Gliwice, Maj 2004

Tworzenie programów dla platformy .NET

ASP.NET

Odsłona trzecia

Marek Mittmann

Plan wykładu

- Kontrolki weryfikujące
- Kontrolki list i wiązanie danych
- ADO.NET w aplikacjach ASP.NET
- Kontrolki użytkownika
- Korzystanie z pamięci podręcznej
- Testowanie i uruchamianie stron
- Uwierzytelnianie

Kontrolki weryfikujące

- ReguiredFieldValidator
  - Sprawdza czy pole jest puste
- RangeValidator
  - Sprawdza czy wartość mieści się w zakresie
- CompareValidator
  - Porównuje wartości
- CustomValidator
  - Wywołuje funkcję sprawdzającą użytkownika
- ValidationSummary
  - Łączy informacje o wyniku walidacji

Kontrolki list

- DataGrid
  - `<asp:DataGrid id=dGrid runat=server/>`
- DataList
  - `<asp:DataList id=dList runat=server/>`
- Repeater
  - `<asp:Repeater id=rep runat=server/>`

Kontrolki list umożliwiają wyświetlanie danych przechowywanych przez kolekcje lub obiekty DataSet i DataReader

Wiązanie danych

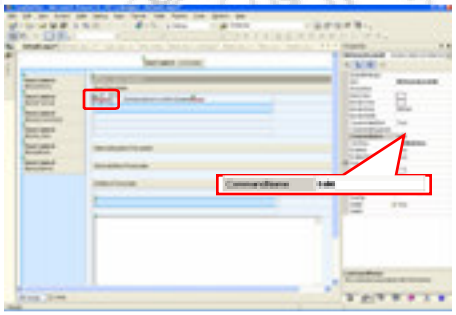
- Aby powiązać dane ze stroną ASP.NET, należy:
  - w miejscu wyświetlania danych wstawić wyrażenie `<%# atrybut lub kolekcja %>`
  - ustawić właściwość DataSource obiektu sterującego
  - wywołać metodę DataBind
- Przykład

```
<asp:Repeater id=„rep” runat=„server”>
<ItemTemplate>
  <%# Container.DataItem(„NazwaKolumny”) %>
</ItemTemplate>
```


Dostęp do danych

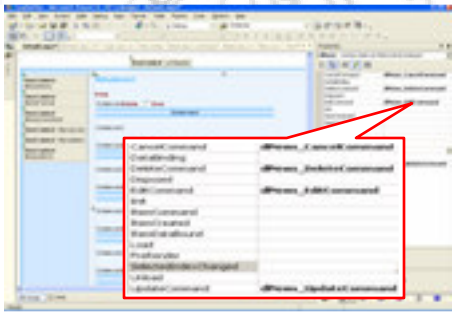
- Aby wyświetlać dane z bazy na stronach ASP.NET należy:
  - Zestawić połączenie z bazą za pomocą komponentów *Connection* i *DataAdapter*
  - Dodać *DataSet* i wypełnić go danymi
  - Wstawić i skonfigurować obiekty do wyświetlania danych (*DataGrid, DataList, Repeater*)
  - Powiązać dane z kontrolkami za pomocą wyrażeń zawartych w `<%# ... %>`
  - Wywołać funkcję *Page.DataBind()*

Kontrolka DataList



7

Kontrolka DataList



8

Kontrolka DataList

```
private void BindDgvw()
{
    SqlConnection dataConnection = new SqlConnection(Global.DefConnectionString);
    SqlDataAdapter dataAdapter = new SqlDataAdapter(SelectCommand, dataConnection);
    DataSet dataSet = new DataSet();

    dataAdapter.Fill(dataSet, CurrentPageIndex*pageSize, pageSize, "Users");
    dlUsers.DataSource = dataSet;
    dlUsers.DataBind();
}
```

```
private void Page_Load(object sender, System.EventArgs e)
{
    if (!IsPostBack)
    {
        BindUsers();
    }
}
```

9

Kontrolka DataList

```
private void BindDgvw()
{
    SqlConnection dataConnection = new SqlConnection(Global.DefConnectionString);
    SqlDataAdapter dataAdapter = new SqlDataAdapter(SelectCommand, dataConnection);
    DataSet dataSet = new DataSet();

    dataAdapter.Fill(dataSet, CurrentPageIndex*pageSize, pageSize, "Users");
    dlUsers.DataSource = dataSet;
    dlUsers.DataBind();
}
```

