# Strona Wzorcowa (Master Page)

Strony wzorcowe, będące funkcją programu ASP.NET 2.0, działają tak samo w programach Windows SharePoint Services 3.0 i Office SharePoint Server 2007, jak w programie ASP.NET 2.0. Przy użyciu stron wzorcowych można utworzyć jeden szablon strony, a następnie zamiast budować każdą nową stronę od początku, używać tego szablonu jako podstawy wielu stron w aplikacji.

Aby można było renderować strony wzorcowe w przeglądarce, tak naprawdę wymagają one dwóch osobnych części: samej strony wzorcowej i strony zawartości. Strona wzorcowa definiuje wspólny układ i nawigację, jak również wspólną zawartość domyślną dla wszystkich dołączonych do niej stron zawartości. Strona zawartości jest unikatowa. Gdy strona jest renderowana w przeglądarce, strona wzorcowa dostarcza wspólnej zawartości, a strona zawartości dostarcza zawartości specyficznej dla strony.

Najpierw tworzona jest jedna strona wzorcowa w celu zdefiniowania wyglądu i działania oraz standardowego zachowania wszystkich stron w witrynie lub określonej grupy tych stron. Następnie można utworzyć indywidualne strony zawartości zawierające unikatową zawartość, która ma być wyświetlana na każdej z nich. Strona wzorcowa jest scalana ze stronami zawartości w celu utworzenia ostatecznej, wyrenderowanej strony, która łączy w sobie układ ze strony wzorcowej oraz zawartość ze strony zawartości.

# Strony z zawartością

Zawartość strony są strony ASP.NET, które mają rozszerzenie nazwy pliku aspx regularne. Ponadto każdej strony zawartości zawiera dyrektywy @page, który identyfikuje strony wzorcowej, do której dołączono tej zawartości strony. Poniżej przedstawiono przykład dyrektywy @page.

< % @ MasterPageFile="~masterurl/default.master strony" % >

Strona zawartości może umieszczad kontrolki tylko w wyznaczonych miejscach. Istnieje możliwośd umieszczenia kontrolki w dowolnym miejscu na stronie przy pomocy pozycjonowania CSS, ale jest to w najwyższym stopniu niewskazane. Zmiana szablonu nie spowoduje zmiany położenia kontrolki. Jeśli musimy coś dodad w określonym miejscu, najłatwiej jest zdefiniowad w stronie wzorcowej kolejną kontrolkę ContentPlaceHolder. Aby ułatwid tworzenie rozbudowanych serwisów internetowych, możliwe jest zagnieżdżanie stron wzorcowych. Visual Studio 2008 wspiera ich wizualne projektowanie.

# Sterownie wyglądem aplikacji przy użyciu kompozycji

Przy użyciu arkusza styli (CSS) możliwe jest definiowanie wyglądu elementów w języku HTML, nie jest to jednak możliwe w przypadku kontrolek serwerowych ASP.NET. CSS może zmienid interpretacje kodu HTML generowanego przez kontrolkę serwerową, ale nie ma wpływu na samą kontrolkę. Aby umożliwid scentralizowany sposób sterowania wyglądem kontrolek, wprowadzono *kompozycje* (ang. *themes*). Kompozycje są umieszczone na serwerze w specjalnym katalogu *App\_Themes*, w podkatalogach, których nazwy są nazwami kompozycji. Podkatalogi zawierad mogą pliki .skin (plik motywu) oraz pliki arkuszy styli .css. Pliki styli zostaną dołączone do nagłówka każdej ze stron. Wizualne cechy kontrolek Web określa się za pośrednictwem pliku motywu (.skin).

Pojedynczy folder kompozycji może zawierad dowolną liczbę takich plików. Pliki motywów zawierają znacznik identyfikujący daną kontrolkę serwerową wraz z parametrami. Definicja motywów ma wpływ tylko na wygląd kontrolek serwerowych Web, nie ma natomiast wpływu na następujące elementy:

* elementy HTML
* kontrolki serwerowe HTML
* kontrolki serwerowe Web, które nie mają właściwości wizualnych
* niewizualne właściwości kontrolek serwerowych Web (np. AutoPostBack)

Stosowanie kompozycji do wszystkich stron witryny jest możliwe w pliku **Web.config** w sekcji <system.web> przy pomocy znacznika <pages>. Przykład:

<pages theme="NiebieskaSkorka" />

W celu utworzenia kompozycji należy w oknie **Solution Explorer** kliknąd prawym przyciskiem i wybrad **Add ASP.NET Folder**, a następnie **Theme**. W tym momencie możliwe jest określenie dowolnej nazwy folderu, która jednocześnie będzie nazwą kompozycji. W celu dodania pliku kompozycji należy kliknąd prawym przyciskiem katalog zawierający kompozycję i wybrad z menu opcje **Add New Item**, a następnie **Skin File** i wcisnąd przycisk **Add**.

