# ASP .NET MVC vs ASP .NET WebForms

Gwoli wstepu, dla osób nie zaznajomionych z Web Developmentem przy pomocy narzedzi Microsoftu wspomne, ze stosowane sa 2 diametralnie odmienne podejscia do tej kwestii. Pierwsze z nich to ASP .NET Web Forms. Drugim natomiast, jest ASP .NET MVC.pojawia sie pytanie 'czemu rozwijac 2 rownolegle technologie zamiast skupic sie na jednej aby uczynic ja jak najlepsza?'. wszystko sprowadza sie do gustow odbiorcow, czyli developerow aplikacji internetowych.

# Rozliczenie WebForms:

Co moze przesdzkadzac w web formsach? Zacznijmy od tego, iz sa one proba przeniesienia modelu programistycznowego Windows Forms do srodowiska aplikacji internetowego. Bardzo wielu developerow (szczegolnie programujacych w VB) migrowalo na developmnet w ASP .NET bez podstawowej znajomosci HTTP i sieci.

slajd + dobrze znany wszystkim sposob tworzenia aplikacji

- czy aby na pewno odpowiedni dla internetu?

Doskonala okazja byl wiec dla nich model symulujacy windows forms, kierowany eventami oraz posiadajacy elementy takie jak viewstate czy postback. Niestety w nowoczesnej sieci wlasnie te elementy sprawiaja rowniez problem. Viewstate, przechowywany jako ukryte pole w markupie widoków, dla stron przechowujacych wiele danych bardzo przybiera na wadze.

slajd: - oslabiona nadmiarem przesylanych danych wydajnosc

Jak mozna sie domyslic- cierpi na tym wydajnosc aplikacji.

slajd: - nie nadazanie za standardami

brak kontroli nad generowanym HTMLem powoduje tworzenie tagów z inline'owymi cechoami, które nie spelniaja najnowszych, dynamicznie rozwijanych standardow. Integracja bibliotek javascriptowych rowniez jest utrudniona, przez narzucane w WebForms konwencje nazewnictwa elementow HTML.

slajd: - slaby rozdzial odpowiedzialnosci (separation of concerns)

- niska testowalnosc

Co jednak jest najslabszym punktem tego rozwiazania, to skomplikowany cykl zycia strony i bardzo ciasne powiazanie odpowiedzialnosci elementow aplikacji (poprzez model code behind). Utrudnia to rozwoj aplikacji oraz znacznie redukuje testowalnosc.

# Rozliczenie ASP .NET MVC:

Majac na uwadze te problemy, stworzono ASP .NET MVC, ktore ma swoje mocne strony tam, gdzie webforms zawodzi.

slajd: + luzne powiazanie odpowiedzialnosci (wzorzec MVC)

+ skalowalnosc, modularnosc

Zaimplementowano tutaj wzorzec MVC dla aplikacji internetowych, pozwalajacy na towrzenie aplikacji o luzno powiazanych funkcjonalnosciach, co owocuje wysoka modularnoscia i skalowalnoscia. Mamy tutaj podzial na Model implementujacy logike dla danych aplikacji i reprezentuje obiekty biznesowe udostepniajace dane Widokom do wyswietlenia w interfejsie uzytkownika. Widok jest niczym wiecej jak tylko szablonami HTML, prezentujacymi dane Modelu uzytkownikowi. Kontroler natomiast odpowiada za interakcje z uzytkownikiem.

slajd: + zastapienie eventow wywolaniami metod klas kontrolerow

Zastepuje on webformsowe eventy akcjami. Dzieki tym rozwiazaniom ASP .NET MVC nie musi porzystac z Viewstate, Postbackow oraz kontrolek serwerowych majacych pelna kontrole nad generowanym HTMLem.

slajd: + URLe przyjazne zarowno dla uzytkownika jak i dla przegladarek (REST, SEO)

Stosuje za to URLe oparte na REST, przyjazne pod katem optymalizacji wyszukiwarkowej.

[Odpalic strone tytulowa]

Najnowsza wersja ASP .NET MVC ma numer 4 i oferuje szereg nowoczesnych funkcjonalnosci. Glownym 'smakiem' tego wydania jest pelne wsparcie i czerpanie z najnowszych standardow (HTML5, CSS3) oraz tworzenie aplikacji na urzadzenia mobilne.

Jak widac w domyslnym, przykladowym projekcie MVC 4, juz na starcie dolaczone sa biblioteki wzbogacajace interfejs aplikacje takie jak jQuery, jQuery UI oraz Modernizr.

Jezeli nie jestescie zaznajomieni z jQuery lub jQuery UI, goraco zachecam do poczytania na ich temat- obydwie posiadaja bogate dokumentacje a linki beda dostepne pod koniec prezentacji.

# Modernizr

Modernizr natomiast jest swego rodzaju nowoscia, utworzona specjalnie dla HTML5. Pozwala na korzystanie z HTML5 i CSS3 podczas tworzenia strony, dbajac jednoczesnie o prawidlowe wyswietlanie zawartosci w przegladarkach z niepelna obsluga nowych standardów badz jej brakiem.

slajd: framework wspomagajacy utrzymanie jednolitego wygladu UI w starych przegladarkach

Jako przyklad ropzatrzyc mozna kwestie stylizacji elementów HTML5 w przestarzalych przegladarkach, gdzie czynnosc taka jest niemozliwa przy samym zastosowaniu CSS. Tu wlasnie wkracza Modernizr, ktory bez problemu zajmie sie tym dla nas.

