

Reporting Services

WinProg 2011/2012

Krzysztof Jeliński
Dawid Gawroński

1. SSRS – SQL Server Reporting Services

SQL Server Reporting Services udostępnia pełen zakres gotowych do użycia narzędzi i usług ułatwiających tworzenie, wdrażanie i zarządzanie raportami. Udostępnia również narzędzia programistyczne służące dostosowaniu raportów do naszych indywidualnych potrzeb.

Reporting Services (usługi raportujące) są platformą po stronie serwera, które umożliwiają wszechstronną funkcjonalność dla różnych źródeł danych. Usługi raportujące zawierają kompletny zasób narzędzi do tworzenia, zarządzania i dostarczania raportów oraz API, które pozwala deweloperom na wykorzystywanie danych w różnych aplikacjach. W skład usług raportujących wchodzi narzędzia do projektowania, konfiguracji, administracji i podglądu raportów. Narzędzia reporting services działają w środowisku Microsoft Visual Studio i są w pełni zintegrowane z narzędziami i komponentami SQL Server.

Używając usług raportujących można tworzyć interaktywne, tablicowe, graficzne lub dowolne raporty z relacyjnego, wielowymiarowego lub opartego na XML źródła danych. Reporting Services umożliwia publikowanie raportów, czasowe generowanie raportów lub generowanie raportów na żądanie. Reporting Services udostępnia wiele gotowych formatów widoku, pozwala na eksportowanie raportów do innych aplikacji i subskrypcje do już opublikowanych raportów. Utworzone raporty mogą być oglądane za pomocą przeglądarki internetowej lub jako część aplikacji Microsoft SharePoint. Tworzenie raportów i zestawień jest bardzo ważnym zagadnieniem w praktycznie każdej dziedzinie.

2. Korzyści płynące z korzystania z Reporting Services

- Dostęp do danych z poziomu przeglądarki www
- Mnóstwo specjalistycznych narzędzi ułatwiających pracę
- Pełen zasób gotowych rozwiązań
- Zdalny i łatwy dostęp do ważnych informacji
- Łatwiejsze wnioskowanie na podstawie dobrze zaprezentowanych danych
- Nawet najprostsze strony WWW korzystają z baz danych
- Prostota w analizowaniu danych

3. ASP.NET MVC + SQL Server Reporting Services

Twoja aplikacja ASP.NET MVC potrzebuje raportów. Co robisz? W tym tutorialu dowiesz się jak w prosty sposób wykorzystać SQL Server Reporting Services (SSRS) w swojej aplikacji ASP.NET MVC.

Prawie każda aplikacja ma do czynienia z danymi w takiej lub innej formie. Wydaje się być również prawdą stwierdzenie, że żadna aplikacja nie jest kompletna jeśli nie zawiera przynajmniej jednego raportu. W ASP.NET MVC, najprostszą opcją jest stworzenie własnego rozwiązania. Jest to oczywiście wykonalne, ale nie jest to rozwiązanie lepsze niż używanie narzędzi przeznaczonych do konkretnego zadania. W tym wypadku raportowania.

Innymi słowy, po co wywierać otwarte drzwi?

Zamiast wdawać się w drobne szczegóły na temat ASP.NET MVC i SSRS, w tym tutorialu skupimy się na włączaniu SSRS w ASP.NET MVC. Nawet jeśli nie jesteś zaznajomiony, z którąś z tych technologii, ten tutorial pomoże Ci pojąć podstawy. Jeśli jednak chcesz zdobyć szczegółową wiedzę na temat każdej z tych technologii, z pewnością znajdziesz wiele informacji na stronie Microsoftu.

Do wykonania tego tutorialu potrzebny będzie zainstalowany w systemie SQL Server 2008 z zainstalowanym Reporting Services. Dodatkowo potrzebne będzie również Visual Studio 2010 z zainstalowanym ASP.NET MVC. Zilustrowane tu przykładowe kody zostały stworzone w Visual Studio 2008 i ASP.NET MVC 2.0. Na zajęciach wykorzystywaliśmy MVC w wersji 3.0, jednak SRSS i MVC są niezależnymi technologiami, dlatego nie robi to żadnej różnicy. Przykładowy raport pokazuje tabele i kolumnę metadanych z Northwind Traders. Jeśli nie posiadasz bazy Northwind, żaden problem. Każda baza danych będzie dobra ponieważ zawiera taką samą strukturę metadanych zawartą w każdej bazie danych SQL Serwera. Zostało również założone, że implementujesz te przykłady na lokalnej maszynie, zalogowany jako użytkownik z prawami administratora. Dzięki temu unikniemy problemów związanych z bezpieczeństwem, na które byśmy natrafili pracując na kilku maszynach.