# Nawigacja w serwisie internetowym

Większośd serwisów internetowych posiada menu – zbiór odnośników do poszczególnych stron. Możliwe jest określenie tego zbioru ręcznie na każdej stronie lub na stronie wzorcowej, jednak nie jest to rozwiązanie elastyczne. Dodatkowo problemem jest ukrywanie części odnośników przed użytkownikami anonimowymi oraz ograniczanie dostępu do odnośników dla różnych grup użytkowników.

W ASP.NET 2.0 zaproponowano spójny, łatwy do zarządzania, scentralizowany system dostarczania informacji o mapie serwisu. Możliwości systemu nawigacji:

* **Centralne zarządzanie logiczną strukturą witryny** – dodawanie lub usuwanie odnośników do stron domyślnie odbywa się w pliku **Web.sitemap**. Możliwe jest określenie innego dostawcy.
* **Kontrolki nawigacyjne** – kontrolki umożliwiające wyświetlenie menu serwisu bazujące na pliku **Web.sitemap**.
* **Reguły dostępu** – możliwośd określania reguł wyświetlania lub ukrywania odnośników w menu nawigacji.
* Możliwośd programowego dostępu do kontrolek nawigacji i wyświetlanych w nich informacji.

# Tworzenie mapy witryny

Pierwszym krokiem wykorzystania systemu nawigacji w ASP.NET jest określenie struktury serwisu. Standardowo system nawigacji serwisem wykorzystuje plik XML **Web.sitemap** zawierający hierarchię stron. Plik **Web.sitemap** musi byd umieszczony w katalogu głównym aplikacji. Zawiera on elementy <siteMapNode> dla każdej strony serwisu. Hierarchię serwisu ustala się poprzez osadzanie elementów <siteMapNode>. Każdy z nich zawiera następujące atrybuty:

* title – określa tekst wyświetlany w kontrolkach nawigacyjnych
* description - określenia tekst pomocy pojawiający się po ustawieniu kursora nad pozycją menu (ang. *tool tip*)
* url – określa lokalizację strony

# 6.Kontrolki nawigacyjne

Kolejnym krokiem po określeniu struktury jest wyświetlenie jej na stronie serwisu. ASP.NET udostępnia następujące kontrolki nawigacyjne:

* SiteMapPath
* TreeView
* Menu

## Kontrolka SiteMapPath

Kontrolka SiteMapPath wyświetla ścieżkę nawigacji. Prezentuje ona użytkownikowi lokalizację i wyświetla odnośniki umożliwiające nawigację do strony domowej. 

Tworzy ona panel nawigacyjny(inaczej nazywane ścieżka nawigacyjną). To liniowa ścieżka określająca aktualne położenie użytkownika w strukturze nawigacyjnej.

Zadaniem takiego typu nawigacji jest pokazanie użytkownikom aktualnego położenia względem innych elementów strony. Dawniej zaprogramowanie takiego panelu nawigacyjnego było dość trudne.

/\*

Jedną z ważniejszych właściwości stylu kontrolki SiteMapPath jest PathSeparator. Domyślnie kontrolka SiteMapPath korzysta ze znaku większośći i za jego pomocą oddziela kolejne hiperłącza\*\

## Kontrolka TreeView

Kontrolka TreeView wyświetla mapę serwisu w postaci struktury. Węzły zawierające elementy potomne mogę być rozwijane lub zwijane po kliknięciu 

Kontrolka serwerowa TreeView jest skomplikowaną kontrolka, która pozwala wyświetlać dane w postaci hierarchicznej i doskonale nadaje się do poazania elementów umieszczonych w pliku .stiemap

TreeView nie wykonuje automatycznego wiązania z mapą portalu, tak jak to robi kontrolka SiteMapPath

TreeView ma przygotowany liczy zestaw styli. Najlepszy sposób z korzystania z tych stylów polega na użyiu widoku design strony

## Kontrolka Menu

Kontrolka Menu wyświetla menu rozwijane. Węzły zawierające elementy potomne są rozwijane w momencie, kiedy przesunie się nad nimi kursor

Kontrolka jest idealna wtedy, gdy należy udostępnić użytkownikowi długa hierarchiczną listę opcji, wykorzystując tylko niewielką powierzchnię strony