Przy uruchomieniu tworzy on javascriptowy obiekt, zawierajacy definicje klas i elementów, na podstawie których rozpoznaje wsparcie dla nowych standardów przegladarki, w której zostal uruchomiony. Nastepnie na bazie tych informacji przydziela elementom HTML odpowiednie klasy CSS, umozliwiajace tworzenie regul dla starych oraz nowych przegladarek. Mozna okreslic go mianem biblioteki spelniajacej funkcje browser sniffera.

slajd: przyklad zastosowania Modernizra

.no-js .glossy,

.no-cssgradients .glossy {

background: url("images/glossy.png");

}

.cssgradients .glossy {

background-image: linear-gradient(top, #555, #333);

}

W tym przykladzie cssgradients to klasa wspierana przez modernizra i w przegladarace bez wsparcia dla gradientów tla elementy klasy glossy beda posiadaly jako tlo poday obrazek, natomiast jezeli wsparcie bedzie obecne- zastsowany zostanie gradient.

slajd tytul: wsparcie dla urzadzen mobilnych

Z biegiem czasu, w coraz wiekszym stopniu aplikacje internetowe sa uruchamiane za posrednictwem urzadzem mobilnych- smartphone'ow, tabletow etc. W znakomitej wiekszosci uzytkownik porusza sie po sieci nie za pomoca sztucznych interfejsow takich jak klawiatura i mysz, lecz z pomoca swoich wlasnych dloni poprzez interfejs typu touch. Z tymi dwoma trendami wiaze sie bardzo wazne pytanie- co zrobic aby przygotowac aplikacje internetowa do dostepu mobilnego?

slajd: nowe wymagania w kwestii interfejsu uzytkownika

Dokladnie takie pytanie zadali sobie ludzie pracujacy nad ASP .NET MVC4. W zwiazku z ograniczona przestrzenia ekranowa, interfejs uzytkownika musi byc bardziej kompaktowa przy zachowaniu czytelnosci. Majac na uwadze stosunek grubosci palca ludzkiego do grubosci kursora myszy, oczywistym staje sie, ze nie warto stosowac juz standardowych, niewielkich guzikow czy mikroskopijnych list rozwijanych. Swietnie, zatem wszystko jasne- tworzymy ogromne guziki i przepastne polacie pustego miejsca, a rozmiar czcionki nie schodzi ponizej 20. A co z uzytkownikami desktopowymi? Czy w takim razie trzeba napisac 2 interfejsy?

slajd: odpowiedz: adaptive rendering.

W tym scenariuszu przychodzi nam z pomoca zastosowany w ASP .NET MVC4 mechanizm adaptywnego renderingu za pomoca nowosci w CSS3- media queries. Co tak naprawde to oznacza?

[Pokazujemy desktop layout i phone layout]

Jak widac, strona wyglada dobrze zarówno na desktopie jak i na telefonie, linki sa duze, wszystko jest latwo dostepne.

# Zobaczmy jak to dziala w praktyce.

[resize desktopa]

Jak widac, kiedy okno przegladarki osiaga pewna szerokosc, wsyzstko nagle uklada sie w odpowiedni sposób aby zachowac jak najbardziej korzystny uklad strony. W czym tkwi sekret?

[CSS]

Nie potrzeba nam osobnych plików HTML ani niczego w ten desen- korzystamy jedynie z mechanizmu media query udostepnianego przez CSS3. Obecna tu regula okresla, ze w sytuacji gdy mamy do czynienia z pojedynczym ekranem o maksymalnej szerokosci 850px, obecne cechy elementów sa zastapione nowymi, powodujacymi wlasnie [taki] wyglad strony.

slajd: problemy?

Glówna wada media queries jest niestety fakt, iz bez wzgledu na docelowe urzadzenie, caly CSS zarówno dla urzadzenia mobilnego jak i desktopa musi zostac przeslany, co powoduje zbedne obciazenie lacza. Dlatego twórcy mvc 4 poszli o krok dalej, tworzac tryby wyswietlania (display modes).

slajd: rozwiazanie!

Mechanizm trybów wyswietlania pozwala na wysylanie róznych wersji danego widoku w zaleznosci od urzadzenia które wyslalo zadanie wyswietlenia strony.

[dodanie pliku .mobile do projektu i dostep przez ripple lub uzycie switchera]

Istnieje równiez alternatywna wersja zastosowania tego mechanizmu. (tworzenie z kodu)

Kolejnym elementem wpisujacym sie w plan wsparcia dla urzadzen mobilnych jest biblioteka jQuery Mobile, ktora mozna pobrac za pomoca narzedzia nuGet, zintegrowanego z VS 2011. Zadaniem tej biblioteki jest automatyczne nadawanie interfejsowi uzytkownika ladniejszego, bardziej 'mobilnego' wygladu.

[przyklad z menu]

jQ mobile uzywa atrybutow danych HTML5 do udostepnienia nam elmeentow automatycznie upiekszajacych nasza strone. Jak widac, korzystajac z wlasnych zasobow i danych, przy minimalnym wysilku z naszej strony jestesmy w stanie szybko przerobic UI naszej aplikacji na dobrze wygladajace w urzadzeniu mobilnym.

Zalety tego rozwiazania sa oczywiste - rozpoznanie urzadzenia wysylajacego zadanie i zwrocenie mu w odpowiedzi odpowiednich zasobow jest optymalne pod wzgledem transferu. Aplikacja stworzona przy pomocy jQ Mobile jest rowniez wysoce elastyczna, gdyz w zadnym miejscu UI nie wymaga kompromisow pomiedzy wersja desktop a mobile.

Zalety: max flex, wydajne pod wzgledem transferu.

Wady: powtarzalnosc, wykrywanie urzadzen nie jest trywialne