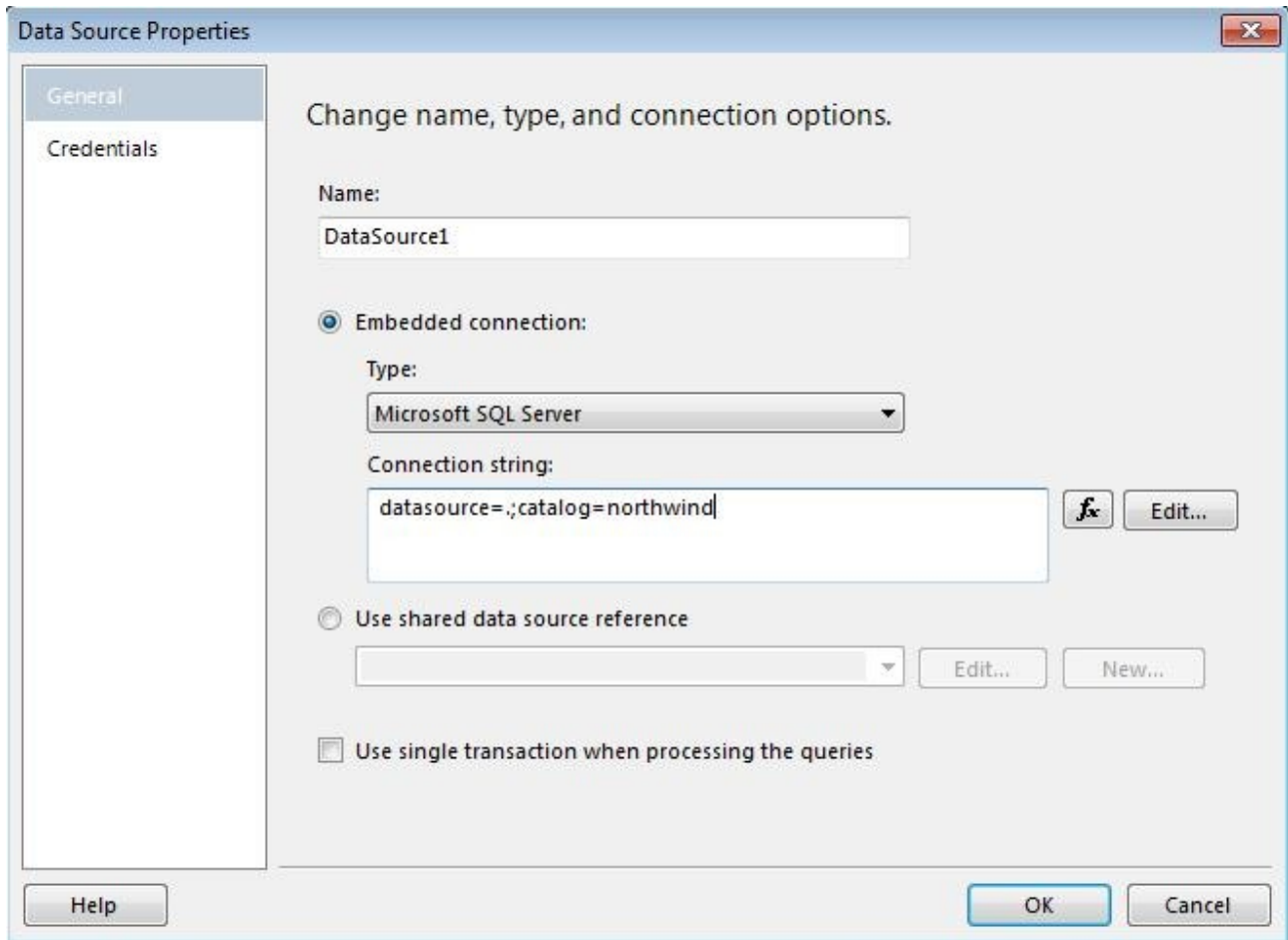
Plan naszego przykładu jest bardzo prosty. Najpierw każdy komponent będzie rozwijany osobno. Później natomiast połączymy ASP.NET MVC i SRSS w jedną funkcjonalną całość. Wytlumaczone zostanie również w jaki sposób przekazywać argumenty ze środowiska ASP.NET MVC do SRSS.

3.1. Komponent SRSS

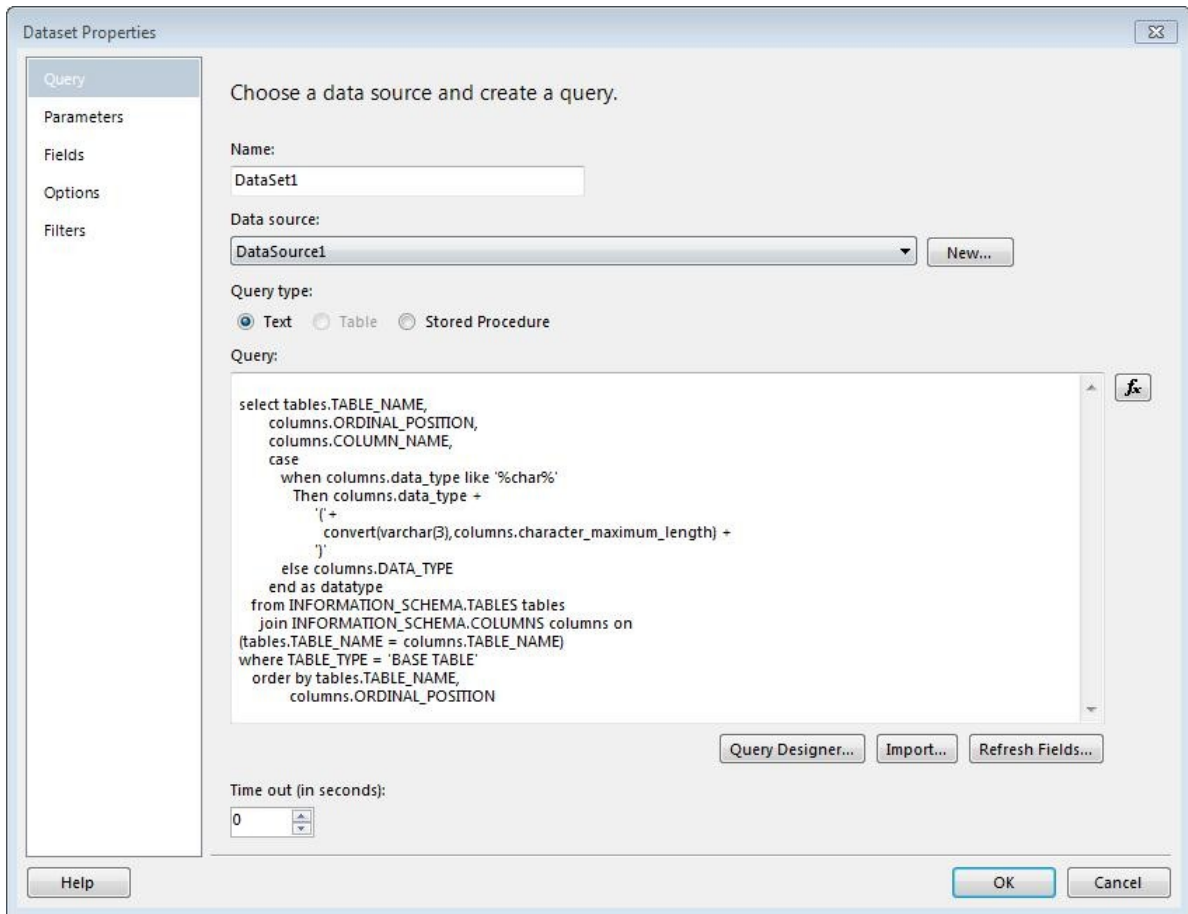
Przykładowy raport będzie używał tylko jednego zbioru danych opartego na zapytaniu SQL:

```
select tables.TABLE_NAME,
       columns.ORDINAL_POSITION,
       columns.COLUMN_NAME,
       case
         when columns.data_type like '%char%'
           Then columns.data_type +
                '(' +
                convert(varchar(3), columns.character_maximum_length) +
                ')'
         else columns.DATA_TYPE
       end as datatype
from INFORMATION_SCHEMA.TABLES tables
join INFORMATION_SCHEMA.COLUMNS columns
  on (tables.TABLE_NAME = columns.TABLE_NAME)
where TABLE_TYPE = 'BASE TABLE'
order by tables.TABLE_NAME,
       columns.ORDINAL_POSITION
```

