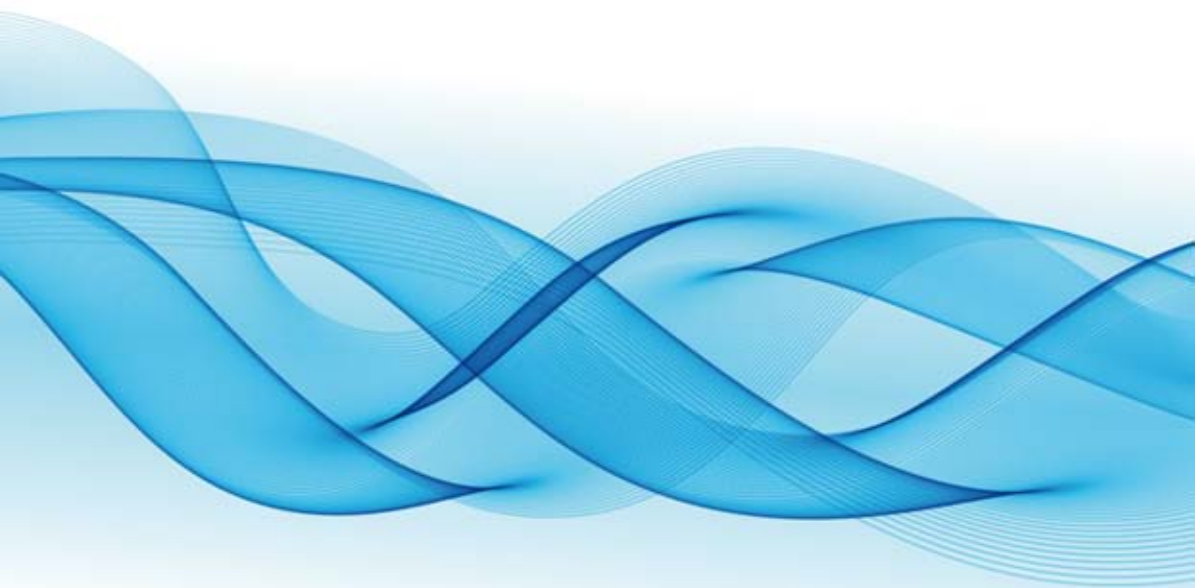




Dawid Borycki, Jacek Matulewski, Maciej Pakulski, Maciej Grabek

ASP.NET MVC

Kompletny przewodnik dla programistów
interaktywnych aplikacji internetowych
w Visual Studio



ASP.NET MVC

Skorzystaj z pomocy potężnego sprzymierzeńca!

Jak stworzyć wydajną aplikację działającą po stronie serwera?

Jak zaprząć bazy danych do ścisłej współpracy z aplikacją internetową?

Jak połączyć technologie ASP.NET MVC i ASP.NET WebForms dla uzyskania najlepszego efektu?

Jak zaimplementować serwis WebAPI oraz aplikację ASP.NET MVC 4 kompatybilną
z urządzeniami mobilnymi?

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Ewelina Burska

Projekt okładki: Studio Gravite/Olsztyn

Obarek, Pokoński, Pazdrijowski, Zapucki

Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/aspnvs>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-246-3238-1

Copyright © Helion 2014

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

- Przedmowa 9
- Wstęp 11
- Część I **ASP.NET MVC** 13
 - Rozdział 1. **Podstawowe aspekty projektu aplikacji ASP.NET MVC** 15
 - ASP.NET MVC 15
 - Projekt aplikacji MVC 18
 - Szablon projektu 19
 - Struktura projektu aplikacji ASP.NET MVC 21
 - Podsumowanie 23
 - Rozdział 2. **Kontroler i widok** 25
 - Funkcje i metody kontrolera 25
 - Projektowanie kontrolera 27
 - Widok 30
 - Projektowanie widoku 31
 - Transfer danych między widokiem a kontrolerem 36
 - Podsumowanie 37
 - Rozdział 3. **Model** 39
 - Wprowadzenie 39
 - Baza danych 40
 - Warstwa dostępu do bazy danych 42
 - Logika biznesowa 44
 - Implementacja widoku typu GridView 48
 - Podsumowanie 52
 - Rozdział 4. **Walidacja danych** 53
 - Wprowadzenie 53
 - Atrybuty modelu i walidacja danych w praktyce 55
 - Przygotowanie projektu aplikacji Kadry 55
 - Widoki kontrolera 58

Kontroler	63
Konfiguracja walidacji danych	66
Walidacja danych po stronie klienta	67
Walidacja danych a technologia ADO.NET Entity Framework	68
Podsumowanie	71
Rozdział 5. Wyrażenia regularne a kontrola poprawności danych	73
Podstawowe informacje o wyrażeniach regularnych	73
Odnajdywanie znaków specjalnych	74
Odnajdywanie znaków i cyfr	74
Odnajdywanie znaków w wyznaczonych miejscach łańcucha	75
Określanie liczby wystąpień podanych znaków	76
Tworzenie wzorców	76
Aplikacja	76
Kod pocztowy	80
Imię żeńskie	81
Walidacja adresu URL	83
Atrybuty modelu wykorzystujące wyrażenia regularne	84
Podsumowanie	88
Rozdział 6. Asynchroniczne aplikacje internetowe — Microsoft AJAX	91
Wprowadzenie	91
Aktualizacja zawartości wybranego elementu witryny	92
Częściowe odświeżanie witryny a fragment widoku	97
Ankieta (moduł do głosowania)	102
Odświeżenie widoku mapy serwisu Google	108
Rozdział 7. Technologia AJAX a biblioteka jQuery	113
Wprowadzenie	113
Aktualizacja zawartości wybranego elementu witryny	113
Automatyczne uzupełnienie pola tekstowego	115
Mechanizm IntelliSense a składnia jQuery	119
Podsumowanie	120
Rozdział 8. Trasowanie adresów URL	121
Wprowadzenie	121
Wzorce URL	122
Domyślne reguły trasowania i wzorce URL	123
Własne reguły trasowania	124
Zablokowanie obsługi wybranych żądań	124
Podsumowanie	126
Rozdział 9. Filtry	127
Wprowadzenie	127
Podstawowe atrybuty filtrów	128
[Authorize]	128
[HandleError]	129
[OutputCache]	133
[RequireHttps]	135
[NonAction]	135
Kolejność wykonywania filtrów	137

Implementacja własnych filtrów	139
Filtry akcji	139
Filtry autoryzacyjne, wyniku i wyjątków	144
Podsumowanie	146
Rozdział 10. Bezpieczne aplikacje internetowe	147
Potrzeba zabezpieczeń	147
Uwierzytelnienie i autoryzacja	149
Mechanizmy uwierzytelnienia zintegrowane z ASP.NET	149
Konfiguracja uwierzytelnienia	153
Autoryzacja	157
Grupy użytkowników	158
Podsumowanie	160
Rozdział 11. Konfiguracja zabezpieczeń i panel Web Site Administration Tool	161
Wprowadzenie	161
Użytkownicy, grupy i reguły dostępu	163
Ustawienia aplikacji oraz odczyt zawartości pliku konfiguracyjnego	167
Wyłączenie aplikacji i konfiguracja serwera SMTP	171
Komunikaty o błędach	171
Zarządzanie dostawcami usług	178
Podsumowanie	179
Rozdział 12. Zabezpieczenie witryny przed złośliwym kodem	181
Wprowadzenie	181
Działanie złośliwego kodu	181
Zabezpieczanie pliku konfiguracyjnego	187
Podsumowanie	192
Rozdział 13. Kontroler uwierzytelnienia w aplikacji ASP.NET MVC	195
Wprowadzenie	195
Model	195
Klasa kontrolera	198
Widoki kontrolera	203
Podsumowanie	209
Rozdział 14. Dynamiczne zarządzanie użytkownikami i grupami	211
Wprowadzenie.....	211
Kontroler panelu administracyjnego.....	212
Podział użytkowników na grupy	219
Podsumowanie.....	224
Rozdział 15. Biblioteka Knockout	227
Knockout, czyli interfejs użytkownika z wykorzystaniem wzorca MVVM	227
Knockout — wykorzystanie szablonów JavaScript	230
Część II Technologie bazodanowe w aplikacjach internetowych	235
Rozdział 16. Podstawy ADO.NET	237
Podstawy relacyjnych baz danych	237
Technologia ADO.NET	238

Instalacja bazy Northwind	240
Połączenie ze źródłem danych	241
Modyfikacja danych w obiekcie DataSet	249
Tworzenie nowych rekordów	250
Edycja rekordów	257
Wersjonowanie obiektu DataRow i kontrola wprowadzanych danych	263
Usuwanie rekordów	270
Rozdział 17. Obiekt TableAdapter jako pomost między DataSet a źródłem danych	273
Parametryzacja kwerend	274
Edycja danych przy użyciu obiektu TableAdapter	279
Integralność danych i ich usuwanie	288
Wstawianie danych z wykorzystaniem obiektu TableAdapter	294
Rozdział 18. Scaffolding, czyli ekspresowe tworzenie internetowych aplikacji bazodanowych	301
Lista rekordów	301
Usuwanie rekordów	305
Edycja rekordów	307
Tworzenie rekordów	311
Formularz z podformularzem (Master/Detail Form)	315
Rozdział 19. Transakcje	319
Tworzenie i ręczna kontrola transakcji	319
Automatyczne zarządzanie transakcjami	327
Konkurencyjność i poziom izolacji transakcji	332
Rozdział 20. Bezpieczeństwo w aplikacjach bazodanowych	335
Szyfrowanie połączenia	335
Zabezpieczenie kodu pośredniego przy pomocy Code Access Security (CAS)	338
ConnectionString a sposoby uwierzytelnienia w serwerze MS SQL Server	344
Uwagi końcowe dotyczące projektowania aplikacji bazodanowych	345
Rozdział 21. Entity Framework	347
Podstawy Entity Framework	348
Pierwsze zapytanie	348
Detale produktu	352
Dodawanie nowego produktu	353
Edycja produktu	355
Usuwanie produktu	357
Tworzenie schematu bazy danych na bazie modelu encji	357
Entity Framework i POCO	360
Część III ASP.NET MVC a ASP.NET WebForms	365
Rozdział 22. Integracja komponentów WebForms z aplikacją ASP.NET MVC na przykładzie technologii SQL Server Reporting Services	367
Pierwszy raport	368
Konfiguracja źródła danych	368
Projektowanie raportu	371

Kolumny obliczeniowe raportu	378
Stosowanie formatu prezentacji danych	379
Grupowanie danych	381
Projektowanie graficznego układu raportu	383
Etykiety, rysunki i listy	384
Prezentacja danych w postaci macierzowej	390
Parametry raportu i filtrowanie danych	394
Filtr definiowany na poziomie raportu	395
Filtrowanie raportu na etapie pobierania danych źródłowych	401
Kluczowe wskaźniki efektywności	403
Wykresy	406
Rozdział 23. Eksport danych	413
Rozdział 24. Księga gości w MVC. Porównanie ASP.NET Web Forms i ASP.NET MVC	423
Tworzenie projektu	423
Widok — formularz dodawania wpisu	425
Model — obsługa plików XML	427
Aplikacja — rejestrowanie zdarzeń i trasowanie	430
Kontroler — obsługa POST	431
Widok — wyświetlanie wpisów	432
Widok — walidacja	435
Kontroler — użycie modelu odsyłanego przez widok	440
Dodatki	447
Dodatek A ASP.NET MVC 4	449
Wsparcie dla urządzeń mobilnych	449
Sformułowanie problemu	450
Przystosowanie widoków aplikacji do urządzeń mobilnych	457
OAuth, czyli otwarta autoryzacja	460
Implementacja zdalnej autoryzacji w powiązaniu z portalem Facebook	461
Przetwarzanie asynchroniczne	464
Wprowadzenie	464
Potrzeba wykorzystania asynchronicznych kontrolerów	466
Kontrolery asynchroniczne i klasa Thread	469
Kontrolery asynchroniczne a zadania	472
Grupowanie i optymalizacja skryptów oraz kaskadowych arkuszy stylów	475
Web API	477
Wprowadzenie	477
Implementacja serwisu sieciowego	479
Implementacja klienta Web API	483
Podsumowanie	494
Dodatek B Publikacja aplikacji ASP.NET za pomocą internetowych usług informacyjnych (IIS)	495
Wprowadzenie	495
Instalacja i konfiguracja serwera internetowych usług informacyjnych IIS	496
Instalacja serwera IIS w systemie Windows 7	496
Instalacja serwera IIS w systemie Windows 8	496

Publikowanie aplikacji na serwerze IIS	497
Konfiguracja puli aplikacji	498
Instalacja ASP.NET 4/4.5 na serwerze IIS	498
Publikacja aplikacji	500
Uruchomienie aplikacji	503
IIS jako serwer do rozwoju aplikacji	504
Dodatek C ASP.NET Identity w MVC 5 i Visual Studio 2013	505
Wstęp.....	505
Szablony projektów ASP.NET w Visual Studio 2013	505
Kontroler zarządzający użytkownikami i rolami	508
Dynamiczne zarządzanie użytkownikami	509
Statyczne zarządzanie rolami	519
Dynamiczne zarządzanie rolami.....	519
Przykładowe wykorzystanie ról.....	524
Podsumowanie.....	526
Skorowidz	527

Rozdział 18.

