

Mechanizm Sesji – Sprawozdanie

Maciej Nowak

1. Wstęp

W niniejszym sprawozdaniu, omówię mechanizm sesji w oparciu o technologię ASP.NET oraz PHP. Na wstępie wyjaśnię czym jest sesja oraz gdzie ma ona zastosowanie. W dalszej części pokażę działanie swojej aplikacji wykorzystującej mechanizm sesji. Przedstawię mechanizm sesji traktując osobno ASP.NET oraz PHP.

2. Czym jest sesja?

W tym paragrafie przedstawię mechanizm sesji. Odpowiem na takie pytania jak: Jak definiujemy pojedynczą sesję? Do czego służą sesje?

Protokół http jest protokołem bezstanowym. Zatem serwer każde żądanie traktuje jako niezależne. Prowadzi to więc do komplikacji w budowie aplikacji, portali czy internetowych systemów informacyjnych. Z pomocą przychodzi mechanizm sesji. Sesja jest traktowana jako pewien określony odcinek czasu począwszy od utworzenia sesji przez wysłanie żądania do momentu końca czasu sesji określonego przez programistę lub wcześniejszego zamknięcie przeglądarki przez użytkownika. W obrębie sesji są przechowywane dane przypisane do konkretnej sesji. Znikają one w momencie zakończenia sesji. Sesje więc umożliwiają przesyłanie i zapamiętywanie danych między żądaniami. Dane sesji są przechowywane po stronie serwera. Zatem sesję nie mają ograniczeń po stronie klienta związanych z przeglądarką (z wyjątkiem przechowywania opcji przechowywania plików sesji w cookies).

3. Działanie mechanizmu sesji w praktyce – czyli moja własna aplikacja

W tym miejscu omówię działanie własnej aplikacji mechanizmu sesji wykonanej w technologii ASP.NET oraz PHP.

Moja aplikacja jest złożona z trzech podstron:

- a. **Index.php / Main.aspx** – strona główna aplikacji
- b. **Info.php / Info.aspx** – strona informacyjna (liczniki) aplikacji
- c. **Finish.php / Finish.aspx** – strona końcowa aplikacji

Projektowanie Internetowych Systemów Informacyjnych - Mechanizm Sesji

Autor: Maciej Nowak

Witaj na stronie głównej projektu. Aby poznać działanie aplikacji, wpisz swoje imię i nazwisko oraz wciśnij przycisk DALEJ. Zostaniesz przeniesiony do okna w którym ujrzysz liczniki aplikacji oraz sprawdzisz działanie zmiennych sesyjnych. Czas sesji: 60sek

Imię:

Nazwisko:

Rys. 1. Okno strony startowej index.php / main.aspx

W tym miejscu aplikacja rozpoczyna nową sesję (jeśli bieżąca sesja nie istnieje). Użytkownik wypełnia pola Imię oraz Nazwisko, wciska przycisk dalej. W ten sposób zostały pobrane dane wpisane do pól i przypisane do zmiennych sesyjnych. Użytkownik zostaje przeniesiony do kolejnej strony.

Nazywasz się: Jan Kowalski

Statystyki strony:

Ilość wizyt: 3

Ilość aktywnych użytkowników: 1

Ilość wyświetleń strony INFO w bieżącej sesji: 10

Rys. 2. Okno strony info.php / info.aspx

Strona info wyświetla wpisane wcześniej dane, zapamiętane w zmiennych sesyjnych. Poniżej pokazuje statystyki dotyczące wszystkich wizyt od momentu uruchomienia aplikacji, ilość aktywnych sesji oraz ilość wyświetleń strony informacyjnej dla bieżącej sesji. W momencie kliknięcia przycisku ZAKOŃCZ SESJE, użytkownik zostaje przekierowany do strony finish.php / finish.aspx. Sesja zostaje zakończona.

Zakończyłeś sesję. Aby powrócić do strony głównej i rozpocząć nową sesję kliknij w STRONA GŁÓWNA

Rys. 3. Okno strony finish.php / finish.aspx

W tym miejscu, użytkownik otrzymuje komunikat o zakończeniu sesji. Po wciśnięciu przycisku STRONA GŁÓWNA, powróci on do strony startowej. Korzystając ze strony info oraz przycisków dodanych do aplikacji, łatwo zauważyć zmieniające się liczniki.

4. Sesja w ASP.NET

Niniejszy punkt będzie traktować o mechanizmie sesji w technologii ASP.NET. Przedstawię sposób zarządzania mechanizmem sesji, jego startem, końcem, czasem trwania oraz pokaże jak używa się zmiennych sesji i aplikacji.