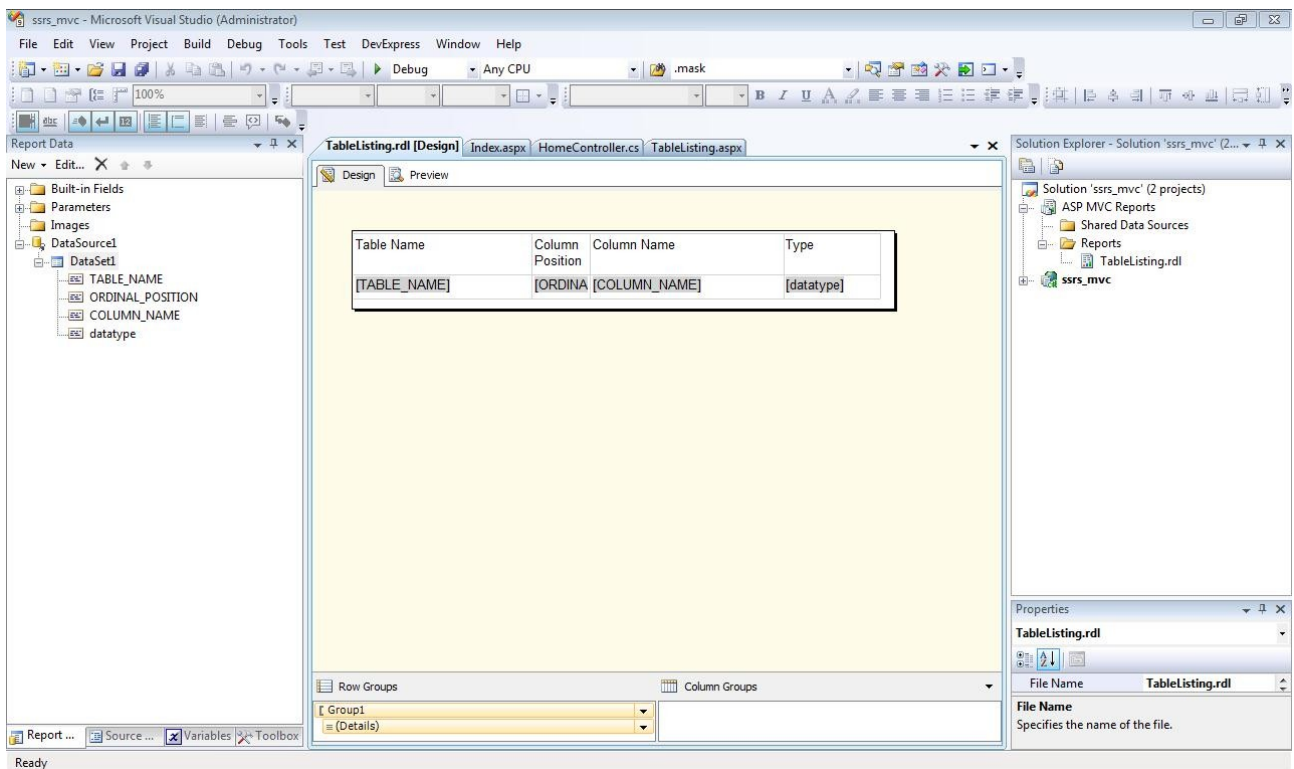
To proste zapytanie używa INFORMATION_SCHEMA do zwrócenia listy tabel, kolumn i powiązanych z bieżącą bazą typów danych. **Rysunek 1** i **rysunek 2** pokazują źródło danych i zbiór danych wymagane do wygenerowania raportu. **Rysunek 3** przedstawia sesję projektowania dla naszego przykładowego raportu, a **rysunek.4** pokazuje nasz raport w trybie podglądu. Tak jak już zostało wspomniane, jeśli nie posiadasz bazy danych Northwind Traders, możesz użyć innej dostępnej bazy danych. Zapytanie SQL będzie działało z każdą bazą danych. Upewnij się, że źródło danych jest poprawnie ustawione, żeby wybrana baza danych była tą właściwą.