Scaffolding, czyli ekspresowe tworzenie internetowych aplikacji bazodanowych

Dawid Borycki

W dwóch poprzednich rozdziałach samodzielnie utworzyliśmy komponenty prezentujące dane rekordów z bazy danych, a także umożliwiające ich edycję i usuwanie. Wykorzystaliśmy do tego celu bibliotekę jQuery UI. W tym rozdziale chciałbym zaprezentować tak zwany *scaffolding*. Jest to procedura dostępna między innymi w aplikacjach ASP.NET MVC, umożliwiająca automatyczne wygenerowanie widoków (komponentów) aplikacji internetowej, które realizują bazodanowy schemat CRUD (ang. *Create Read Update Delete*). Schemat CRUD obejmuje podstawowe elementy wszystkich aplikacji bazodanowych, czyli tworzenie, odczytywanie, aktualizowanie i usuwanie rekordów z tabel danych.

Lista rekordów

Opis techniki *scaffolding* w aplikacjach ASP.NET MVC rozpoczne od wygenerowania widoku listy rekordów w aplikacji *NorthwindMVC*. Automatycznie wygenerowany widok skoryguję w celu dostosowania go do potrzeb aplikacji:

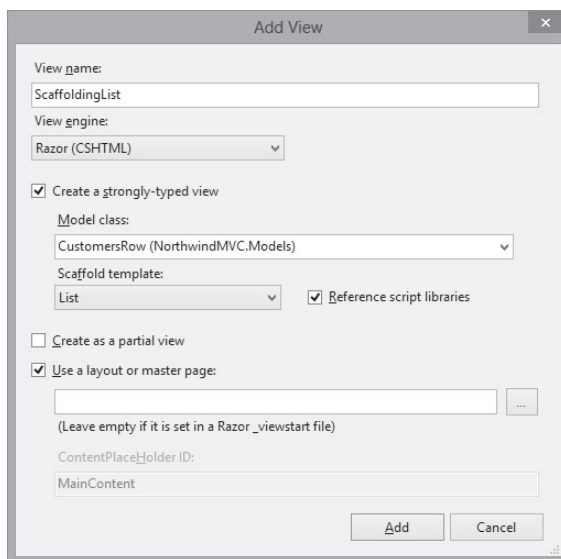
1. Główny widok aplikacji *NorthwindMVC* (plik *Index.cshtml*) uzupełnij o następujące polecenie:

```
<p>  
    @Html.ActionLink("Scaffolding", "ScaffoldingList")  
</p>
```

2. W *Solution Explorer* rozwiń węzeł *Views*, a następnie kliknij prawym przyciskiem myszy folder *Home*.
3. Wybierz opcję *Add*, a następnie *View...* z menu kontekstowego.
4. W kreatorze *Add View* (rysunek 18.1):

Rysunek 18.1.

Widok kreatora
Add View



- a) Zmień nazwę widoku na *ScaffoldingList*.
 - b) Zaznacz pole wyboru *Create a strongly-typed view*.
 - c) Z listy dostępnych modeli wybierz pozycję *CustomersRow*.
 - d) Z listy rozwijanej *Scaffold template* wybierz *List*.
 - e) Kliknij przycisk *Add*.
5. Utworzony w ten sposób plik *ScaffoldingList.cshtml* zmodyfikuj według wzoru z listingu 18.1.

Listing 18.1. Zmodyfikowany widok listy klientów

```
@model IEnumerable<NorthwindMVC.Models.NorthwindDataSet.CustomersRow>

@{
    ViewBag.Title = "ScaffoldingList";
}

<h2>ScaffoldingList</h2>
<h2>Lista klientów</h2>

<p>
    @Html.ActionLink("Create New", "Create")
    @Html.ActionLink("Nowy rekord", "ScaffoldingCreate")
</p>
```

```
<table>
  <tr>
    <th>
      CustomerID
    </th>
    <th>
      CompanyName
    </th>
    <th>
      ContactName
    </th>
    <th>
      ContactTitle
    </th>
    <th>
      Address
    </th>
    <th>
      City
    </th>
    <th>
      Region
    </th>
    <th>
      PostalCode
    </th>
    <th>
      Country
    </th>
    <th>
      Phone
    </th>
    <th>
      Fax
    </th>
    <th>
      RowError
    </th>
    <th>
      HasErrors
    </th>
    <th></th>
  </tr>

  @foreach (var item in Model) {
    <tr>
      <td>
        @Html.DisplayFor(modelItem => item.CustomerID)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.CompanyName)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.ContactName)
      </td>
      <td>
```

```

        @Html.DisplayFor(modelItem => item.ContactTitle)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Address)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.City)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Region)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.PostalCode)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Country)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Phone)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Fax)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.RowError)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.HasErrors)
    </td>
    <td>
        @Html.ActionLink("Edytuj", "ScaffoldingEdit", new { id =
            ↪ item.CustomerID }) |
        @Html.ActionLink("Zamówienia", "ScaffoldingMasterDetail", new { id =
            ↪ item.CustomerID }) |
        @Html.ActionLink("Usuń", "ScaffoldingDelete", new { id =
            ↪ item.CustomerID })
    </td>
</tr>
}
</table>

```

6. W pliku *HomeController.cs*:

a) Utwórz pole klasy:

```
CustomersTableAdapter customersTableAdapter = new CustomersTableAdapter();
```

b) Dodaj polecenia z listingu 18.2.

Listing 18.2. Prezentacja listy klientów

```

private void PobierzDaneKlientow()
{
    northwindDataSet.EnforceConstraints = false;
    customersTableAdapter.Fill(northwindDataSet.Customers);
}

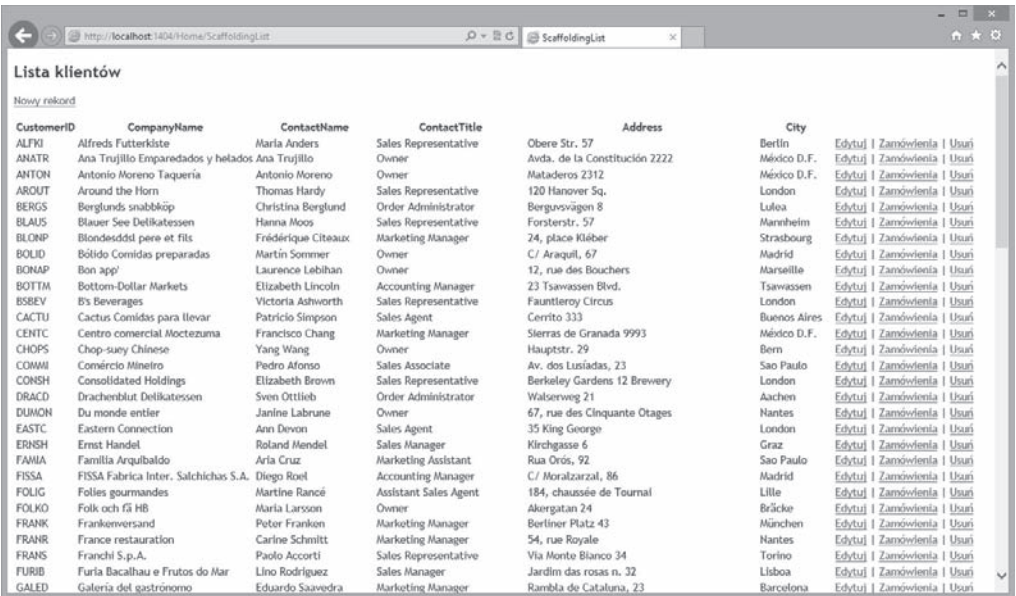
```

```
northwindDataSet.EnforceConstraints = true;
}

public ActionResult ScaffoldingList()
{
    PobierzDaneKlientow();
    return View("ScaffoldingList", northwindDataSet.Customers);
}
```

7. Skompiluj i uruchom aplikację.

Po uruchomieniu aplikacji i kliknięciu hiperłącza z etykietą *Scaffolding* zostanie wyświetlona lista klientów, której postać powinna być analogiczna do przedstawionej na rysunku 18.2. W kolejnym kroku zaimplementujemy możliwość usuwania, edycji oraz tworzenia rekordów w tabeli Customers.



CustomerID	CompanyName	ContactName	ContactTitle	Address	City
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.
AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London
BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Bergsgsvägen 8	Luleå
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim
BLOMP	Blondiesd ddt pere et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg
BOLID	Bólido Comidas preparadas	Martín Sommer	Owner	C/ Araquil, 67	Madrid
BONAP	Bon app'	Laurence Leblan	Owner	12, rue des Bouchers	Marseille
BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.	Tsawassen
BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus	London
CACTU	Cactus Comidas para llevar	Patricio Simpson	Sales Agent	Cerrito 333	Buenos Aires
CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Manager	Sierras de Granada 9993	México D.F.
CHOPS	Chop-suey Chinese	Yang Wang	Owner	Hauptstr. 29	Bern
COWMI	Comércio Mineiro	Pedro Afonso	Sales Associate	Av. dos Lusíadas, 23	Sao Paulo
CONSH	Consolidated Holdings	Elizabeth Brown	Sales Representative	Berkeley Gardens 12 Brewery	London
DRACD	Drachenblut Delikatessen	Sven Ottlieb	Order Administrator	Walserweg 21	Aachen
DUMON	Du monde entier	Janine Labrune	Owner	67, rue des Cinquante Otages	Nantes
EASTC	Eastern Connection	Ann Devon	Sales Agent	35 King George	London
ERNSH	Ernst Handel	Roland Mendel	Sales Manager	Kirchgasse 6	Graz
FAMIA	Família Arquibaldo	Arla Cruz	Marketing Assistant	Rua Orós, 92	Sao Paulo
FISSA	FISSA Fabrica Inter. Salchichas S.A.	Diego Roel	Accounting Manager	C/ Moralzarzal, 86	Madrid
FOLIG	Foiles gourmandes	Martine Rancé	Assistant Sales Agent	184, chaussée de Tournai	Lille
FOLKO	Folk och fä HB	Maria Larsson	Owner	Akergatan 24	Bräcke
FRANK	Frankenversand	Peter Franken	Marketing Manager	Berliner Platz 43	München
FRANR	France restauration	Carline Schmitt	Marketing Manager	54, rue Royale	Nantes
FRANS	Franchi S.p.A.	Paolo Accorti	Sales Representative	Via Monte Bianco 34	Torino
FURIB	Furia Bacalhau e Frutos do Mar	Lino Rodriguez	Sales Manager	Jardim das rosas n. 32	Lisboa
GALED	Galería del gastronómo	Eduardo Saavedra	Marketing Manager	Rambla de Catalunya, 23	Barcelona

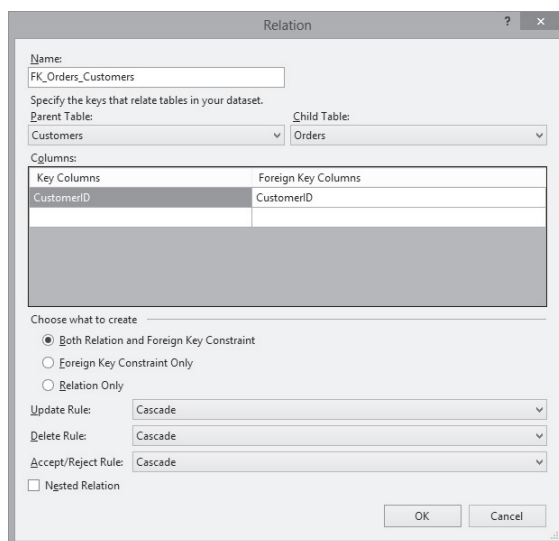
Rysunek 18.2. Lista klientów

Usuwanie rekordów

W celu zaimplementowania możliwości usuwania rekordów z tabeli Customers wykonaj następujące czynności:

- 1. W widoku projektowania obiektu NorthwindDataSet skonfiguruj relację pomiędzy tabelami Orders i Customers według wzoru z rysunku 18.3.
- 2. Klasę kontrolera Home uzupełnij o polecenia z listingu 18.3.