W momencie połączenia z aplikacją tworzone są obiekty Session i Application, które umożliwiają zarządzanie stanem sesji i aplikacji. W ASP.NET konfiguracja ta jest dostępna w pliku Global.asax.cs. Znajdują się tam metody m.in. startu oraz końca aplikacji i sesji, które są wywoływane w momencie zaistnienia odpowiadających im zdarzeń.

★

```
protected void Session_End(object sender, EventArgs e)
{
    //ciało metody
}
```

Czas sesji ustawiamy za pomocą metody Timeout() wraz z rozpoczęciem sesji w metodzie Session_Start() w Global.asax.cs

★

```
Session.Timeout = 1;
```

Lub też w pliku Web.config w sessionState timeout.

★

```
<configuration>
  <system.web>
    <sessionState mode="InProc" timeout="1"/>
  </system.web>
</configuration>
```

Poza tym w sessionState możemy ustawić m.in. tryb przechowywania informacji sesji. Domyślnie mode="InProc" przechowuje informacje w pamięci na serwerze.

Aby odwołać się do zmiennych sesji, należy skorzystać z właściwości Session obiektu Page. Do zmiennych sesyjnych można odwołać się poprzez indeks typu string lub integer. Zmienne sesyjne mogą być dowolnego typu, a więc programista może stworzyć własną klasę i przechowywać ją w sesji.

★

```
Session["WyswietleniaWSesji"] = 0;
Application["AktywneSesje"] = 0; //dla zmiennej aplikacji
```

Identyfikacja sesji polega na przypisywaniu sesji unikatowe SessionID. Wartości SessionID mogą być przechowywane w cookies lub adresie URL. Z tym, że wybór pierwszej opcji, wymusza na użytkowniku włączoną obsługę ciasteczek w przeglądarce.

Aby zakończyć sesję wcześniej niż poprzez zamknięcie przeglądarki, należy użyć metody Abandon(). Aby wyczyścić wartości zmiennych sesji należy użyć metody Clear().

★

```
Session.Abandon();
Session.Clear();
```

5. Sesja w PHP

W tym punkcie zostanie omówiony mechanizm sesji w PHP. Pokaże w jaki sposób zainicjować sesję, ustawić jej czas, zakończyć oraz ustawić zmienne.

PHP w mechanizmie sesji, nie korzysta z podobnego rozwiązania jak to w Global.asax.cs w ASP.NET. O start sesji musi zadbać programista. W momencie wywołania startu sesji, gdy identyfikator sesji nie istnieje, zostaje utworzony nowy i rozpoczyna się nowa sesja. Trwa do momentu jej zakończenia. W PHP w każdym pliku w którym chcemy używać mechanizmu sesji, należy ją zainicjować poniższą metodą:

★ `session_start();`

Czas sesji można ustawić w php.ini lub w pliku strony, przed inicjacją sesji, za pomocą metody `ini_set()`:

★ `ini_set('session.gc_maxlifetime',60);`

W momencie rozpoczęcia sesji, PHP tworzy superglobalną `$_SESSION`. Odwołując się do niej poprzez indeks, tworzy się zmienne sesji.

★ `$_SESSION['WszystkieSesje'] = $licznik;`

Do identyfikacji sesji wykorzystuje się niepowtarzalne identyfikatory. Przeglądarka za pomocą identyfikatorów wiąże sesję z grupą stron. Identyfikator wygasa wraz z końcem sesji. Zmienne sesji są zapisywane w cookie lub też przesyłane w adresie URL. Dane sesji są zapisane w osobnych plikach. Miejsce ich przechowywania można zmienić w php.ini. Poniżej metoda zwracająca ścieżkę:

★ `session_save_path();`

Aby zakończyć sesję należy wywołać poniższą metodę:

★ `session_destroy();`

Jednak zakończenie sesji, nie powoduje usunięcia danych sesji. Aby usunąć wszystkie zmienne sesji, należy do superglobalnej przypisać pustą tablicę.

★ `$_SESSION = array();`

6. Podsumowanie

Mechanizm sesji jest bardzo przydatny w tworzeniu portali informacyjnych czy też aplikacji webowych. Znacznie usprawnia proces tworzenia internetowych rozwiązań. Dzięki sesji, aplikacja potrafi zidentyfikować konkretnego użytkownika i wyświetlić interesujące go informacje. Niezależnie od użytej technologii, sposób działania sesji jest bardzo zbliżony. Warto jednak zwrócić uwagę, że pomimo użyteczności mechanizmu sesji, najczęstszymi rozwiązaniami jest łączenie mechanizmu sesji wraz np. z cookies, co daje większą funkcjonalność, gdyż ciasteczka są w stanie przechowywać informacje wiecznie, a sesje, tylko w obrębie swojego czasu działania.