

# Tecniche per aggirare lo "stateless" HTTP

L'*HTTP* è, per sua natura, un **protocollo "stateless"** (*senza stato*).

Il *Server Web* opera nella logica *request-response*, ossia risponde alle richieste, **senza avere memoria** di quanto realizzato prima.

Al momento dell'invio della pagina al client (*response*), il server **termina l'esecuzione** del codice e **dealloca tutte le variabili e gli oggetti** usati fino a quel momento.

Ciò significa che eventuali *dati memorizzati nella RAM* del *Server Web* (*variabili, oggetti, ecc.*) vengono **cancellati** all'invio della risposta, anche nel caso la richiesta sia un *PostBack*.

Questo, per le moderne *Web-App* non è accettabile, per cui esistono delle **tecniche specifiche** per aggirare il problema.

# Tecniche per aggirare lo "stateless" HTTP

## Utilizzo di **FILE** o di **DATABASE** sul Server

Il server web memorizza i dati di cui evitare la perdita, in un File o un DataBase situati su una memoria permanente a lui accessibile, evitandone così la cancellazione.

## **Passaggio di Parametri** da una Pagina all'altra

Il server web trasmette i dati da non perdere, a un'altra pagina web attraverso i parametri previsti nelle classiche operazioni di GET e POST del protocollo HTTP.

## Invio di **Dati in Controlli ASPX** (rientrano con il Postback)

Il server web pone il dati da non perdere, in un controllo ASPX e invia la pagina al client: il dato sopravvive perché, in caso di Postback, esso viene "ritrasmesso" automaticamente al server. Ovviamente il dato viene perso in caso di un nuovo request iniziale alla pagina.

# Tecniche per aggirare lo “stateless” HTTP

## Utilizzo dei **Cookies**

Un server web può inviare al client un piccolo insieme di dati (chiamato Cookie) che il browser memorizza sul dispositivo client, in un piccolo file permanente. Durante le successive request (verso lo stesso dominio), il client ritrasmette il Cookie, consentendo al server di recuperare i dati precedentemente inviati. I Cookies sono spesso usati per “riconoscere” il client o per memorizzare impostazioni personali dell'applicazione, per il client in questione.

## Utilizzo delle **SESSIONI**

Il server web gestisce delle speciali aree dati, chiamate Aree di Sessione, che, anche attraverso l'uso dei Cookies, vengono associate al browser che sta navigando. Un dato posto nell'Area di Sessione non solo sopravvive al response, ma è disponibile per tutte le successive richieste provenienti del browser associato.

## **Variabili Globali** in una Applicazione ASPX

Le applicazioni web basate su ASPX consentono di definire e gestire delle Variabili Globali create al momento dell'attivazione dell'applicazione sul Server Web e sempre disponibili durante l'esecuzione del codice delle pagine dell'applicazione stessa.

# Oggetti Request e Response nel codice ASPX

## L'oggetto **Request**

L'oggetto Request è sempre disponibile nel codice C# di una pagina ASPX: grazie ai suoi numerosi Metodi, Proprietà ed Eventi, consente di accedere a tutte le informazioni riguardanti la Request pervenuta al server e di gestire tutte le operazioni inerenti la Request stessa.

## L'oggetto **Response**

L'oggetto Response è sempre disponibile nel codice C# di una pagina ASPX: grazie ai suoi numerosi Metodi, Proprietà ed Eventi, consente di accedere a tutte le informazioni riguardanti la Response e di gestire tutte le operazioni inerenti la Response stessa, inclusa la modifica/costruzione della pagina da inviare al client.