Rysunek 18.3.
Konfiguracja więzów
integralności danych



Listing 18.3. Usuwanie rekordów z tabeli Customers

```
private void AktualizujDaneKlientow()
{
    northwindDataSet.EnforceConstraints = false;

    orderDetailsTableAdapter.Update(northwindDataSet.Order_Details);
    ordersTableAdapter.Update(northwindDataSet.Orders);
    customersTableAdapter.Update(northwindDataSet.Customers);

    northwindDataSet.EnforceConstraints = true;
}

public ActionResult ScaffoldingDelete(string id)
{
    NorthwindDataSet.CustomersRow cdr =
        ↪northwindDataSet.Customers.FindByCustomerID(id);
    if (cdr != null)
    {
        // Usuwanie rekordów potomnych
        DataRow[] zamowienia =
            ↪cdr.GetChildRows(northwindDataSet.Relations["FK_Orders_Customers"]);

        foreach (DataRow dr in zamowienia)
            dr.Delete();

        // Usuwanie rekordu klienta
        cdr.Delete();

        // Aktualizacja wszystkich powiązanych tabel
        AktualizujDaneKlientow();
    }

    return ScaffoldingList();
}
```

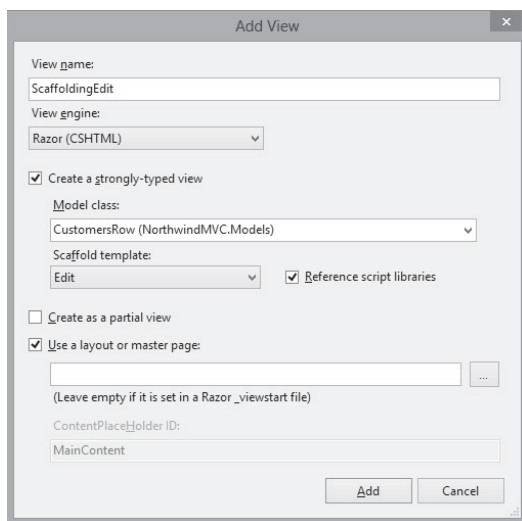
Podczas usuwania danego rekordu należy zadbać o rekordy z innych tabel, które są z nim związane relacjami. Wiersze tabeli `Customers` są skojarzone z rekordami z tabel `Orders` oraz `OrdersDetails`. Z tego powodu aktualizuję zawartości tych tabel po usunięciu wybranego rekordu klienta. Osierocone rekordy z tabeli `Orders` są usuwane razem z nim.

Edycja rekordów

Do edycji rekordów w tabeli `Customers` potrzebny będzie odpowiedni widok, który przygotuję na bazie automatycznie wygenerowanego kodu źródłowego:

1. Zgodnie z przepisem z podrozdziału „Lista rekordów” utwórz widok kontrolera `Home` i skonfiguruj go według wzoru z rysunku 18.4.

Rysunek 18.4.
Konfiguracja widoku edycji klienta



2. Wygenerowany automatycznie plik widoku `ScaffoldingEdit.cshtml` zmodyfikuj zgodnie z listingiem 18.4.

Listing 18.4. *Definicja widoku edycji klienta*

```
@model NorthwindMVC.Models.NorthwindDataSet.CustomersRow

@{
    ViewBag.Title = "ScaffoldingEdit";
}

<h2>ScaffoldingEdit</h2>
<h2>Edycja rekordu</h2>

<script src="@Url.Content("~/Scripts/jquery.validate.min.js")"
    type="text/javascript"></script>
```

```
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
type="text/javascript"></script>
```

```
@using (Html.BeginForm("EdytujDaneKlienta", "Home", new {id = Model.CustomerID})) {
    @Html.ValidationSummary(true)
    <fieldset>
        <legend>Dane klienta</legend>

<div class="editor-label">
    @Html.LabelFor(model => model.CustomerID)
</div>
<div class="editor-field">
    @Html.EditorFor(model => model.CustomerID)
    @Html.ValidationMessageFor(model => model.CustomerID)
</div>

        <div class="editor-label">
            @Html.LabelFor(model => model.CompanyName)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.CompanyName)
            @Html.ValidationMessageFor(model => model.CompanyName)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.ContactName)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.ContactName)
            @Html.ValidationMessageFor(model => model.ContactName)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.ContactTitle)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.ContactTitle)
            @Html.ValidationMessageFor(model => model.ContactTitle)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.Address)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.Address)
            @Html.ValidationMessageFor(model => model.Address)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.City)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.City)
            @Html.ValidationMessageFor(model => model.City)
        </div>

<div class="editor-label">
```

```

        @Html.LabelFor(model => model.Region)
    </div>
    <div class="editor-field">
        @Html.EditorFor(model => model.Region)
        @Html.ValidationMessageFor(model => model.Region)
    </div>

    <div class="editor-label">
        @Html.LabelFor(model => model.PostalCode)
    </div>
    <div class="editor-field">
        @Html.EditorFor(model => model.PostalCode)
        @Html.ValidationMessageFor(model => model.PostalCode)
    </div>

    <div class="editor-label">
        @Html.LabelFor(model => model.Country)
    </div>
    <div class="editor-field">
        @Html.EditorFor(model => model.Country)
        @Html.ValidationMessageFor(model => model.Country)
    </div>

    <div class="editor-label">
        @Html.LabelFor(model => model.Phone)
    </div>
    <div class="editor-field">
        @Html.EditorFor(model => model.Phone)
        @Html.ValidationMessageFor(model => model.Phone)
    </div>

    <div class="editor-label">
        @Html.LabelFor(model => model.Fax)
    </div>
    <div class="editor-field">
        @Html.EditorFor(model => model.Fax)
        @Html.ValidationMessageFor(model => model.Fax)
    </div>

    <div class="editor-label">
        @Html.LabelFor(model => model.RowError)
    </div>
    <div class="editor-field">
        @Html.EditorFor(model => model.RowError)
        @Html.ValidationMessageFor(model => model.RowError)
    </div>

    <p>
        <input type="submit" value="SaveZapisz" />
    </p>
</fieldset>
}

<div>
    @Html.ActionLink("Back to List", "Index")
    @Html.ActionLink("Powrót do listy rekordów", "ScaffoldingList")
</div>

```

3. W kontrolerze Home zdefiniuj metody z listingu 18.5.

Listing 18.5. *Edycja danych klienta*

```
public ActionResult ScaffoldingEdit(string id)
{
    NorthwindDataSet.CustomersRow cdr =
        ↪northwindDataSet.Customers.FindByCustomerID(id);

    if (cdr != null)
        return View("ScaffoldingEdit", cdr);
    else
        return ScaffoldingList();
}

private void PrzepiszDaneKlienta(FormCollection daneKlienta, ref
    ↪NorthwindDataSet.CustomersRow rekordKlienta)
{
    rekordKlienta.CompanyName = daneKlienta["CompanyName"].ToString();
    rekordKlienta.ContactName = daneKlienta["ContactName"].ToString();
    rekordKlienta.ContactTitle = daneKlienta["ContactTitle"].ToString();
    rekordKlienta.Address = daneKlienta["Address"].ToString();
    rekordKlienta.City = daneKlienta["City"].ToString();
}

public ActionResult EdytujDaneKlienta(FormCollection daneKlienta, string id)
{
    NorthwindDataSet.CustomersRow cdr =
        ↪northwindDataSet.Customers.FindByCustomerID(id);
    if (cdr != null)
    {
        PrzepiszDaneKlienta(daneKlienta, ref cdr);
        AktualizujDaneKlientow();
    }
    return ScaffoldingList();
}
```

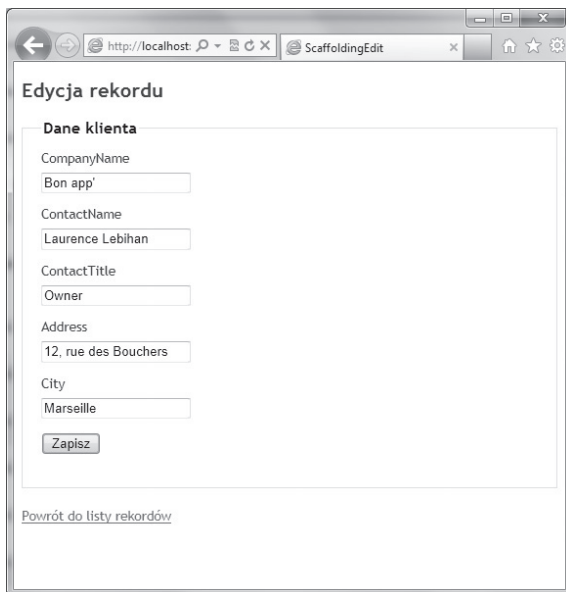
Proces edycji rekordu w bazie danych za pomocą technologii ADO.NET opiera się na informacjach zebranych w poprzednich rozdziałach, więc nie będę go tu szczegółowo opisywał. Celem tego przykładu było automatyczne wygenerowanie widoku umożliwiającego edycję danych klienta, przystosowanie go do własnych potrzeb oraz integracja z kontrolerem.

Separacja modelu, widoku i kontrolera umożliwia narzędziom Visual Studio 2012 automatyczne generowanie widoku na podstawie definicji modelu. Jednakże, jak pokazałem w tym podrozdziale, kreator tworzy widok, który umożliwia edycję wszystkich pól rekordu (nawet identyfikatora), co sprawia, że w większości przypadków należy samodzielnie edytować plik widoku w celu przystosowania go do potrzeb projektowanej aplikacji. Jednak, jak pokazuje powyższy przykład, nie jest to rzeczą skomplikowaną.

Integracja widoku z kontrolerem polega na odpowiednim przekazywaniu informacji o edytowanym rekordzie oraz wartości jego pól. Do tego celu wykorzystałem identyfikator klienta, który jednoznacznie identyfikuje rekordy w tabeli Customers.

Przykładowy wynik poleceń zaimplementowanych w tym podrozdziale przedstawia rysunek 18.5.

Rysunek 18.5.
Edycja danych klienta



The screenshot shows a web browser window with the address bar displaying 'http://localhost:...' and the page title 'ScaffoldingEdit'. The main content area is titled 'Edycja rekordu' (Edit record). Below this title is a section labeled 'Dane klienta' (Client data) containing several text input fields: 'CompanyName' with the value 'Bon app'', 'ContactName' with 'Laurence Lebihan', 'ContactTitle' with 'Owner', 'Address' with '12, rue des Bouchers', and 'City' with 'Marseille'. A 'Zapisz' (Save) button is located below the 'City' field. At the bottom of the form, there is a link labeled 'Powrót do listy rekordów' (Return to record list).