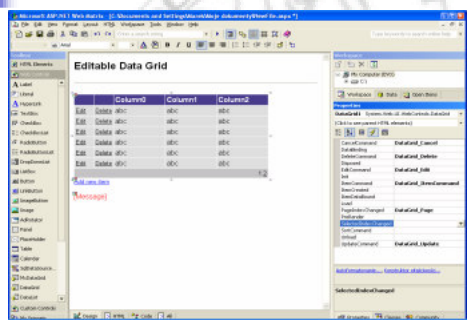
```
if (!IsPostBack)
{
    dlUsers.EditItemIndex = e.Item.ItemIndex;
    BindUsers();
}
```

10

Kontrolka DataList

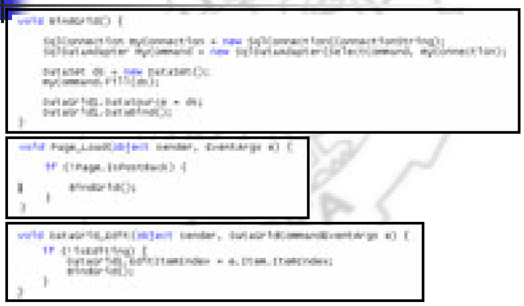
11

Kontrolka DataGrid



12

Kontrolka DataGrid

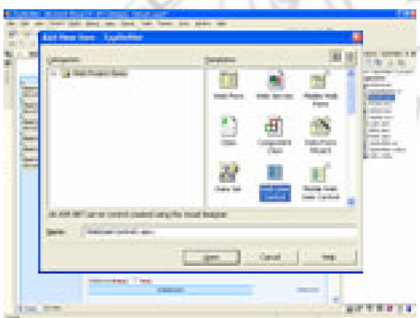


13

Kontrolka DataGrid

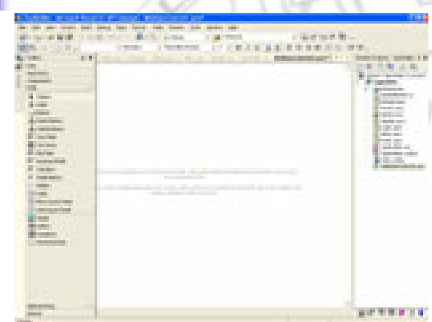
14

Kontrolki użytkownika



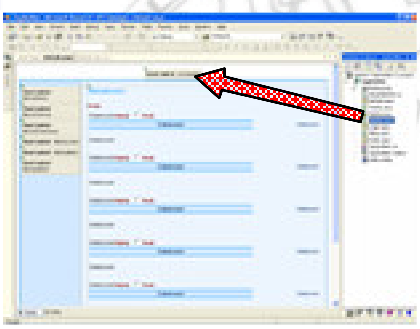
15

Kontrolki użytkownika



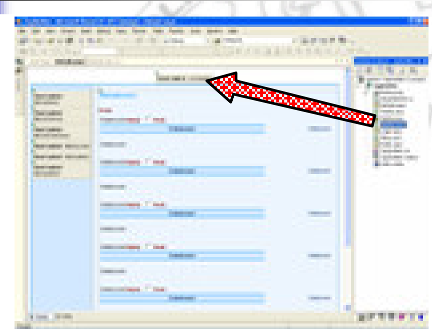
16

Kontrolki użytkownika



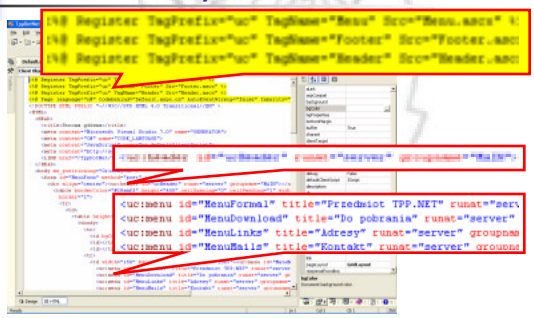
17

Kontrolki użytkownika



18

Kontrolki użytkownika



19

Kontrolki użytkownika

20

Używanie pamięci podręcznej

- Przechowywanie stron w pamięci podręcznej – dyrektywa OutputCache
  - `<%@ OutputCache Duration=„60” VaryByParam=„none” %>`
- Umieszczanie danych w pamięci podręcznej
  - `Cache[„MyDataSet”] = SomeDataSet`

Dzięki pamięci podręcznej można znacznie przyspieszyć działanie aplikacji ASP.NET

