

ASP.NET

Veerle Ongenaë

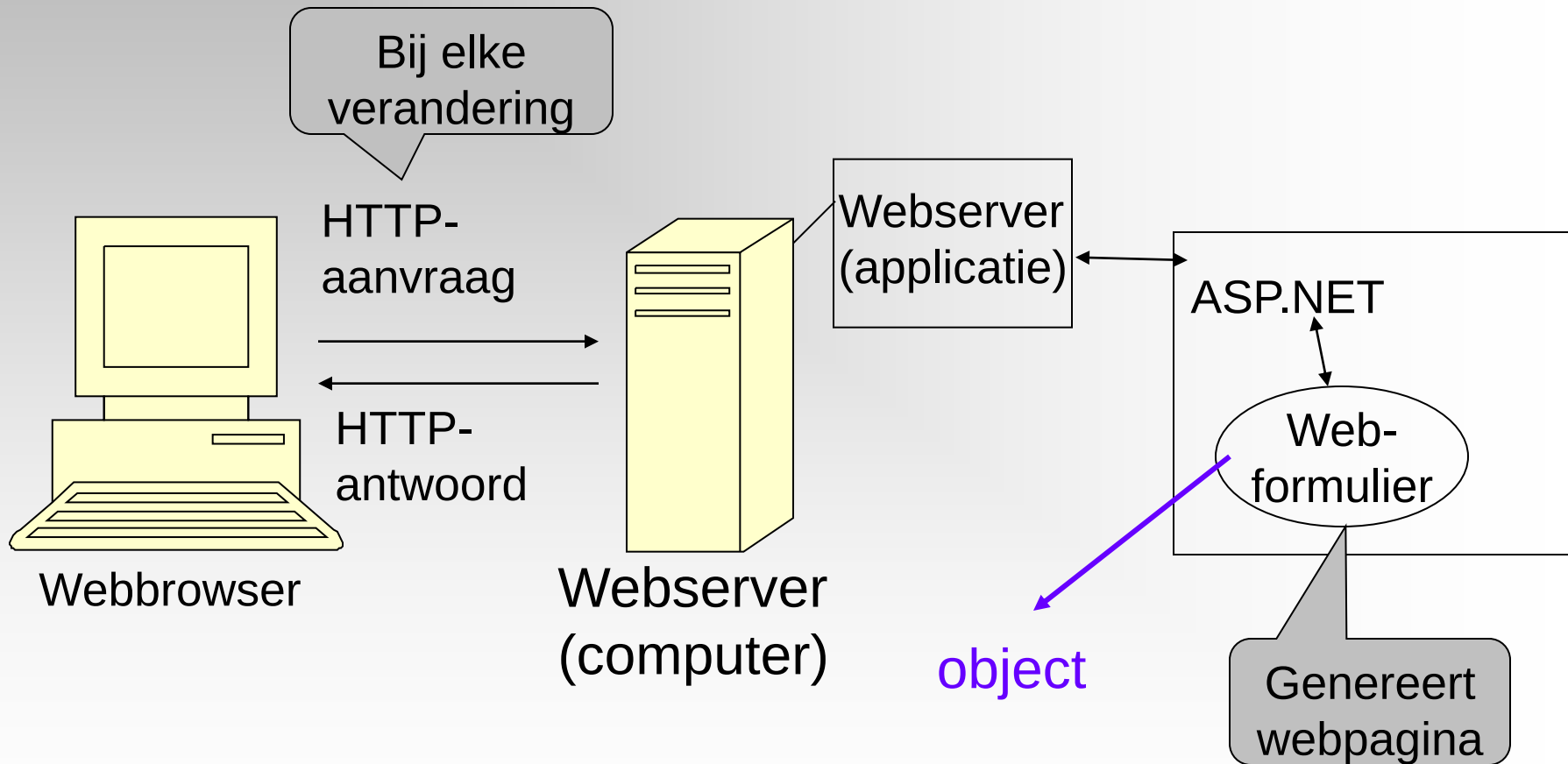
Overzicht

- Wat is ASP.NET?
- Gebeurtenissen
- Servercomponenten
- Statusinformatie bewaren
- Editeerbare tabellen in ASP-pagina's
- Levensloop
- Beveiliging
- Sjablonen en Navigatie