Tworzenie rekordów

W poprzednich podrozdziałach zaimplementowałem trzy spośród czterech podstawowych elementów schematu CRUD, czyli odczytywanie (R), aktualizację (U) oraz usuwanie (D) rekordów. Ostatnim elementem jest tworzenie (C) nowych rekordów. W celu zaimplementowania procedury uzupełniania tabeli *Customers* nowymi rekordami należy postąpić następująco:

1. W *Solution Explorer* rozwiń węzeł *Views*, a następnie kliknij prawym przyciskiem myszy folder *Home*.
2. Wybierz opcję *Add*, a następnie *View...* z menu kontekstowego.
3. W kreatorze *Add View*:
 - a) Zmień nazwę widoku na *ScaffoldingCreate*.
 - b) Zaznacz pole wyboru *Create a strongly-typed view*.
 - c) Z listy dostępnych modeli wybierz pozycję *CustomersRow*.
 - d) Z listy rozwijanej *Scaffold template* wybierz *Create*.
 - e) Kliknij przycisk *Add*.
4. Utworzony w ten sposób plik *ScaffoldingCreate.cshtml* zmodyfikuj według wzoru z listingu 18.6.

Listing 18.6. Widok umożliwiający wstawianie rekordów do tabeli *Customers*

```

@model NorthwindMVC.Models.NorthwindDataSet.CustomersRow

@{
    ViewBag.Title = "ScaffoldingCreate";
}

<h2>ScaffoldingCreate</h2>
<h2>Tworzenie rekordu</h2>

<script src="@Url.Content("~/Scripts/jquery.validate.min.js")"
    type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
    type="text/javascript"></script>

@using (Html.BeginForm("DodajKlienta", "Home"))
{
    @Html.ValidationSummary(true);
    <fieldset>
        <legend>Dane klienta</legend>

        <div class="editor-label">
            @Html.LabelFor(model => model.CustomerID)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.CustomerID)
            @Html.ValidationMessageFor(model => model.CustomerID)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.CompanyName)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.CompanyName)
            @Html.ValidationMessageFor(model => model.CompanyName)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.ContactName)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.ContactName)
            @Html.ValidationMessageFor(model => model.ContactName)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.ContactTitle)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.ContactTitle)
            @Html.ValidationMessageFor(model => model.ContactTitle)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.Address)
        </div>
    </fieldset>
}

```

```
<div class="editor-field">
    @Html.EditorFor(model => model.Address)
    @Html.ValidationMessageFor(model => model.Address)
</div>

<div class="editor-label">
    @Html.LabelFor(model => model.City)
</div>
<div class="editor-field">
    @Html.EditorFor(model => model.City)
    @Html.ValidationMessageFor(model => model.City)
</div>

<del><div class="editor-label">
    @Html.LabelFor(model => model.Region)
</del>
</del>
<del><div class="editor-field">
    @Html.EditorFor(model => model.Region)
    @Html.ValidationMessageFor(model => model.Region)
</del>
</del>

<del><div class="editor-label">
    @Html.LabelFor(model => model.PostalCode)
</del>
</del>
<del><div class="editor-field">
    @Html.EditorFor(model => model.PostalCode)
    @Html.ValidationMessageFor(model => model.PostalCode)
</del>
</del>

<del><div class="editor-label">
    @Html.LabelFor(model => model.Country)
</del>
</del>
<del><div class="editor-field">
    @Html.EditorFor(model => model.Country)
    @Html.ValidationMessageFor(model => model.Country)
</del>
</del>

<del><div class="editor-label">
    @Html.LabelFor(model => model.Phone)
</del>
</del>
<del><div class="editor-field">
    @Html.EditorFor(model => model.Phone)
    @Html.ValidationMessageFor(model => model.Phone)
</del>
</del>

<del><div class="editor-label">
    @Html.LabelFor(model => model.Fax)
</del>
</del>
<del><div class="editor-field">
    @Html.EditorFor(model => model.Fax)
    @Html.ValidationMessageFor(model => model.Fax)
</del>
</del>

<del><div class="editor-label">
    @Html.LabelFor(model => model.RowError)
</del>
</del>
<del><div class="editor-field">
```

```

        @Html.EditorFor(model => model.RowError)
        @Html.ValidationMessageFor(model => model.RowError)
    </div>

    <p>
        <input type="submit" value="CreateUtwórz" />
    </p>
</fieldset>
}

<div>
    @Html.ActionLink("Back to List", "Index")
    @Html.ActionLink("Powrót do listy", "ScaffoldingList")
</div>

```

5. Definicję kontrolera Home uzupełnij o polecenia z listingu 18.7.

Listing 18.7. Wstawianie rekordów do tabeli Customers

```

private char PobierzZnak(string lancuch, int index)
{
    Random random = new Random();
    char znak;

    if (lancuch.Length > index)
        znak = lancuch[index];
    else
        znak = (char)random.Next(0, 255).ToString()[0];

    return znak;
}

private void UtworzIdentyfikatorKlienta(ref NorthwindDataSet.CustomersRow
↳ daneKlienta)
{
    string identyfikator = "";
    int indexZnaku = 0;

    identyfikator += PobierzZnak(daneKlienta.CompanyName, indexZnaku);
    identyfikator += PobierzZnak(daneKlienta.ContactName, indexZnaku);
    identyfikator += PobierzZnak(daneKlienta.ContactTitle, indexZnaku);
    identyfikator += PobierzZnak(daneKlienta.Address, indexZnaku);
    identyfikator += PobierzZnak(daneKlienta.City, indexZnaku);

    daneKlienta.CustomerID = identyfikator;
}

public ActionResult DodajKlienta(FormCollection daneKlienta, string id)
{
    NorthwindDataSet.CustomersRow cdr =
↳ northwindDataSet.Customers.NewCustomersRow();

    PrzepiszDaneKlienta(daneKlienta, ref cdr);

    UtworzIdentyfikatorKlienta(ref cdr);
}

```

```
customersTableAdapter.Insert(cdr.CustomerID, cdr.CompanyName, cdr.ContactName,
    cdr.ContactTitle, cdr.Address, cdr.City, " ", " ", " ", " ", " ");

AktualizujDaneKlientow();

return ScaffoldingList();
}
```

Wstawienie rekordów do tabeli Customers jest analogiczne do ich edycji. Jedyną różnicą jest konieczność utworzenia identyfikatora klienta, gdyż nie jest on tworzony automatycznie, jak to było w przypadku tabeli Employees. Identyfikatory rekordów w tabeli Customers są pięcioelementowymi ciągami znaków. Tworzę je na podstawie pierwszych liter pól CompanyName, ContactName, ContactTitle, Address oraz City. Jeśli w tych polach znajdują się puste ciągi, to odpowiednią literę zastępuję losowo wygenerowanym znakiem.

W kolejnym podrozdziale pokażę, w jaki sposób wyświetlić informacje o rekordach potomnych, opierając się na relacji pomiędzy tabelami.

Formularz z podformularzem (Master/Detail Form)

Jako podsumowanie tego rozdziału utworzę formularz z podformularzem, tak zwany formularz *Master/Detail*. Jego działanie polega na tym, że wyświetlanie danych odbywa się na bazie co najmniej dwóch komponentów (np. bloków czy tabel) oraz relacji między tabelami. Pierwszy komponent prezentuje rekordy zawarte w tabeli głównej (ang. *master*), a drugi zawiera dane z tabeli podrzędnej (ang. *detail*). Komponent związany z tabelą podrzędną ma zazwyczaj za zadanie prezentować szczegółowe dane odpowiadające rekordowi z tabeli nadrzędnej, który został wskazany przy pomocy związanej z nim listy rekordów. Pola tych tabel muszą być zatem uporządkowane w taki sposób, aby pokazywały informację o tym samym rekordzie. Uzyskuje się to przy pomocy relacji między tabelami, której zadaniem jest powiązanie informacji zawartych w bazie danych.

W tym podrozdziale tabelą nadrzędną będzie tabela Customers, a podrzędną Orders. Po kliknięciu na hiperłącze z etykietą zamówienia wyświetli się lista rekordów tabeli Orders skojarzonych z danym klientem. Żądania prezentacji zamówień będą realizowane asynchronicznie.

1. Plik widoku *ScaffoldingList.cshtml* zmodyfikuj według wzoru z listingu 18.8.

Listing 18.8. Konfiguracja hiperłącza umożliwiającego wyświetlenie listy zamówień danego klienta

```
...
@foreach (var item in Model) {
    <tr>
        <td>
```

```

        @Html.DisplayFor(modelItem => item.CustomerID)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.CompanyName)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.ContactName)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.ContactTitle)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Address)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.City)
    </td>
    <td>
        @Html.ActionLink("Edytuj", "ScaffoldingEdit", new { id =
            ↪item.CustomerID }) |
        @Ajax.ActionLink("Zamówienia", "ScaffoldingMasterDetail", new { id =
            ↪item.CustomerID },
            new AjaxOptions { UpdateTargetId = "zamowienia", OnSuccess =
                ↪"document.getElementById('zamowienia').scrollIntoView()" }) |
        @Html.ActionLink("Usuń", "ScaffoldingDelete", new { id =
            ↪item.CustomerID })
    </td>
</tr>
}
</table>

<div id="zamowienia"></div>

```

2. Utwórz widok częściowy o nazwie *ZamowieniaPartialView.cshtml* i wstaw w nim polecenia z listingu 18.9.

Listing 18.9. Definicja widoku prezentującego listę zamówień

```

@model IEnumerable<NorthwindMVC.Models.NorthwindDataSet.OrdersRow>

@if (Model == null)
{
    <h2>Brak zamówień</h2>
}
else
{
    <h2>Lista zamówień</h2>

    <table>
        <tr>
            <th>
                OrderDate
            </th>
            <th>
                RequiredDate
            </th>

```

```

        <th>
            ShippedDate
        </th>
        <th>
            ShipName
        </th>
        <th>
            ShipAddress
        </th>
        <th>
            ShipCity
        </th>
        <th></th>
    </tr>

    @foreach (var item in Model)
    {
        <tr>
            <td>
                @item.OrderDate.ToShortDateString()
            </td>
            <td>
                @item.RequiredDate.ToShortDateString()
            </td>
            <td>
                @item.ShippedDate.ToShortDateString()
            </td>
            <td>
                @item.ShipName
            </td>
            <td>
                @item.ShipAddress
            </td>
            <td>
                @item.ShipCity
            </td>
        </tr>
    }
</table>
}

```

3. Kontroler Home uzupełnij o metodę `ScaffoldingMasterDetail`, której definicję przedstawiłem na listingu 18.10.

Listing 18.10. *Pobranie i prezentacja rekordów potomnych wybranego klienta*

```

public ActionResult ScaffoldingMasterDetail(string id)
{
    NorthwindDataSet.CustomersRow cdr =
        ↪northwindDataSet.Customers.FindByCustomerID(id);
    NorthwindDataSet.OrdersRow[] zamowieniaPracownika = null;

    if (cdr != null)
        zamowieniaPracownika =
            ↪(NorthwindDataSet.OrdersRow[])cdr.GetChildRows("FK_Orders_Customers");
}

```

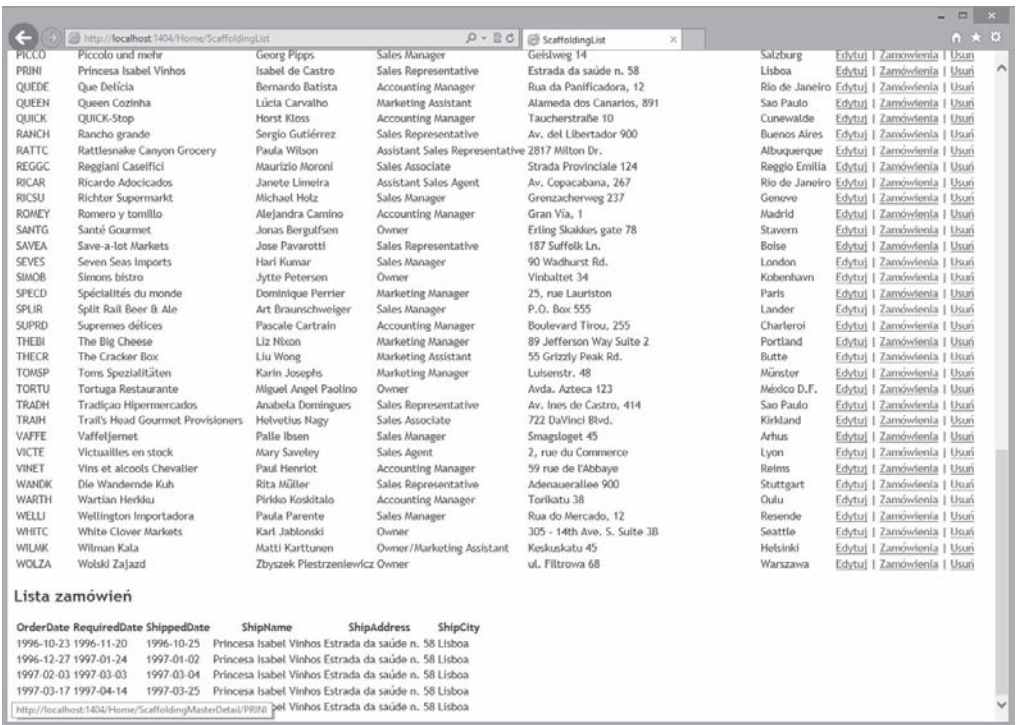
```
if (Request.IsAjaxRequest())
    return View("ZamowieniaPartialView", zamowieniaPracownika);
else
    return ScaffoldingList();
}
```