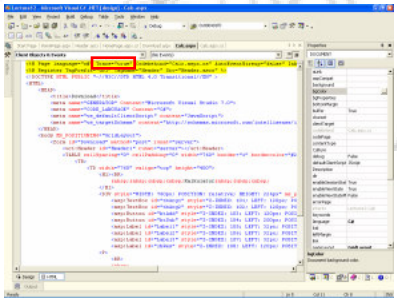
21

Dyrektywa *OutputCache*

- Atrybuty
  - Duration – czas składowania danych
  - Location – miejsce składowania
    - Any – gdziekolwiek
    - None – nie przechowywać
    - Client – na komputerze użytkownika
    - Server – na serwerze
    - Downstream – na serwerze pośredniczącym
  - VaryByParam – składowanie zależne od podanych parametrów
  - VaryByHeader, VarByCustom

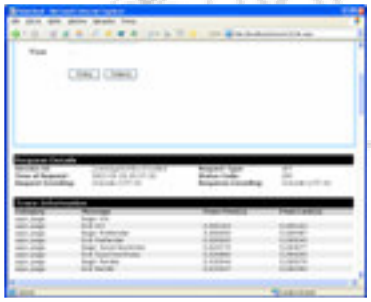
22

Śledzenie stron



23

Śledzenie stron



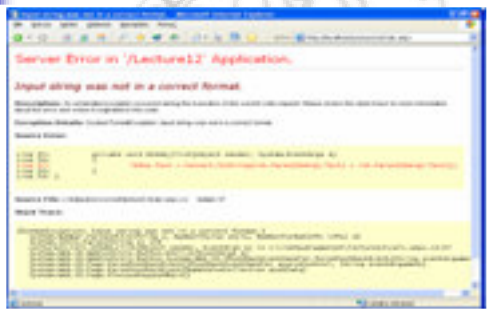
24

Śledzenie stron

- Informacje wyświetlane w trybie śledzenia:
  - *Request Details* – informacje o żądaniu
  - *Trace Information* – przebieg wykonywania
  - *Control Tree* – drzewo kontrolek
  - *Cookies Collection* – informacje o *cookies*
  - *Headers Collection* – informacje zapisane w nagłówkach HTTP
  - *Form Collection* – dane przesłane metodą POST

25

Wyjątki



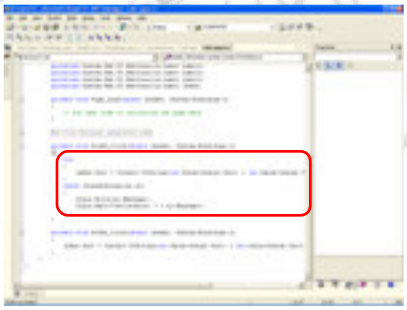
26

Właściwości klasy *Exception*

- *HelpLink* – wskazanie na plik zawierający szczegółowe informacje o błędzie
- *InnerException* – odwołanie do wewnętrznego wyjątku
- *Message* – tekst opisujący błąd
- *Source* – nazwa obiektu, który spowodował błąd
- *StackTrace* – obraz stosu
- *TargetSite* – metoda, w której wystąpił błąd

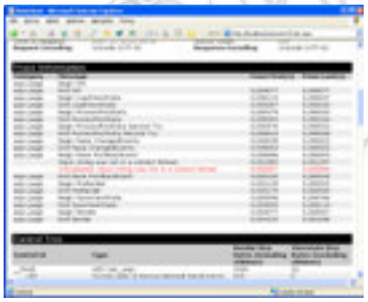
27

Przechwytywanie wyjątków



28

Przechwytywanie wyjątków



29

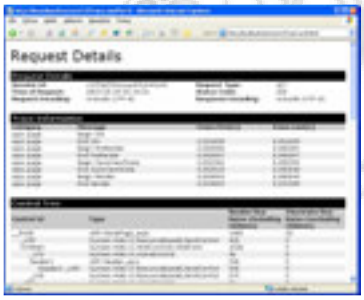
Śledzenie aplikacji

```
<!-- Plik Web.config -->

<configuration>
  <system.web>
    <trace enabled=„true” />
  </system.web>
</configuration>
```

30

Śledzenie aplikacji



31

Obsługa błędów ASP.NET

```
<!-- Plik Web.config -->

<configuration>
  <system.web>
    <customErrors defaultRedirect=„error.htm”
                  mode=„on”>
      <error statusCode=„404”
             redirect=„error404.htm”>
    </customErrors>
  </system.web>
</configuration>
```