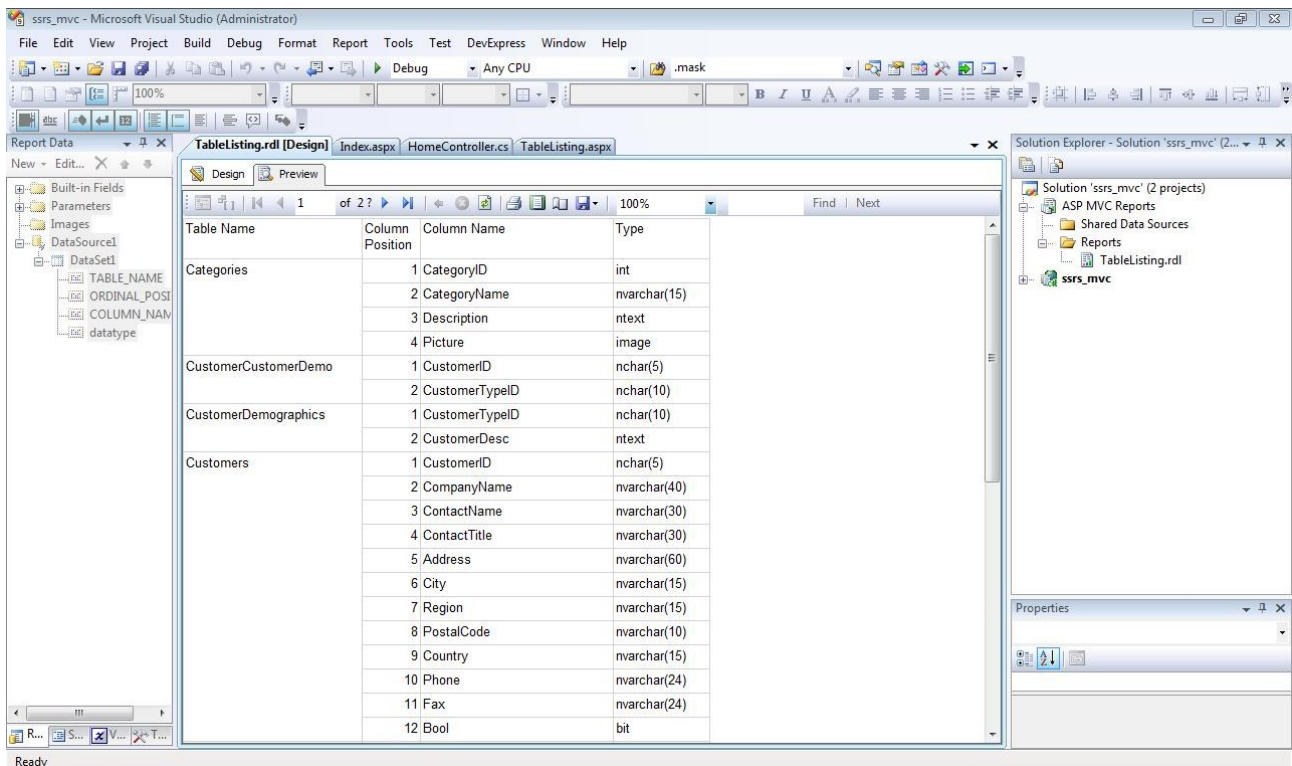
Rysunek 1: Źródło danych SSRS używane do połączenia z bazą danych