4. Sekcję <head> pliku *_Layout.cshtml* uzupełnij o polecenia z listingu 18.11.

Listing 18.11. Import skryptów obsługujących żądania asynchroniczne

```
<script src="@Url.Content("~/Scripts/jquery.unobtrusive-ajax.js")"
    type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/MicrosoftAjax.js")"
    type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/MicrosoftMvcAjax.js")"
    type="text/javascript"></script>
```

5. Po skompilowaniu i uruchomieniu aplikacji powinienś uzyskać efekt analogiczny do przedstawionego na rysunku 18.6.



Rysunek 18.6. Przykład formularza typu Master/Detail

Skorowidz

A

- ACL, 149
- action methods, 25
- ActiveX Data Object, 238
- ADO, 238
- ADO.NET, 238
 - obiekty, 238
 - podstawy, 237
 - synchronizacja buforu ze źródłem danych, 273
- adresy URL
 - autoryzacja, 157
 - parametry, 122
 - trasowanie, 121
 - wzorce, 122, 123
- agile programming, 15
- AJAX, 91
 - aktualizacja bloku witryny, 113
 - dodanie referencji do bibliotek JavaScript, 94
- akcja
 - Create, 354
 - Delete, 357
 - Details, 353
 - dodająca nowy produkt, 355
 - Edit, 356
- aktualizacja
 - zawartości elementu witryny, 92
- algorytm
 - RSA, 187
- ambient transaction, 329
- ankieta, 102
- AntiXss, 187
- aplikacje
 - bazodanowe, 235
 - bezpieczeństwo, 335
 - mechanizm transakcji, 319
 - scaffolding, 301
 - synchronizacja ze źródłem danych, 273
 - uwagi do projektowania, 345
 - możliwość edytowania wpisów, 441
 - rejestrowanie zdarzeń i trasowanie, 430
- Application Domain, 339
- architektura obiektowa, 347
- ASP.NET
 - konfiguracja aplikacji, 161
 - mechanizmy uwierzytelnienia, 149
 - metody globalne, 107
 - włączanie i wyłączanie aplikacji, 171
- ASP.NET AJAX, 91
- ASP.NET MVC, 15
 - komponenty, 16
- ASPX, 21, 30
- asynchroniczne przetwarzanie żądań http
 - jQuery, 113
- Asynchronous JavaScript and XML, 48
- Asynchronous Programming Model, 464
- ataki
 - rodzaje, 147
 - SQL Injection, 148, 344
 - wyludzenie informacji, 148
 - złośliwy kod, 181
- atrybut, 347
 - AcceptVerbs(Http.Post), 47
 - AcceptVerbs(HttpVerbs.Post), 65
 - AssociationAttribute, 58
 - Authorize, 200
 - AuthorizeAttribute, 128
 - autoryzacyjny, 145
 - ConcurrencyCheckAttribute, 58
 - Confirm, 95
 - cookieless, 154
 - CustomValidationAttribute, 58, 66
 - data-bind, 230
 - DataType, 54, 57

atrybut

- DataTypeAttribute, 58
- defaultRedirect, 173
- defaultUrl, 154
- DisplayAttribute, 58
- DisplayColumnAttribute, 58
- DisplayName, 436
- EditableAttribute, 58
- enableCrossAppRedirects, 155
- EnumDataTypeAttribute, 58
- FilterUIHintAttribute, 58
- HandleError, 138, 175
- HandleErrorAttribute, 129
- HttpMethod, 95
- InsertionMode, 95
- KeyAttribute, 58
- klasy do konfiguracji metadanych, 57, 58
- LoadingElementDuration, 110
- LoadingElementId, 95, 110
- Log
 - klasa implementująca, 142
- loginUrl, 154
- MetadataType, 69
- MetadataTypeAttribute, 58
- mode, 153, 173
- modelu, 55
- name, 155
- NonAction, 26, 135
- OnBegin, 95
- OnComplete, 95
- OnFailure, 95
- OnSuccess, 95, 110
- OutputCache, 133
- OutputCacheAttribute
 - parametry, 134
- passwordFormat, 156
- protection, 155
- Range, 54, 66
- RangeAttribute, 58
- redirectMode, 173
- RegularExpressionAttribute, 58
- Required, 54, 66
- RequiredAttribute, 58
- RequireHttps, 135
- requireSSL, 155
- slidingExpiration, 155
- StringLength, 66
- StringLengthAttribute, 58
- TestClass, 29
- TestMethod, 29
- ticketCompatibilityMode, 155
- timeout, 155
- TimestampAttribute, 58
- UpdateTargetId, 95

Url, 95

- ValidateInput, 185

atrybuty modelu

- wyrażenia regularne, 84

Attribute, 347

- authentication ticket, 151

- auto post back, 217

- autocomplete textbox, 115

- automatyczne uzupełnienie pola tekstowego, 115

- autoryzacja, 149, 157

- adresów URL, 157

- konfiguracja, 157

- grupy użytkowników, 158

- mechanizm filtrowania, 158

- otwarta, 460

- plików, 157

- zdalna

- powiązanie z portalem Facebook, 461

- Azure, 496

B

- bazy danych, 40, 237

- aktualizacja, 41

- lista rekordów, 301

- Northwind, 240

- detale produktu, 352

- dodawanie połączeń, 350

- pobieranie produktów, 351

- połączenie ze źródłem danych, 241

- relacyjne, 237

- schemat CRUD, 301

- transakcje, 319

- tworzenie, 321

- nowego rekordu, 48

- schematu na bazie modelu encji, 357

- usuwanie danych, 290

- reguły integralności, 293

- usuwanie powiązanych wierszy

- rekordy potomne, 291

- warstwa dostępu, 42

bezpieczeństwo

- aplikacji internetowych, 147

- ataki, 147

- mechanizmy, 149

biblioteka .NET

- programowanie asynchroniczne, 464

biblioteka skryptów

- Mapa.js, 32

- bilet uwierzytelniający, 151

błędy

- działania aplikacji, 171

- strona błędu, 176

- tworzenie użytkownika, 202

- wprowadzenie złego rekordu, 268

body, 383
buforowanie
 danych ze źródeł zewnętrznych, 133
bundling, 475
Business Intelligence, 368

C

callback, 35
CAS, 151, 338
CDN, 113
Central Authentication Service, 151
centralna usługa uwierzytelnienia, 151
certyfikat, 337
CLR
 procesy systemowe, 466
Code Access Security, 338
Confirm, 96
connString, 341
controllers, 16
Coordinated Universal Time, 155
Create Read Update Delete, 301
cross-site scripting, 181
CRUD, 54, 301
Crystal Reports, 368

D

dane
 agregacja, 378
 bezpośredni dostęp do źródła, 346
 eksport, 413
 do dokumentu tekstowego, 419
 do PDF, 420
 skoroszyt MS Excel, 421
 tabele dynamiczne, 418
 integralność, 288, 319
 pominięcie sprawdzania, 292
 reguły, 291
 modyfikacja, 249
 platforma modelowania, 347
 prezentacja
 graficzna, 406
 w postaci macierzowej, 390
 przekazywanie
 z formularza do źródła, 46
 z kontrolera do widoku, 437, 439
 serializacja i transfer struktur, 117
 szyfrowanie, 337
 typu
 ActionResult, 25
 walidacja, 53
 weryfikacja, 53
 więzy integralności, 306

 wyświetlanie, 315
 zabezpieczenie, 338
Data Binding, 389
Data Definition Language, 345
Database Management System, 237
DBMS, 237
 więzy integralności, 238
DDL, 345
denial of service, 147
detail, 315
Development Server, 504
DIE, 53
DisplayFormatAttribute, 58
document object model, 92
DOM, 92
domena aplikacji, 339, 466
 instancja obiektów, 341
 tworzenie, 339
 współdzielenie obiektu, 341
dostawcy usług, 178
dostęp anonimowy, 150
DotNetOpenAuth, 149
DPAPI, 187
DRY, 53
dyrektywa
 @model, 46

E

element
 customErrors, 173
 success, 115
elevation of privilege, 147
encapsulation, 107
encja, 347
 atrybuty, 54
 bazodanowa, 54
 KategoriaProduktu, 358
 odzworowanie w bazie danych, 360
 Order_Details, 362
 Orders, 361
 Products, 361
 Produkt, 358
Entity, 347
Entity Framework, 43, 347
 POCO, 360
 potwierdzenie explicite, 356
 technologie, 349
 wersje, 347
Entity Relationship Model, 347
Entity SQL, 349
ERM, 347
Event-based Asynchronous Pattern, 465
event-driven programming, 17

F

- File Transfer Protocol, 495
- filtry, 127
 - akcji, 127, 139
 - autoryzacyjne, 127, 144
 - dynamiczne, 395
 - kolejność wykonywania, 137
 - priorytety, 138
- Log
 - metody, 141
- LogAttribute
 - definicja klasy, 140
- podstawowe atrybuty, 128
- statyczne, 395
- własne, 139
- wyjątków, 128, 144
 - niestandardowe, 145
- wyniku, 128, 144
- for, 434
- foreach, 51
- Forms Authentication Provider, 149
- formularz
 - Master/Detail, 315
 - z podformularzem, 315
- funkcja
 - delay, 217
 - ListaPracowników, 57
 - Mid, 390
 - Roles.IsUserInRole, 217
 - Today, 390
 - wyznaczająca trasę, 109
 - zmienPrzynaleznoscDoGrupy, 217, 218

G

- Gauge, 383, 403
- GDI, 368
- graficzny interfejs użytkownika, 17, 30
- Graphics Device Interface, 368
- GridView, 48
- grupy użytkowników
 - autoryzacja, 158
- GUI, 17

H

- hard-coding, 167
- haszowanie, 150
- helper methods, 25
- html helper, 455

I

- identyfikator
 - klienta, 315
 - rekordów, 315
- IIS, 121, 149, 495
 - instalacja
 - ASP.NET 4/4.5, 498
 - serwera w Windows 7, 496
 - serwera w Windows 8, 496
 - menedżer Windows, 499
 - uruchamianie aplikacji, 503
 - serwer do rozwoju aplikacji, 504
- information disclosure, 147
- instancje aplikacji, 465
- instancjonowanie modelu, 39
- IntelliSense, 247
- interfejs użytkownika
 - przycisk umożliwiający dodanie pracownika, 294
 - przycisk usunięcia wybranego rekordu, 289
- Internet Information Services, 121
- internetowe usługi informacyjne, 495
- inżynieria społeczna, 148
- Isolation Level, 332
- izolacja, 339, 466

J

- JavaScript Object Notation, 117
- jQM, 457
- jQuery
 - autocomplete, 116
 - pliki źródłowe, 113
- jQuery Mobile, 457
- jQuery UI, 367
 - wykorzystanie przycisków, 284
- JSON, 117
 - obiekt typu adres, 118

K

- Key Performance Indicators, 403
- klasa
 - AccountController, 198
 - ActionExecutingContext, 143
 - AdminController, 212
 - Ajax, 94
 - AjaxOptions, 94
 - atrybuty, 95
 - AspNetHidden, 442
 - AssemblyName, 170
 - Assert, 29