32

Obsługa błędów ASP.NET

33

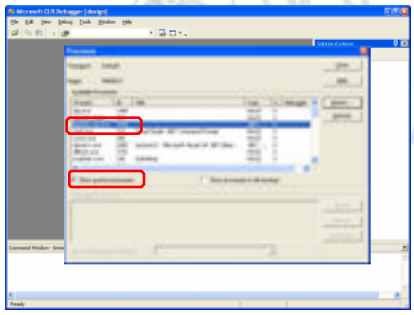
Program uruchomieniowy CLR

```
<!-- Plik Web.config -->

<configuration>
  <system.web>
    <compilation debug=„true” />
  </system.web>
</configuration>
```

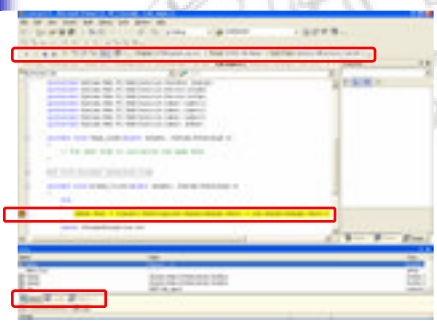
34

Program uruchomieniowy CLR



35

Program uruchomieniowy Visual Studio.NET



36

Sposoby uwierzytelniania

- Uwierzytelnianie systemu Windows
  - Podstawowe
  - Z wykorzystaniem funkcji skrótu
- Uwierzytelnianie z użyciem *Paszportów*
- Uwierzytelnianie za pośrednictwem formularzy

37

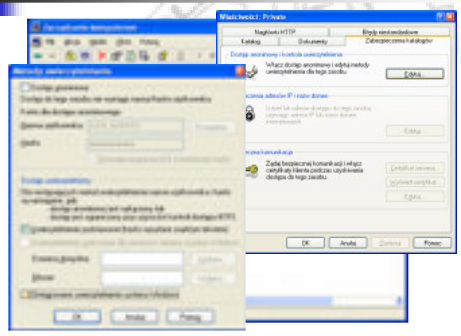
Uwierzytelnianie systemu Windows

```
<!-- Plik Web.config -->

<configuration>
  <system.web>
    <authentication mode=„Windows”>
    </authentication>
  </system.web>
</configuration>
```

38

Uwierzytelnianie systemu Windows



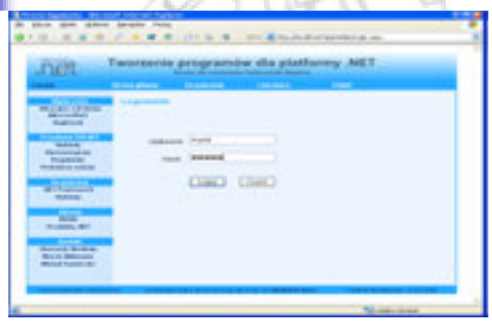
39

Uwierzytelnianie za pośrednictwem formularza

```
<!-- Plik Web.config -->
<configuration>
  <system.web>
    <authentication mode=„Forms”>
      <forms name=„nazwaCiasteczka”
             loginUrl=„adresStronyLogowania”>
        <credentials>
          <user name=„user1” password=„pass1” />
          <user name=„user2” password=„pass2” />
        </credentials>
      </forms>
    </authentication>
  </system.web>
</configuration>
```

40

Uwierzytelnianie za pośrednictwem formularza



41

Uwierzytelnianie za pośrednictwem formularza

42

Autoryzacja

- Autoryzacja dostępu do plików
  - Bazuje na ustawieniach systemu operacyjnego
- Autoryzacja dostępu do adresów URL
  - Zarządzanie kontrolą dostępu za pomocą wpisów w pliku *Web.config*

Autoryzacja

```
<!-- Plik Web.config -->
<configuration>
  <location path=„nazwaPodkatalogu”>
    <system.web>
      <authorization>
        <allow users=„user1,user2,user3”
               roles=„role1,role2,role3” />
        <deny users=„user4,user5,?”
              roles=„role4,role5,role6” />
      </authorization>
    </system.web>
  </location>
</configuration>
```


Personalizacja

```
<!-- Plik Web.config -->

<configuration>
  <system.web>
    <identity impersonate=„true”
              user=„użytkownik”
              password=„hasło” />
  </system.web>
</configuration>
```


Dziękuję za uwagę