmbiente) %>`

Variabili d'ambiente	Descrizione
SERVER_PORT	contiene il numero della porta su cui è stata fatta la richiesta
HTTP_ACCEPT_LANGUAGE	contiene il linguaggio del documento
SCRIPT_NAME	è il nome dello script
HTTP_HOST	è il nome del dominio associato all'indirizzo IP
HTTP_USER_AGENT	è lo user agent del browser utilizzato
CONTENT_LENGTH	informazioni sulla lunghezza del pacchetto

QUERY_STRING	La stringa d'interrogazione (equivalente a **`Request.QueryString`**)
REMOTE_HOST	fornisce l'indirizzo IP
REMOTE_ADDR	fornisce l'indirizzo dell'host remoto
URL (uguale a PATH_INFO)	L'URL della pagina ASP a partire dalla fine di "http:www.WebServer.it/" fino alla stringa d'interrogazione
PATH_TRASLATED	Il percorso fisico completo della pagina ASP attualmente in esecuzione
SERVER_NAME	Il nome del computer del Web server
SERVER_SOFTWARE	Il nome del SW del Web server
APPL_PHISICAL_PATH	L'indirizzo fisico delle dir principali del Web Server
ALL_RAW	Fornisce tutte le informazioni

Le ***property*** per l'oggetto Request sono:

TotalBytes Read-only: Restituisce il numero di byte spediti da un client durante una richiesta al server

I ***metodi*** dell'oggetto Request sono:

BinaryRead: Restituisce i dati spediti al server da un client come parte di un POST

Il parametro *variabile* è una stringa che specifica il valore da utilizzare in una colection o che deve essere usata come input per un metodo o una property.

Se la variabile non è presente quando si usa una delle 5 collection viste, l'oggetto Request restituisce il valore EMPTY

Tutte le variabili, possono essere visualizzate direttamente senza il nome della collection mediante l'istruzione:

```
Request(variabile)
```

In questo caso, il sever cerca il valore della variabile, analizzando le collection nell'ordine che segue: QueryString, Form, Cookies, ClientCertificate, ServerVariables.

L'oggetto Server

Fornisce l'accesso ad alcuni strumenti di base sul server
Sintassi oggetto Server:

Server.*property*/*metodo*

Le ***property*** per l'oggetto Server sono:

ScriptTimeout: il tempo massimo, in secondi, che uno script può funzionare,prima che venga "disattivato" dal server

I ***metodi*** dell'oggetto Server sono:

CreateObject: crea un istanza di un componente del server
Esempio: `<% Server.CreateObject(NomeComponente) %>`

Execute: esegue una pagina ASP (utile rispetto all'include perché il nome del file può essere generato dinamicamente)

HTMLEncoding: Codifica la stringa argomento in modo che il browser non la interpreti come HTML

Esempio: `<% Server.HTMLEncoding(Stringa) %>`

MapPath: Stabilisce una corrispondenza fra il percorso virtuale specificato, sia relativo, sia assoluto, e il percorso fisico.

Esempio: `<% Server.MapPath(path) %>`

URLEncode: Applica le regole di encoding, inclusi i caratteri di escape, ad una stringa in modo che essa possa esser posta in una stringa d'interrogazione

GetLastError: Restituisce un'istanza dell'oggetto ASPError che descrive l'ultimo errore avvenuto

L'oggetto Session

Trasporta i valori richiesti da un singolo client nell'intera sessione, che può essere di diverse pagine.

Il Web server attiva un'istanza dell'oggetto Session, ogni volta che un utente accede ad una pagina.

Il server poi, distruggerà l'istanza una volta che l'utente si disconnette o dopo un certo tempo di timeout.

Attraverso l'uso di Session è possibile memorizzare le preferenze di ciascun utente collegato.

Sintassi oggetto Session :

Session.*collection*.*property*/metodo

Le ***collection*** per l'oggetto Session sono:

Contents: Contiene gli ITEMS che sono stati aggiunti alla sessione con i comandi di script e quindi diverse da un oggetto

StaticObjects: Contiene tutti gli oggetti creati con il tag <OBJECT>

Le ***property*** per l'oggetto Session sono:

CodePage: Il codice della pagina che sarà usato per il mapping simbolico del sito

LCID: L'identificatore locale per la sessione

SessionID: Restituisce un ID di sessione per l'utente connesso nella sessione avviata

Timeout: Indica il tempo, in minuti, prima del Timeout della sessione in corso.

I ***metodi*** dell'oggetto Session sono:

Abandon: Questo metodo, distrugge un oggetto Session e rilascia tutte le risorse tenute dall'oggetto fino a quel momento.

Contents.Remove: Rimuove l'elemento dalla collection Contents

Esempio: `<% Session.Contents.Remove(ele) %>`

Contents.RemoveAll: Rimuove tutti gli elementi dalla collection Contents

Session possiede anche la gestione di **eventi**. Essi sono gestibili nel file global.asa attraverso i seguenti script:

Session_OnStart: Si verifica quando si crea una nuova sessione

Session_OnEnd: Si verifica quando termina una sessione, a causa di Abandon o per Timeout

L'oggetto Application

ù utenti che visitano lo stesso gruppo di pagine. Solo un'istanza dell'oggetto Application è creata per una applicazione ed è condivisa fra tutti i client che accedono a quell'

Application:

Application. ***collection*** **/** ***metodo***

Le ***collection*** dell'oggetto Application sono:

Contents: Contiene tutte le variabili dell'applicazione diverse da un oggetto

StaticObjects: Contiene tutti gli oggetti dell'applicazione

I ***metodi*** dell'oggetto Application sono:

Contents.Remove: Rimuove l'elemento dalla collection Contents

Contents.RemoveAll: Rimuove tutti gli elementi dalla collection Contents

Lock: Impedisce a tutti gli altri client di modificare i valori dell'oggetto Application

Unlock: rilascia il bloccaggio e consente a tutti gli altri client di modificare i valori nell'oggetto Application

Application, come Session, possiede anche la gestione di **eventi**. Essi sono gestibili nel file global.asa attraverso i seguenti script:

Application_OnStart: Si verifica quando si avvia un'applicazione prima di avviare la sessione

Application _OnEnd: Si verifica quando termina l'applicazione, dopo che tutte le sessioni sono terminate

L'oggetto ASPError

E' un nuovo oggetto di IIS 5.0 e consente di ottenere informazioni sugli errori che si sono verificati nello scriptFornisce l'accesso ad alcuni strumenti di base sul server

Sintassi oggetto ASPError:

ASPError.*property*

Le ***property*** per l'oggetto Server sono:

ASPCode: Restituisce una stringa con il codice di errore

ASPDescription: Restituisce una lunga stringa che descrive l'errore che si è verificato

Category: Restituisce una stringa che indica se l'errore è generato dal linguaggio script, da una pagina ASP o da un oggetto

Column: Restituisce il numero di colonna responsabile dell'errore

Description: Breve stringa che descrive l'errore avvenuto

File: restituisce una stringa che indica il nome del file che ha causato l'errore

Line: Restituisce il numero di riga che ha causato l'errore

Number: Restituisce il numero di errore restituito da un componente COM

Source: Restituisce il codice che ha causato l'errore