- AuthConfig, 461
- BundleConfig, 475
- Configuration, 170
- ConfigurationManager, 170
- ConfigurationSection, 187, 188
- ConnectionStringBuilder, 345
- DpapiProtectedConfigurationProvider, 187
- FiltrRaportuMacierzowego, 397
- Form1, 484
- FormsAuthentication, 152
- Global, 430
- HandleErrorInfo, 133
- HtmlHelper, 437
- HttpClient, 483, 486
- HttpResponseMessage, 486
- kontekstu, 362
- LogOnModel, 198
- MarshallByRefObject, 341
- Membership, 152, 198, 211
- MvcApplication, 430
- NorthwindContext, 362
- ObjectContext, 348
- ObjectQuery<T>, 348
- Order_Details, 362
- Orders, 361
- Osoba, 228
- ParametryWatku, 470
- Pracownik, 484
- Products, 361
- Produkt, 230
- Raport, 392
- Request.Browser, 460
- ResultExecutedContext, 144
- ResultExecutingContext, 144
- Roles, 211
- RsaProtectedConfigurationProvider, 187
- SQLConnCAS
 - definicja, 340
- stowarzyszona z modelem, 69
- System.Net.Http.Formatting.JsonMediaTypeF
 - ormatter, 491
- System.Text.RegularExpressions, 74
- System.Web.Security.Roles, 159
- Task, 465, 472
- testująca, 29
- Thread, 465, 469
- TransactionScope, 327
- UrlRoutingModule, 121
- ViewData, 434, 436, 439
- WebApiConfig, 478
- WebConfigurationManager, 170
- Wpisy, 427, 436
- klucze, 237
 - MAC, 155
 - obce, 238
 - podstawowe, 238
- kluczowe wskaźniki efektywności, 403
 - dynamiczna zmiana wartości, 404
- Knockout, 227
 - wykorzystanie szablonów JavaScript, 230
 - wywołanie skryptu, 229
- kodowanie danych
 - przekazywanych przez użytkowników, 186
- komponent
 - FormDanePracownika, 489
 - obiektu
 - PartialView, 99
 - typu ListBox, 99
 - ReportViewer, 368, 375
 - eksport danych, 413
 - nawigacja między stronami raportu, 377
- komponenty użytkownika, 30
- komunikaty o błędach, 171
 - kod statusowy, 174, 175
- koncepcja programowania zdarzeniowego, 16
- konfiguracja
 - połączenia z bazą danych, 241
 - uprawnień puli aplikacji, 454
 - walidacji danych, 66
- konkurencyjność aplikacji, 332
- kontekst wiązania, 389
- kontrola wprowadzanych danych, 263
 - metoda onColumnChanged, 269
 - wstrzymanie walidacji wiersza, 267
 - zakończenie edycji rekordu, 267
 - wymuszenie, 267
- kontroler, 16, 25
 - AccountController, 195
 - Admin, 223
 - AdminController, 212
 - aplikacja Kardy, 63
 - asynchroniczny, 466
 - a zadania, 472
 - dane wyjściowe, 25
 - domeny, 150
 - ErrorController, 174
 - Głosowania, 103
 - Home, 182, 224, 228, 231
 - Bezpieczeństwo, 336
 - filtr wyjątku, 131
 - HomeController, 184
 - EksportDanych, 415
 - Northwind, 245
- interfejs Web API, 477
- KsiegaGosciController, 424

kontroler

- Lista, 93
- Mapa, 33
 - autozupelnianie, 117
 - modyfikacja, 45
- niedostępne metody publiczne, 136
- obsługa POST, 431
- panelu administracyjnego, 212
 - zarządzanie grupami użytkowników, 220
- Pracownicy, 57, 64
- PracownicyController
 - implementacja, 481
- Produkty, 351
- projektowanie, 27
- przekazywanie danych do widoku
 - zmiany, 439
- uwierzytelnienia, 195
 - klasa, 198
 - model, 196
 - widoki, 203
- uzupełnienie o widok, 31
- użycie modelu odsyłanego przez widok, 440
- ValuesController, 478
- weryfikacja funkcjonalności, 28
- wiązanie adresu URL z akcją, 430
- widoki, 58

konwersja obrazu, 278

kreator

- Add New Item, 242
- Add View, 59, 302
- Application, 162
- blokowania szczegółów błędów, 172
- dodanie użytkowników, 164
- grup użytkowników, 165
- Provider, 162
- Publish Web, 501
- Security, 162
- tworzenia pracownika, 59
- ustawień aplikacji, 167
- widoków, 60
- wybór dostawców usług, 178

kwerendy parametryczne, 274, 401

L

layout witryny, 30

LINQ

- to Entities, 349

lista rekordów, 301

- prezentacja, 304

logika

- aplikacji, 30
- bazodanowa, 30
- biznesowa, 44

Ł

łańcuch QueryString, 26

M

master, 315

mechanizm

- automatycznych właściwości, 57
- autoryzacji, 157
- eksportu danych, 413
- filtrów, 127
- grupowania plików, 475
- IntelliSense, 119
- Query-String, 117
- szyfrujący, 187
- transakcji, 319, 321
 - przelewy bankowe, 326
- wyznaczania trasy przejazdu, 34

meotda

- Insert, 299
- ProtectSection, 191
- WersjonowanieDataRow
 - definicja widoku, 265

message authentication code, 155

metoda

- .on, 262
- @Html.ValidationSummary, 204
- About, 124
- AcceptChanges, 255, 263
- ActionLink
- ActionLink, 61, 63, 353
- AddUserToRole, 159
- akcji, 25
 - parametry opcjonalne, 27
- aktualizacja statystyk ankiety, 105
- AktualizujZdjeciePracownika, 288
- Application_AuthenticateRequest, 108
- Application_BeginRequest, 107, 108
- Application_End, 108
- Application_EndRequest, 107
- Application_Error, 108, 172
- Application_Start, 108, 123, 139, 430
- asynchroniczna kontrolera, 472
- AttachTo, 356
- Authenticate, 152
- AutoUzupelnienie, 117
 - zapytanie LINQ, 118
- BeginEdit, 267
- BeginForm, 47, 94
- BeginTransaction, 326, 332
- CancelEdit, 263
- ChangeObjectState, 356

ChangePassword, 200
 hiperłącze, 208
 widok, 206
ChangePasswordSuccess
 widok, 206
Create{NazwaEncji}, 354
CreateRole, 159
CreateUser, 153
czyBazaDanychIstnieje, 326
daneDostawcyForm, 262
DBeginEdit, 263
Delete, 270
DeleteObject, 357
DeleteRole, 159
DeleteUser, 153
DeszyfrujPlikKonfiguracyjny, 191
DodajDostawce, 255
DodajElementDoListy, 94, 97, 101
DodajGrupe, 223
DodajKomentarz, 185
DodajPracownika, 299, 491
DopiszDoPlikuLog, 430
Edit, 356
EdytujDaneDostawcy, 261
EdytujDanePracownika, 493
Eksportuj, 418
EmptyResult, 26
EndEdit, 263
EndForm, 47
ErrorCodeToString, 201
Fill, 246
FillByCityName, 275
FormatDateTime, 390
FormsAuthentication.SetCookie, 199
GenerujDane, 468
Get(int id), 478
GetAllRoles, 159
GetAsync, 486
GetChildRecords, 291
GetSection, 170
GetSectionGroup, 170
GetUser, 153
GetUsersInRole, 160
globalna
 aplikacji ASP.NET, 108
Grupy, 223
HasVersion, 263
Html.ListBox, 97
Html.RenderPartial, 101
Html.ValidationSummary, 61
HtmlHelper.Label, 437
ImiePierwszegoPracownika, 246, 247
Include, 475
Index, 47, 79, 97, 101
 kontrolera Mapa, 34
 testowanie, 28
IsAjaxRequest, 101
IsMatch, 76
IsNotNull, 29
IsUserInRole, 159
Label, 35
ListaDostawcow, 252, 255
ListaPracownikow, 248
 hiperłącze skojarzone, 249
LogOff, 198
LogOn, 198
Macierzowy, 397
mapowalna, 26
MetodaAsynchronicznaWatki, 472
ModelState.AddModelError, 199
MvcApplication.RegisterRoutes, 430
obsługa zapytań POST, 431
odczyt aktualnych bilansów kont, 323
OdczytajStanyKont, 326
OnActionExecuted, 143
OnActionExecuting, 127, 143
onColumnChanged, 269
OnResultExecuted, 144
OnResultExecuting, 144
onRowChanged, 266
onRowChanging, 268
OpenConfiguration, 170
otwarcie połączenia z bazą danych, 341
OtworzPlikDanych, 107
Page_Load, 399
PartialViewResult, 26
PobierzDane, 394
 modyfikacja, 401
PobierzPlikLogu, 141, 143
PobierzZdjecie, 278
PokazTrasy, 51
PokazWyniki, 107
pomocnicza, 25, 455
 Html.Encode, 51
 wykorzystanie, 456
PostAsJsonAsync, 486
PostAsXmlAsync, 486
PostAsync, 486
PostResolveRequestCache, 121
PracownicyWedlugMiast
 przechowywanie wartości filtru, 285
PutAsJsonAsync, 493
PutAsXmlAsync, 493
PutAsync, 493
ReadAsAsync, 486
ReadXml, 107
ReadXmlSchema, 107
RedirectFromLoginPage, 152

metoda

- RedirectResult, 26
- RedirectToAction, 26
- RedirectToLoginPage, 152
- RedirectToRoute, 26
- RedirectToRouteResult, 26
- RefreshReport, 384
- Register, 199, 478
- RegisterAuth, 461, 462
- RegisterBundles, 475
- RegisterGlobalFilters, 171
- RegisterRoutes, 94, 123
- RejectChanges, 256, 263
- rejestrwanie zdarzeń aplikacji, 430
- RemoveUserFromRole, 160
- Render, 384, 413, 418
- RenderPartial, 99
- Response.Write, 223
- Result.ExecuteResult, 143
- RoleExists, 159
- Roles.IsUserInRole, 216
- Roles.RemoveUsersFromRole, 223
- route, 35
- Rows.Add, 255
- RozpocznijEdycje, 269
 - hiperłącze uruchamiające, 267
- SaveChanges, 354
- ScaffoldingMasterDetail, 317
- Session_End, 108
- Session_Start, 108
- SetAuthCookie, 152
- SetParameters, 401, 405
- SignOut, 152
- SprawdzDopasowania, 79
- SprawdzPoprawnoscDatyZatrudnienia, 67
- SqlPermission.Add, 343
- SqlTransaction.Commit, 326
- SqlTransaction.Rollback, 326
- StronaNieIstnieje, 175
- submit, 115
- SzyfrowaniePlikuKonfiguracyjnego, 190
- szyfrująca i deszyfrująca, 189
- SzyfrujPlikKonfiguracyjny, 191
- TextBox, 35
- Thread.Sleep, 469, 472
- TransactionScope.Complete, 329
- Transfer, 36
- UpdateUser, 153
- UstawParametryRaportu, 399
- UsunGrupe, 223
- UsunTrase, 50, 51
- UtworzPracownika, 64
- Uzytkownicy, 216
- ValidateUser, 152, 153
- ValidationMessage, 437
- ValidationMessageFor, 61
- ValidationSummary, 437
- ViewResult, 26
- WebApiConfig.Register, 478
- WersjonowanieDataRow, 265
 - sparametryzowanie, 266
- Wpisy.ZapiszDoPlikuXml, 430
- wykonywanie przelewów, 323
- WyswietlLog, 141, 142, 143
- wyswietlMape, 34, 35
- WyswietlTrase, 50
 - modyfikacja, 135, 166
- wyznaczTrase, 35
- ZaktualizujWyniki, 107
- ZapamiętajNazwęUzytkownika, 432
- ZmienStanKont, 328
- Microsoft SQL Server, 319
 - sposoby uwierzytelnienia, 344
- MIME type, 390
- minification, 476
- minimalizacja, 476
- model, 16, 39
 - APM, 464
 - ChangePasswordModel, 196
 - EAP, 465
 - integracja z elementami aplikacji, 39
 - LogOnModel, 196
 - obsługa plików XML, 427
 - przesyłanie z kontrolera do widoku, 432, 435
 - RegisterModel, 196
 - TAP, 465
- model binding, 48
- model danych encji, 348
 - dodawanie encji, 353
 - edycja encji, 355
 - kreator, 348, 349
 - łączenie encji, 359
 - metody akcji, 354
 - opcje, 361
 - projektowanie relacji, 43
 - tworzenie, 349
 - schematu bazy danych, 357
 - usuwanie encji, 357
 - wybór tabel, 350
- model first, 52
- model związków encji, 347
- models, 16
- Model-View-Controller, 15
- Model-View-ViewModel, 227
- Model-Widok-Kontroler, 15
- moduł
 - UrlAuthorizationModule, 157