Rysunek 2: Zbiór danych SSRS zawierający dane wyświetlane przez raport SSRS



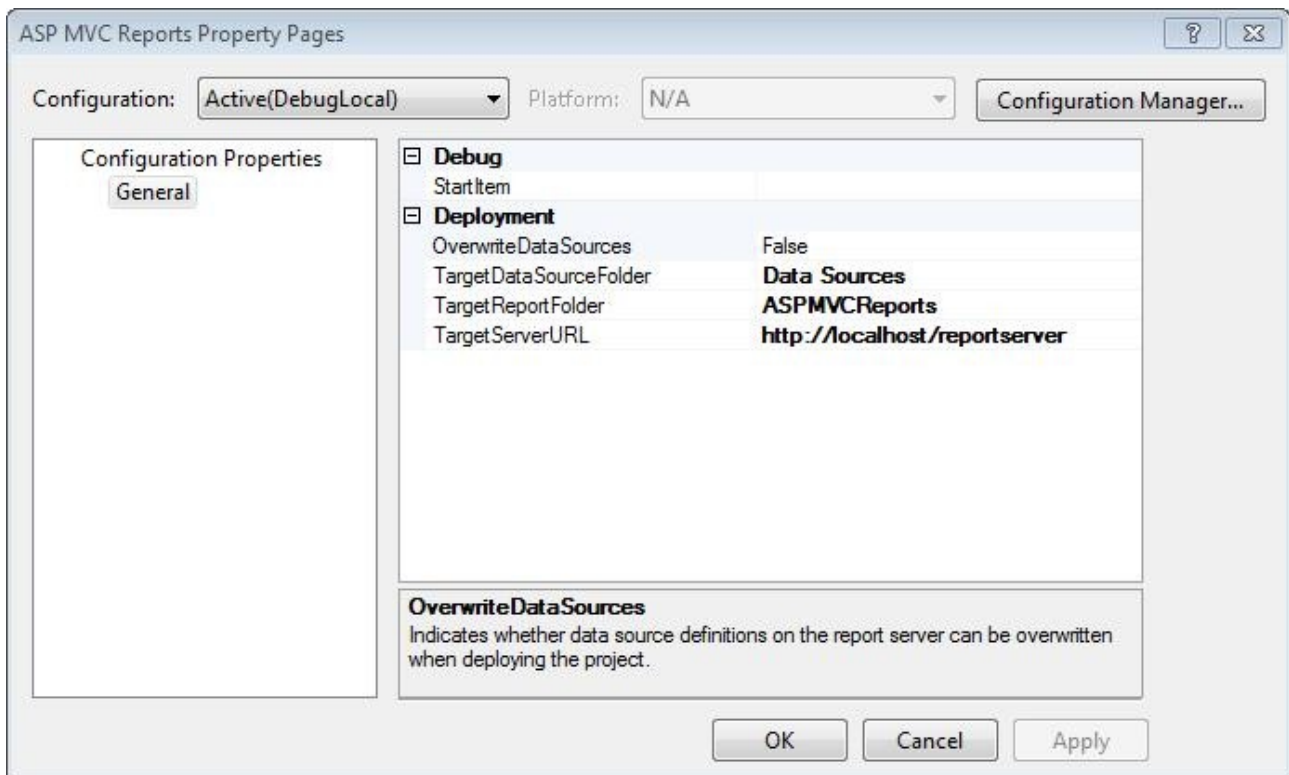
Rysunek 3: Tryb projektowania SSRS



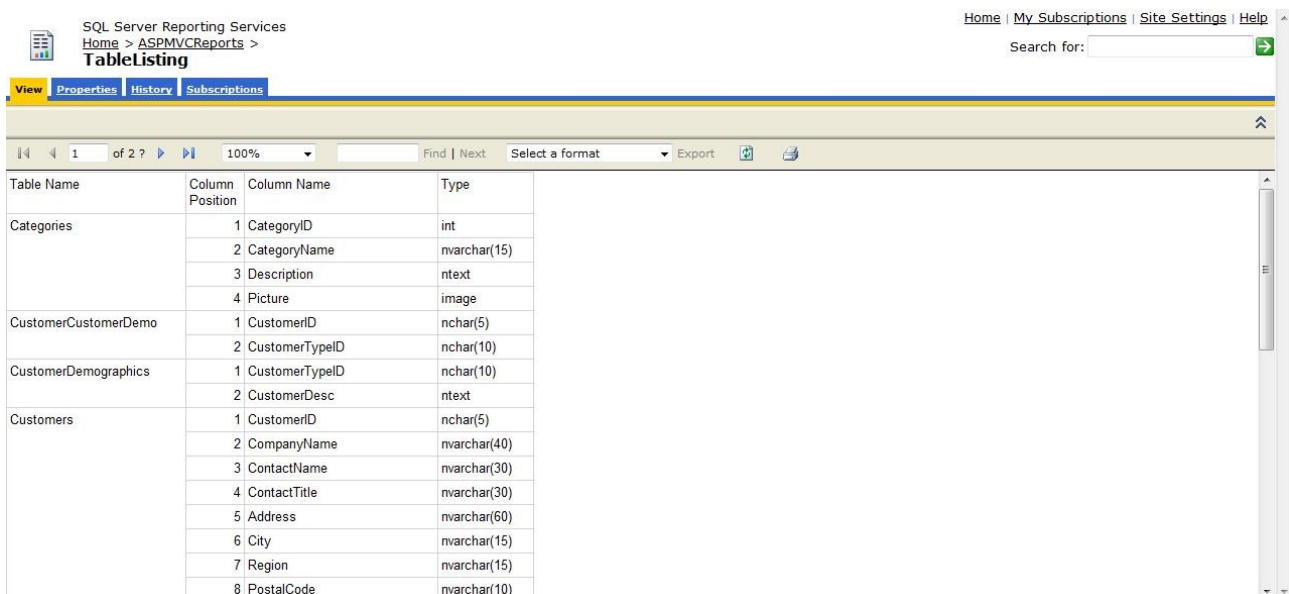
Rysunek 4: Raport SSRS w trybie podglądu

Ostatnią rzeczą jaka nam została to wdrożenie raportu. Jeśli masz zainstalowany lokalnie SQL Server, użyj „local report server”: <http://localhost/reportserver>. Zanim wdrożysz raport, musisz poinformować Reporting Services jaki serwer ma zostać użyty. Żeby to zrobić trzeba ustawić odpowiednio własność **TargetServerURL** w oknie właściwości projektu (**rysunek 5**). Gdy już określisz URL serwera, możesz zacząć wdrażanie raportu i źródeł raportu. By to zrobić po prostu kliknij prawym przyciskiem myszki na główny, węzeł projektu w Solution Explorerze i wybierz menu „Deploy”.

Rysunek 6 pokazuje jak nowo wdrożony raport prezentuje się w przeglądarce internetowej.



Rysunek 5: Właściwości projektu



Rysunek 6: Domyślny widok raportu w przeglądarce

To wszystko co dotyczyło części SSRS projektu. Raport jest wdrożony i jest gotowy do użycia. Kolejnym krokiem jest stworzenie podstawowego komponentu ASP.NET MVC.

3.2. Komponent ASP.NET MVC

Nasz początkowy komponent jest stworzony na bazie domyślnego szablonu projektu MVC. Jednak jest wymaganych kilka dodatkowych rzeczy:

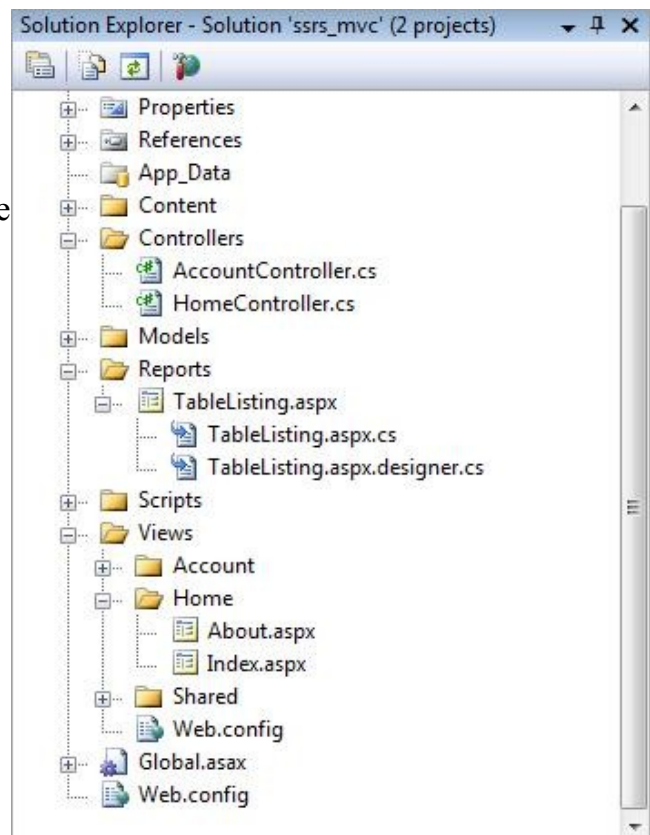
- Metoda kontrolera do uruchomienia raportu
- ASP.NET Web Form, aby można było dodać kontrolkę **ReportViewer**

Zgadza się! W celu realizacji naszego planu musimy włączyć do projektu ASP.NET Web Forms (z uwagi na kontrolkę ReportViewer). Dobra wiadomość jest taka, że MVC bazuje na Web Forms. Spójrz na jakikolwiek kod MVC a znajdziesz tam przestrzeń nazw System.Web. Czyli w pewnym sensie nie dodajesz niczego nowego do projektu MVC.

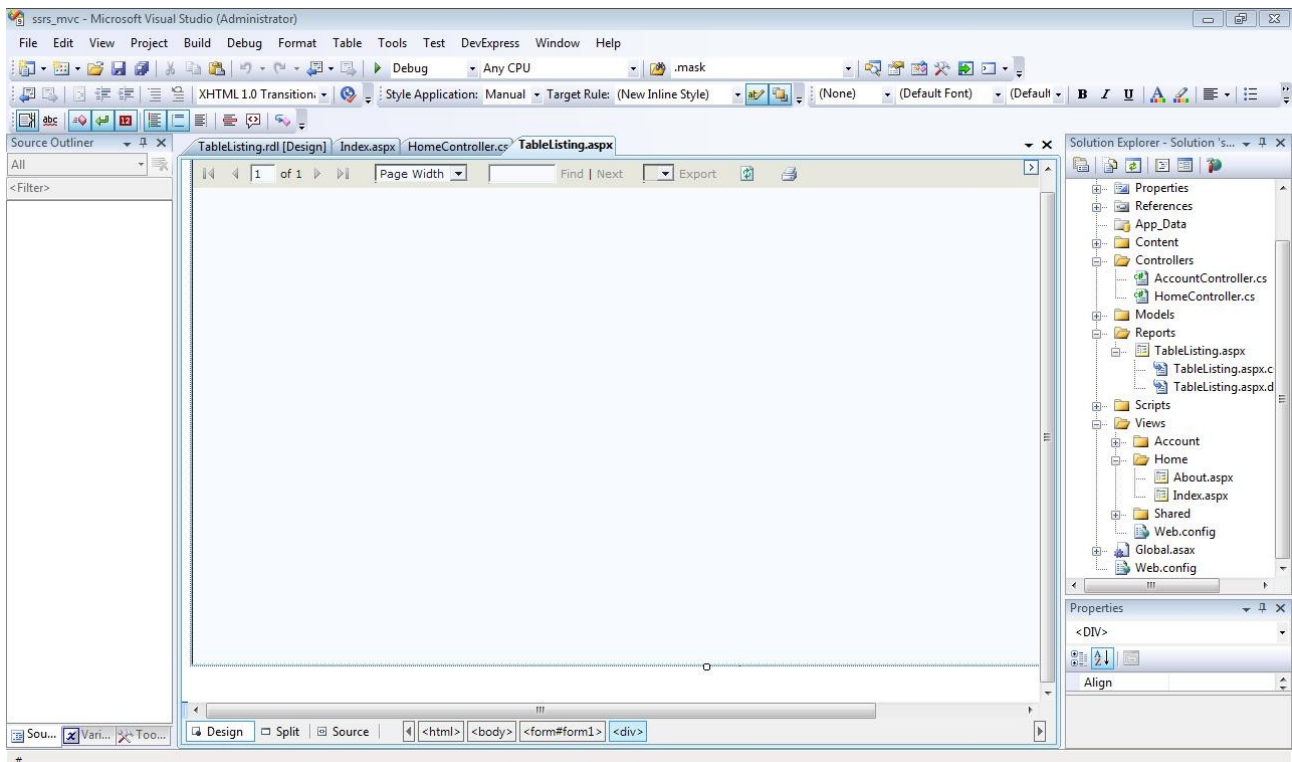
Technicznie rzecz biorąc metoda kontrolera używana do uruchamiania raportu, nie jest wymagana. Jeśli jednak chcesz przekazać informacje do kontekstu SSRS z kontekstu MVC, wtedy metoda kontrolera staje się konieczna. Jeśli nie ma wymogu przekazywania informacji, to można po prostu wywołać URL raportu. Jeśli chcesz tylko otworzyć raport, możesz po prostu użyć adresu URL w przeglądarce tak jak zostało pokazane na **rysunku 6**. Jeśli jednak potrzebujesz przekazać informację z jednego kontekstu do drugiego, wtedy potrzebujesz metody kontrolera aby przesłać dane i Web Formy aby odebrać dane.

Dodawanie Web Form do aplikacji ASP.NET MVC

Startując z domyślnego projektu MVC, **Rysunek 7** prezentuje jak wygląda nowo dodany Web Form do naszego folderu „Reports”. Jak na razie jest to pusta forma. Z toolboxu dodajemy na formę kontrolkę MicrosoftReportViewer. **Rysunek 8** pokazuje nowo dodaną kontrolkę do strony TableListing.aspx. Jak widać kontrolka ReportViewer wygląda jak w widoku przeglądarki (**rysunek 6**). Najlepsze jest to, że dostajemy pełną nawigację i funkcjonalność za darmo!