MVC, 15

- a Web Forms, 16, 365, 367, 423
 - kod, 442
 - serwer internetowych usług informacyjnych, 495
 - wydajność, 443
 - wykorzystanie w projektach, 444
- ataki XSS, 185
- autoryzacja, 158, 162
- bezpieczeństwo aplikacji, 152
- buforowanie, 134
- domyślne wzorce aplikacji, 21
- kreator aplikacji, 20
- mechanizm filtrowania, 127
- Model Binder, 355
- nazwa kontrolera, 27
- obsługa interakcji klient-serwer, 121
- parametry opcjonalne, 27
- pobieranie parametru z adresu URL, 26
- projektowanie aplikacji, 18, 52
- przetwarzanie żądania aplikacji klienckiej, 18
- sprawdzanie poprawności danych, 68
- struktura modelu, 39
- techniki walidacji danych, 53
- utworzenie widoku, 30
- uzupełnianie o komponenty graficzne, 367
- walidacja, 438
- wersja 4, 449
- wybór mechanizmu przetwarzającego widoki, 21
- wzorce URL, 122
- zbiór modeli szablonu aplikacji, 196
- źródło danych, 40

MVC4, 449

- grupowanie i optymalizacja skryptów, 475
- kaskadowe arkusze stylu, 475
- przetwarzanie asynchroniczne, 464
- przystosowanie do urządzeń mobilnych, 457
- testowanie aplikacji, 457
- uwierzytelnianie, 460
 - integracja z portalem Facebook, 462
- wsparcie dla urządzeń mobilnych, 450
- zdalna autoryzacja, 461

MVVM, 227

N

narzędzie

- aspnet_regiis, 187
- przetwarzanie tekstu, 73

Northwind

- instalacja, 240
- kwerenda wybierająca, 240
- pobieranie informacji o dostawcach, 251

NuGET, 483

nullable-types, 27

O

OAuth, 461

obiekt

- AppDomain, 339, 341
- Chart, 406
- Command, 239
- Connection, 239, 241
 - właściwości, 243
- ConnectionString, 343
 - sposoby uwierzytelniania, 344
 - szyfrowanie informacji, 335
- DataAdapter, 239
- DataGridView, 491
- DataReader, 239, 273
- DataRow, 255
 - kontrola wprowadzonych danych, 263
 - usuwanie rekordów, 270
 - wersje, 263
 - wersja proponowana, 265
 - wersjonowanie, 263
 - właściwości, 255
- DataSet, 239, 244, 345
 - edycja rekordów, 257
 - kreator konfiguracji TableAdapter, 274
 - modyfikacja danych, 249
 - tworzenie nowych rekordów, 250
 - usuwanie rekordów, 271
 - wstawianie rekordów, 251
 - wymuszenie więzów integralności, 292
- DataTable, 107
- DirectionsService, 35
- dostawca, 153
- FormCollection, 47
- jqXHR, 218
- List, 389
- logiki biznesowej, 348
- materializacja, 348
- Matrix, 390
- Membership, 152
- MvcRouteHandler, 121, 122
 - o określonym typie danych, 244
 - o silnej typizacji, 244
 - o słabej typizacji, 247
- ObjectStateManager, 348
- PartialView, 98
- ReportViewer, 368
- ReportViewer.LocalReport, 418
- Request, 26, 101, 432
- Response, 26
- Route, 122
- ShippersRow, 255
- SqlConnectionCAS, 341
- SqlConnection, 326

obiekt

- TableAdapter, 239, 244, 273, 274, 345
 - edycja danych, 279, 286
 - wstawianie danych, 294

- TransactionScope, 329
 - parametryzacja, 331

typu

- adres, 118
- AppDomain, 466
- ConcurrentDictionary, 479
- NorthwindDataSet, 247
- ObjectSet<T>, 348
- Route, 121
- Thread, 466

- UrlRoutingModule, 121

- ViewBag, 36
 - przekazywanie danych do widoku Web Forms, 376

- ViewData, 36, 79, 432

Object Relational Mapping, 347

Object Services, 348

obsługa błędów

- konfiguracja, 132
- włączenie, 129

odrzućcie informacji o transakcji, 148

odświeżanie witryny

- a fragment widoku, 97
- aktualizacja zawartości, 113
- częściowe, 96

OnSuccess, 96

operator

- await, 474
- konkatenacji, 382
- Top %, 411
- Top N, 411

ORM, 347

P

parametr

- CountryName, 401
- data, 115
- filtr, 117
- mapDivId, 35
- maxIloscRekordow, 117
- Order, 137
- raportu, 395
- scopeTimeOut, 330
 - wartości, 330
- TransactionScopeOption
 - wartości, 329
- type, 115

parametryzacja kwerend, 274

partial view, 30

piaskownica, 341

Plain Old CLR Object, 360

platforma .NET

- usługi uwierzytelniania, 153

plik

- ascx, 22
- aspx, 22
- bmp, 390
- cookie, 151
- cookies, 154, 155
- cshtml, 22, 434
- css, 475
- edmx, 360
- jpg, 390
- js, 475
- konfiguracyjny
 - zabezpieczenie, 187
- master, 22
- png, 390
- rdl, 368
- rdlc, 368
- tekstowy
 - wyświetlenie w przeglądarce, 143
- vbhtml, 22

POCO, 360

pola klucza, 237

polecenie

- @Html.Encode, 187
- @Scripts.Render, 476
- @Styles.Render, 476
- GET, 18
- POST, 18

połączenie

- ze źródłem danych, 241

pomocnik HTML, 455

Portable Document Format, 413

pośrednik, 341

poziom izolacji, 332

- Chaos, 333
- konfiguracja, 332
- ReadCommitted, 333
- ReadUncommitted, 333
- Serializable, 333
- Snapshot, 333
- Unspecified, 333

poziom zaufania, 339

- ograniczenie, 340

procedura

- ajax, 115
- Ajax.BeginForm, 96
- niemapowalna, 26
- PartialView, 26
- serialize, 115

- proces, 465
- programowanie
 - oparte na zdarzeniach, 17
 - zdarzeniowe, 17
- projekt aplikacji MVC
 - ankieta, 102
 - asynchroniczne przetwarzanie żądań HTTP, 92
 - Bezpieczeństwo, 181, 336
 - nawiązanie komunikacji ze źródłem danych, 342
 - szyfrowanie pliku konfiguracyjnego, 191
 - dodawanie i wyświetlanie komentarzy, 183
 - domyślny, 20
 - EksportDanych, 413
 - Filtry, 131
 - formularz oceny, 107
 - Kadry
 - walidacja adresu e-mail, 84
 - Kardy, 55
 - KsiegaGosci_MVC3, 424
 - lista prezentująca dodawane elementy, 97
 - MojaNawigacja, 18
 - blokowanie samodzielnego tworzenia kont, 224
 - buforowanie danych, 135
 - implementacja warstwy dostępu, 42
 - instalacja na serwerze IIS, 500
 - komunikat powitalny, 168
 - komunikaty, 110
 - komunikaty o błędach, 173
 - mechanizm odpowiedzi, 115
 - model weryfikujący poświadczenia użytkowników, 196
 - modele kontrolera uwierzytelnienia, 196
 - odświeżenie widoku mapy, 109
 - struktura, 21
 - szablon, 19
 - szablon witryny, 225
 - trasowanie żądań HTTP, 123
 - uzupełnienie o bazę danych, 40
 - uzupełnienie o walidację danych, 69
 - widok błędu, 175
 - zablokowanie nieautoryzowanego dostępu, 128
 - zarządzanie użytkownikami, 163
 - zarządzanie grupami użytkowników, 212
 - zarządzanie procesem logowania, 198
- MsAjax, 92
- NorthwindMVC, 241
 - lista rekordów, 301
 - szablon witryny, 250
- POCO, 361
- priorytety reguł filtrowania, 137
- Raport, 368
- szablon
 - mechanizmy uwierzytelnienia, 156
 - zarządzanie grupami użytkowników, 159
- Transakcje, 319
 - aktualizacja danych, 328
 - formularz aktualizacji, 328
 - szablon stron, 320
 - uzupełnianie widokiem GridView, 49
- WyrażeniaRegularne, 76
- Zapytania_EF, 349
- projekt aplikacji MVC4
 - AsynchronicznyKontroler, 466
 - prezentacja danych, 468
 - AutoryzacjaOAuth, 461
 - implementacja serwisu sieciowego, 479
 - konfiguracja grupowania, 475
 - MobileMvc, 450
 - testowanie, 457
 - pobranie zdjęcia, 452
 - przygotowanie listy pracowników, 452
 - SerwisSieciowy, 477
- projekt aplikacji Windows Forms
 - edycja danych, 487
 - KlientWebAPI, 483
 - tworzenie nowych pracowników, 486
 - zadania typu
 - DELETE, 493
 - POST, 486
 - PUT, 491
- projektowanie
 - kontrolera, 27
 - widoku, 31
- protokół
 - FTP, 495
 - Kerberos, 150
 - NTLM, 150
 - SMTP, 495
 - SSL, 337
- proxy, 341
- przechowywanie
 - informacji o trasach, 44
 - wyników głosowania, 107
- przeciążenie aplikacji, 147
- przekazywanie
 - danych, 44
 - między widokiem a kontrolerem, 36
 - parametrów do metod akcji kontrolera, 26
- przestrzeń nazw
 - System.Threading, 465
- przyjazne linki, 122
- publikowanie aplikacji, 495, 500
 - na serwerze IIS, 497

pule aplikacji, 465, 497
konfiguracja, 498
Windows 7, 499
Windows 8, 500

R

raporty
dane w postaci macierzowej, 390
data, 386
definicja
dynamiczna, 418
filtrowanie danych, 394
na poziomie źródła, 401, 418
filtry, 395
formaty prezentacji danych, 379
grupowanie danych, 381
hiperłącza, 387
nawigacyjne, 376
kolumny obliczeniowe, 378
konfiguracja
filtrowania dynamicznego, 396
typu, 392
źródła danych, 368
kreator formuł, 382
macierzowe, 391
numery stron, 386
obrazy, 390
parametry, 383, 394
konfiguracja, 399
paski narzędziowe, 379
pola
bazy danych, 386
formularza, 398
raportu, 390
połączenie z aplikacją, 368
prezentowanie układu graficznego, 383
projektowanie, 371
i prezentowanie, 368
przystosowanie strony do obsługi raportów, 387
renderowanie, 368
tryby definicji, 368
tworzenie
definicji lokalnej, 368
formuł, 378
przy użyciu kreatora, 372
układ, 383
wstawianie nagłówka i stopki, 385
wykresy, 406
wyświetlenie, 375
Razor, 21, 30
przetwarzanie definicji widoku, 208

RDL, 418
rekordy
edycja, 307
lista, 301
tworzenie, 311
usuwanie, 305
Relationship, 347
Report Definition Language, 418
Report Wizard, 371
repudiation, 148
request, 442
response, 442
responsywność, 464
rollback transaction, 319
routing, 22, 430
routowanie, 121

S

SandBox, 341
scaffolding, 301
Secure Socket Layer, 337
sekcja
membership, 178
roleManager, 179
set, 441
serwer
SMTP
konfiguracja, 171
usług informacyjnych, 149
weryfikacja tożsamości użytkownika, 149
Simple Mail Transfer Protocol, 495
słowo kluczowe
@helper, 455
await, 474
using, 47, 329
spaghetti code, 18
spoofing, 147
SQL Server Reporting Services, 367
SSRS, 367
stan pola
typu CheckBox, 269
pobieranie, 264
STRIDE, 148
strona startowa
MojaNawigacja, 226
strona wzorcowa, 30
aplikacji Kadry, 55
referencja do bibliotek skryptów, 70
systemowy serwer usług informacyjnych, 121
szyfrowanie połączenia, 335
ze źródłem danych, 336

T

tabela danych
 Categories, 418
 Customers, 305
 deklaracje kaskadowych arkuszy stylu, 275
 edycja
 danych, 279
 danych w wierszu, 257
 rekordów, 307, 310
 Employees, 240, 288
 filtrowanie listy, 275
 integralność danych, 288
 Konta, 321, 326
 kwerenda wybierająca, 240
 Orders, 291, 315
 OrdersDetails, 307
 pobieranie fotografii, 276
 pobranie danych z wykorzystaniem JScript, 280
 prezentacja
 danych, 277, 325
 informacji, 252
 listy danych, 248
 pierwszego rekordu, 246
 rekordów, 317
 przekazanie do widoku, 248
 Shippers, 251
 tworzenie rekordów, 311
 umieszczenie w bazie danych, 321
 uruchomienie edycji danych, 258
 usuwanie
 danych, 288
 rekordów, 271, 305, 306
 widok błędu, 268
 wstawianie
 danych, 294
 rekordów, 251
 wyświetlenie listy
 konfiguracja hiperłącza, 315
 zakończenie edycji rekordu, 267
tablica asocjacyjna, 479
tablica bajtów
 konwersja do bitmapy, 277
tampering, 147
task pane, 371
Task-based Asynchronous Pattern, 465
TDD, 15
 test jednostkowy funkcjonalności kontrolera, 28
Test-Driven Development, 15
testowanie
 aplikacji, 29
 wzorców wyrażeń regularnych, 76, 78
Toolbox, 274

transakcje, 319
 automatyczne zarządzanie, 327
 otaczające, 329
 poziom izolacji, 332
 tworzenie i ręczna kontrola, 319
trasowanie, 121
 aplikacja, 430
 blokowanie dostępu, 124
 domyślne reguły, 123
 KsięgaGosci_MVC3, 430
 własne reguły, 124
trasowanie adresów URL, 121
T-SQL, 319
 czynności bazodanowe, 345
tworzenie
 bazy danych, 40
 interfejsu użytkownika
 jQM, 459
 szablony, 231
 wzorzec MVVM, 227
 użytkownika, 200
 kody błędów, 201
 widoku Index, 231
 wzorców wyrażeń regularnych, 76
typed DataSet, 244
typizacja
 silna, 46
 słaba, 46
typy puste, 27

U

ujawnienie krytycznych informacji, 147
ukryte pola, 441
Uniform Resource Locator, 18
URL, 18
uruchomienie aplikacji, 503
usługa
 AspNetSqlProvider, 178
ustawienia aplikacji, 167
 odczytanie z pliku konfiguracyjnego, 169
UTC, 155
uwierzytelnienie, 149
 centralny serwis, 151
 certyfikaty, 150
 formularze logowania, 151
 atrybuty elementu forms, 154
konfiguracja, 153
mechanizmy zintegrowane z ASP.NET, 149
na bazie protokołu NTLM, 150
przekazywanie informacji, 151
proste, 150
skrótowe, 150
usługi systemu Windows, 150

V

variable placeholders, 122
 view engine, 21
 views, 16
 viewstate, 442
 Visual Studio
 kreator projektu, 19
 Visual Studio Development Server, 496
 VS 2012
 instalacja, 240
 kreator Add New Item, 242
 VSDS, 496

W

walidacja
 adresu
 e-mail, 83
 URL, 83
 WWW, 84
 danych, 53
 a technologia Entity Framework, 68
 konfiguracja, 66
 modelu Pracownik, 66
 odwołanie do jQuery, 61
 po stronie klienta, 67
 wyrażenia regularne, 73
 skrypty i style walidatorów, 438
 własności klasy encji, 435
 wątek, 465
 Web API, 477
 edycja danych, 491
 klient, 483
 konfiguracja interfejsu, 478
 obsługa żądań typu POST, 486
 pobranie danych z serwera, 485
 serwis sieciowy, 479
 usuwanie danych, 493
 web serwer, 495
 Web Site Administration Tool, 161
 konfigurowanie uprawnień, 165
 zarządzanie
 dostawcami usług, 178
 użytkownikami, 163
 WebForms, 17
 bezpieczeństwo aplikacji, 152
 integracja komponentów z MVC, 367
 kontrolka Repeater, 434
 obsługa wyjątków, 172
 przetwarzanie żądania aplikacji klienckiej, 18
 separacja modelu, 427
 weryfikacja poprawności projektu, 434

weryfikacja
 kodu pocztowego, 81
 tożsamości, 151
 widok, 16, 25, 30
 aplikacji
 MVC4, 453
 przystosowanie do urządzeń mobilnych, 459
 Transakcje, 326
 częściowy, 30
 dodawanie, 31
 edycji klienta, 307
 elementy, 31
 formularz
 dodawania wpisu, 425
 logowania, 203
 kontrolera
 aplikacja Kadry, 58
 panelu administracyjnego, 222
 kontrolera AdminController, 213
 kontrolera Dopasowania, 77
 kontrolera ErrorControler, 174
 kontrolera Głosowania, 103
 kontrolera Home, 131, 169, 182, 307
 metoda WyswietlLog, 142
 Raport, 375
 Transakcje, 320
 kontrolera HomeController, 246
 AsynchronicznyKontroler, 468
 EksportDanych, 417
 kontrolera Lista, 93
 kontrolera Mapa, 33, 34
 formularz, 70
 Trasa, 50
 kreatora
 konta użytkownika, 205
 zmiany hasła, 206
 listy
 klientów, 302
 pracowników, 61
 wyznaczonych tras, 48
 zamówień, 316
 mapy
 odświeżenie, 108
 parametryzacja, 34
 mechanizm przetwarzania, 30
 modyfikowanie formularza, 437
 obsługa wyjątków, 130
 Pracownicy.cshtml, 453
 projektowanie, 31
 transfer danych do kontrolera, 36
 typu CRUD, 52

- typu GridView
 - implementacja, 48
- typy plików, 30
- utworzenie, 30
- uzupełnianie treścią, 33
- walidacja, 435
- wiązanie z modelem, 437
- wstawianie rekordów, 312
- wyników ankiety, 105
- wyświetlanie wpisów, 432
- wielodziedziczenie, 360
- Windows Authentication Provider, 149
- Windows Data Protection API, 187
- Windows Forms
 - implementacja klienta, 483
- Windows Live ID, 149
- właściwość
 - ConnectionString, 243
 - IsMobileDevice, 460
 - ModelState.IsValid, 47
- wstawianie rekordów, 299
- wstrzyknięcie kodu, 181, 185
- wykresy, 406
 - konfiguracja
 - serii danych, 407
 - typu, 407
 - pobranie danych źródłowych, 408
- wyłączenie aplikacji, 171
- wrażenia lambda, 46
- wrażenia regularne, 73
 - atrybuty modelu, 84
 - liczba wystąpień składników, 76
- wzorzec
 - adres URL, 83
 - imię żeńskie, 81
 - kod pocztowy, 280
 - tworzenie, 76
- znaki i cyfry, 74
- znaki specjalne, 74
- znaki w wyznaczonych miejscach łańcucha, 75
- wzorzec
 - odnajdywanie
 - znaków i cyfr, 74
 - znaków specjalnych, 74
 - znaków w wyznaczonych miejscach łańcucha, 75
 - określanie
 - liczby wystąpień znaków, 76

X

XSS, 181

Z

zabezpieczenia

- aplikacji bazodanowych, 335
- ataki XSS, 187
- informacje o błędach, 171
- kodu pośredniego, 338
- konfiguracja, 161
- konfiguracja aplikacji
 - kreatory, 162
- obsługa wyjątków, 171
- pliku konfiguracyjnego, 167, 187
- przed atakami, 148
- ustawienia aplikacji, 167
- witryny, 181
- zarządzanie dostawcami usług, 178
- złośliwy kod, 181

zarządzanie

- użytkownikami i grupami, 211
 - dostęp do panelu administracyjnego, 216
 - lista grup, 221
 - podział na grupy, 220
 - pobieranie listy, 213
 - tworzenie nowej grupy, 222
 - zmiana przynależności do grupy, 215
- złożonością aplikacji, 16

zdarzenie

- DataGridView.CellDoubleClick, 491
- rejestrowanie, 430

złośliwy kod

- działanie, 181

znacznik

- input, 288

znaki specjalne

- odnajdywanie, 74

związanie z modelem, 48

związek, 347

Ż

żądanie, 25

- asynchroniczna obsługa, 92, 466, 470
 - skrypty, 318
 - zadania, 473
- GET, 64
- POST, 64
- przetwarzanie, 127
- zablokowanie obsługi, 124

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA

 **Helion SA**

ASP.NET MVC

Kompletny przewodnik dla programistów
interaktywnych aplikacji internetowych w Visual Studio

ASP.NET MVC to platforma, o której zaletach nie trzeba nikomu przypominać. Potężne narzędzie do budowy aplikacji internetowych działających według wzorca model-widok-kontroler pozwala tworzyć oprogramowanie zapewniające bezkolizyjny przepływ dużych ilości danych. Dzięki temu administratorzy sieci mogą względnie łatwo kontrolować stan aplikacji, a użytkownicy internetu są zadowoleni z szybkości ładowania stron internetowych. Ta książka kompleksowo zapozna Cię z programowaniem aplikacji opartych na technologiach ASP.NET MVC, JavaScript, jQuery oraz AJAX, na przykładzie aplikacji internetowej zaimplementowanej od podstaw w środowisku Visual Studio 2012.

Z części pierwszej dowiesz się więcej o najważniejszych aspektach aplikacji ASP.NET MVC, takich jak projektowanie kontrolerów, dostęp do baz danych, walidacja i ochrona danych, implementowanie interaktywnych i asynchronicznych widoków, trasowanie, filtrowanie i zarządzanie użytkownikami. Poznasz też bibliotekę Knockout, umożliwiającą projektowanie interfejsu użytkownika według wzorca MVVM. Część druga jest poświęcona technologiom ADO.NET i ADO.NET Entity Framework, a także zagadnieniom związanym z transakcjami oraz bezpieczeństwem w aplikacjach bazodanowych. Natomiast w części trzeciej znajdziesz informacje pozwalające zintegrować ASP.NET MVC z ASP.NET WebForms i dowiesz się, co możesz zyskać dzięki takiemu połączeniu. Weź tę książkę do ręki i przetestuj przykłady, a nauczysz się więcej, niż myślisz!

- Podstawowe aspekty projektu aplikacji ASP.NET MVC
- Walidacja danych i wyrażenia regularne
- Asynchroniczne aplikacje internetowe, AJAX a biblioteka jQuery
- Trasowanie adresów URL i filtry
- Bezpieczne aplikacje, konfiguracja zabezpieczeń i panel Web Site Administration Tool
- Podstawy ADO.NET i obiekt TableAdapter
- Scaffolding i transakcje
- Bezpieczeństwo w aplikacjach bazodanowych i Entity Framework
- Integracja komponentów WebForms z aplikacją ASP.NET MVC, tworzenie raportów i eksport danych
- ASP.NET MVC 4 i tworzenie serwisów sieciowych WebAPI
- Mobilne aplikacje internetowe

**Posłuż się ASP.NET MVC
i stwórz wspaniałą aplikację!**

helion.pl
księgarnia
internetowa

Nr katalogowy: 8355



Księgarnia internetowa:
<http://helion.pl>



Zamówienia telefoniczne:
0 801 339900
0 601 339900



Helion

Sprawdź najnowsze promocje:
• <http://helion.pl/promocje>
Książki najchętniej czytane:
• <http://helion.pl/bestsellery>
Zamów informacje o nowościach:
• <http://helion.pl/nawosci>

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

sięgnij po **WIĘCEJ**



KOD KORZYŚCI

ISBN 978-83-246-3238-1



9 788324 632381

Informatyka w najlepszym wydaniu

Cena: 89,00 zł