Rysunek 7: Strona Web Form TableListing.aspx obsługuje kontrolkę ReportViewer

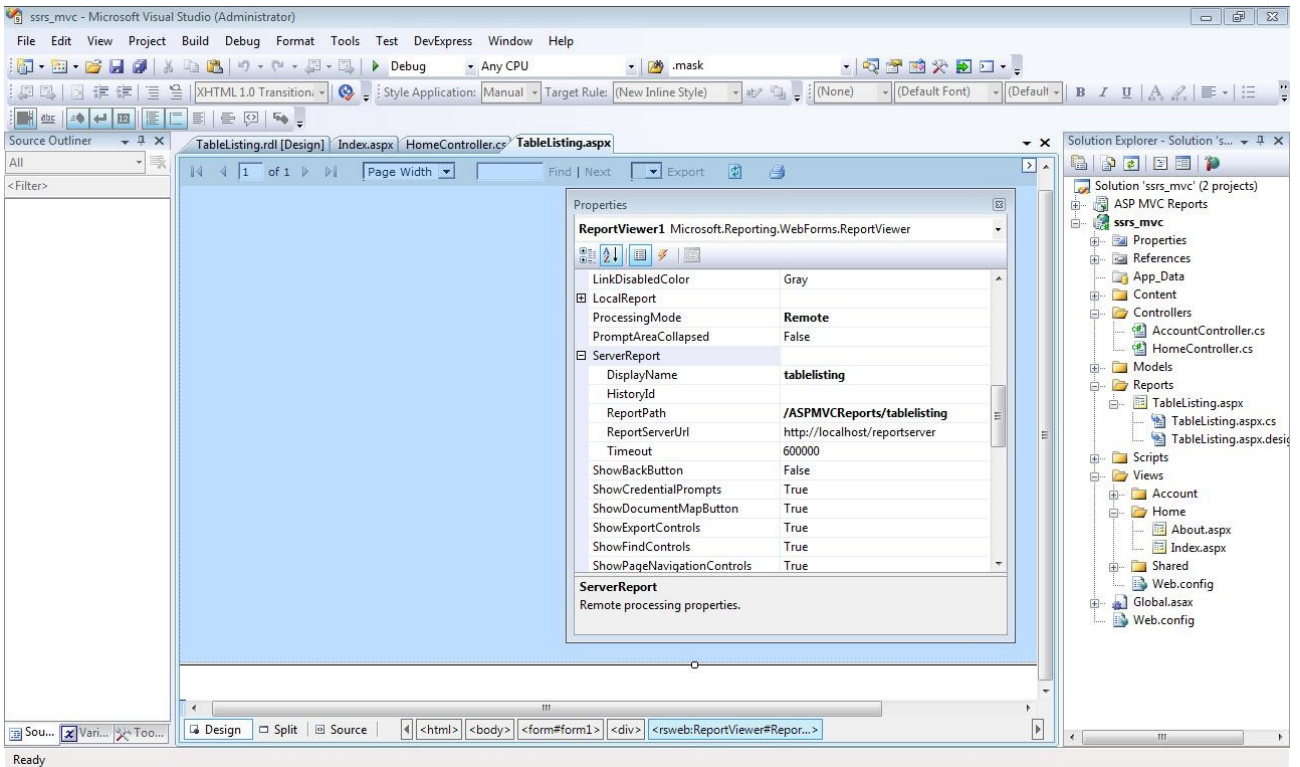


Rysunek 8: Kontrolka MicrosoftReportViewer umieszczona na stronie TableListing.aspx

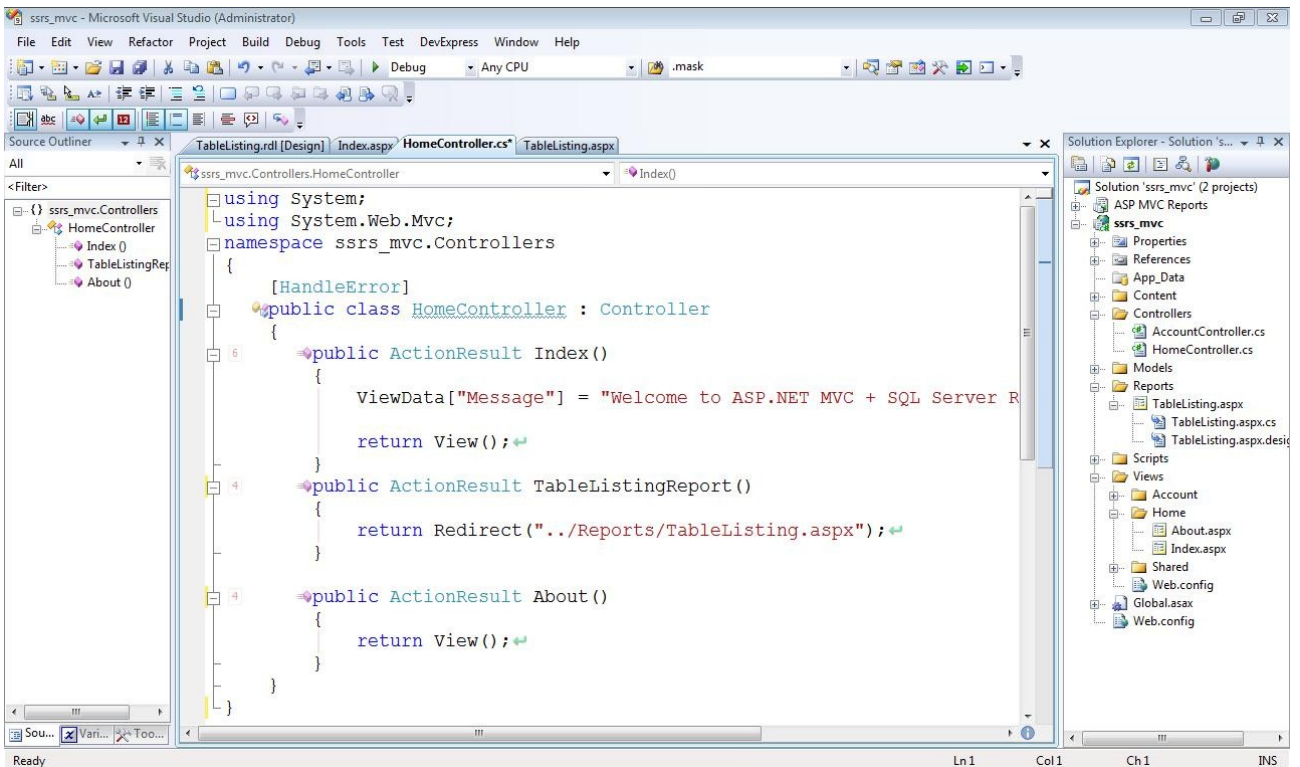
Gdy mamy już dodaną kontrolkę, pozostało nam do wykonania kilka kroków. Musimy ustawić w kontrolce lokalizację i nazwę raportu. **Rysunek 9** pokazuje potrzebne wpisy w oknie właściwości ReportViewer1. Właściwości obiektu ServerReport zostały ustawione na:

- DisplayName: tablelisting
- ReportPath: /ASPMVCReports/tablelisting
- ReportServerUrl: <http://localhost/reportserver>

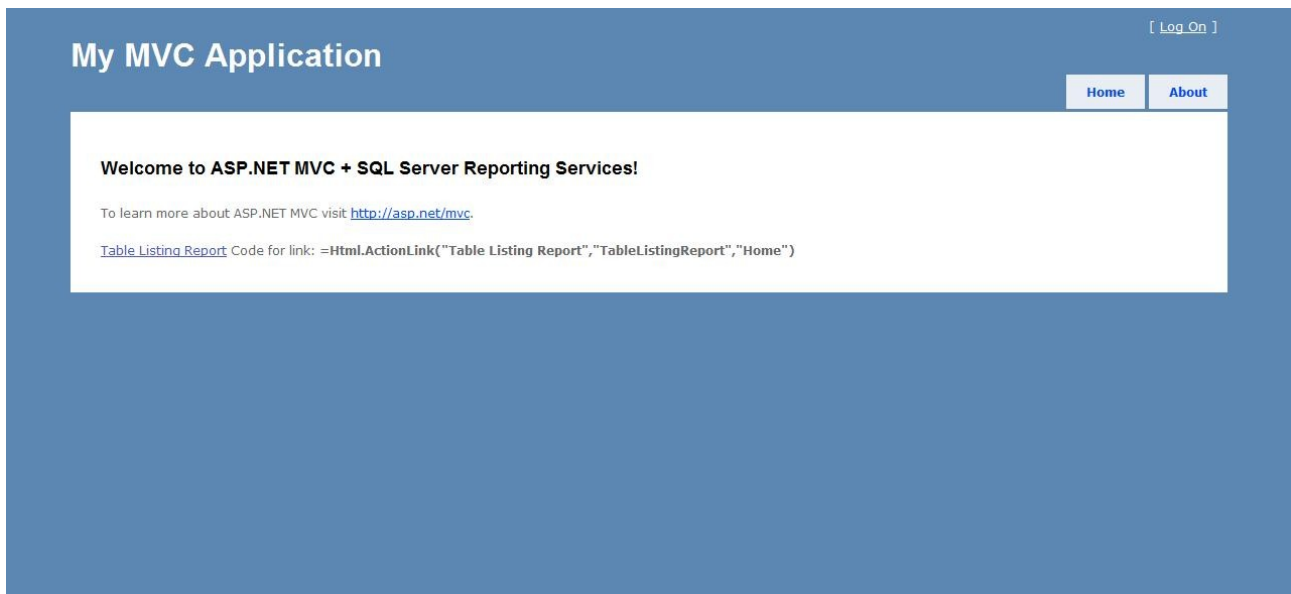
Następnie należy określić metody kontrolera dla kontrolera głównego. W tym celu zostanie wywołana metoda TableListingReport(). **Rysunek 10** pokazuje jak prosta jest ta nowa metoda. Na razie, metoda zawiera jedną linię kodu, który przekierowuje przeglądarkę na stronę podglądu raportów. Kliknięcie linku uruchomi raportu listingu tabel jak widać na **rysunku 12**.



Rysunek 9: Okno właściwości ReportViewer



Rysunek 10: Metoda kontrolera TableListingReport() jest używana do uruchomienia Web Formy TableListing.aspx



Rysunek 11: Widok główny z linkiem do ASP.NET Web Form która obsługuje kontrolkę MicrosoftReportViewer.

Table Name	Column Position	Column Name	Type
Categories	1	CategoryID	int
	2	CategoryName	nvarchar(15)
	3	Description	ntext
	4	Picture	image
CustomerCustomerDemo	1	CustomerID	nchar(5)
	2	CustomerTypeID	nchar(10)
CustomerDemographics	1	CustomerTypeID	nchar(10)
	2	CustomerDesc	ntext
Customers	1	CustomerID	nchar(5)
	2	CompanyName	nvarchar(40)
	3	ContactName	nvarchar(30)
	4	ContactTitle	nvarchar(30)
	5	Address	nvarchar(60)

Rysunek 12: Raport listingu tabel, obsłużony za pomocą Web Form a uruchomiony za pomocą widoku MVC.