ASP.NET

- Opvolger ASP (Active Server Pages)
- Meer dan ASP
 - Grote webapplicaties
 - Syntax grotendeels compatibel met ASP
 - Veiliger
 - Schaalbaarder
 - Stabieler
 - Gecompileerd ipv geïnterpreteerd → performantie en snelheid stijgen

ASP.NET: dynamische pagina's



ASP.NET

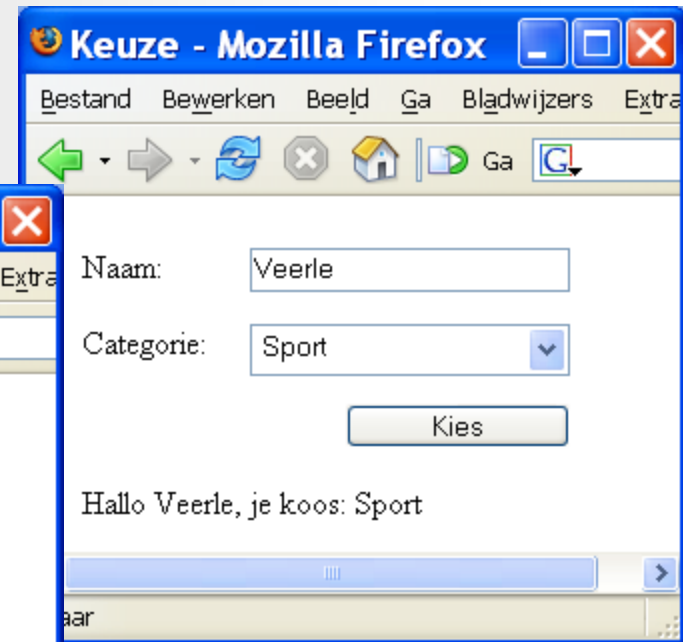
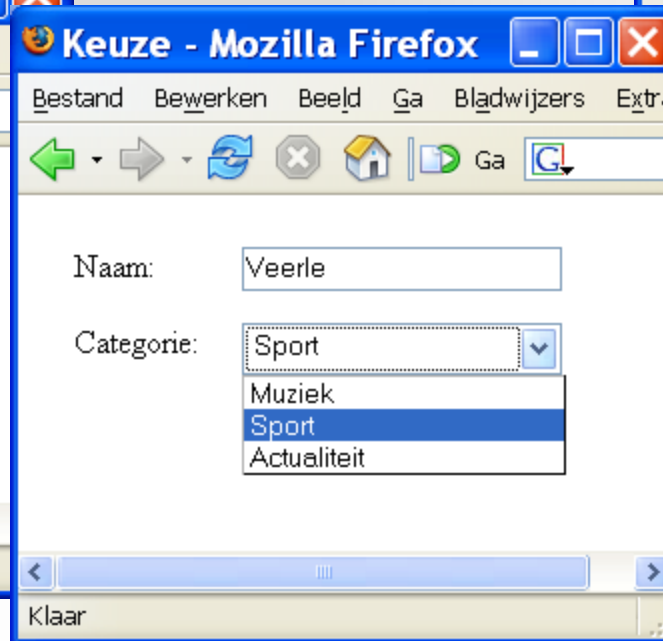
- Webformulieren (Web Forms)
 - Aanmaken webpagina's gebaseerd op formulieren
 - Maakt gebruik van servercomponenten (Server Controls)
 - Webapplicatie maken zoals een GUI-applicatie
- Web Services
 - Mogelijkheid om gebruik te maken van serverfunctionaliteit "vanop afstand"
 - Maakt gebruik van XML en HTTP

Webformulier

- Twee bestanden
 - .aspx-bestand
 - .aspx.cs-bestand
- .aspx-bestand (Grafisch)
 - Layout pagina
- .aspx.cs-bestand (Code Behind)
 - Functionaliteit

Voorbeeld

- Keuze.aspx en Keuze.aspx.cs
- Waarden blijven behouden!



.aspx-bestand

- Bestaat uit
 - Richtlijnen
 - `<%@ Page Language="c#" %>`
 - Stukken HTML
 - Servercomponenten
 - Attribuut: `runat="server"`

Richtlijnen

- Richtlijnen voor de compiler of parser

- Syntax

<%@ NaamRichtlijn LijstAttributen%>

- LijstAttributen

Attr1="Waarde1" Attr2="Waarde2" ... AttrN="WaardeN"

- NaamRichtlijn

- Page

- Attributen: ErrorPage (fout op pagina), Language (programmeertaal), ...

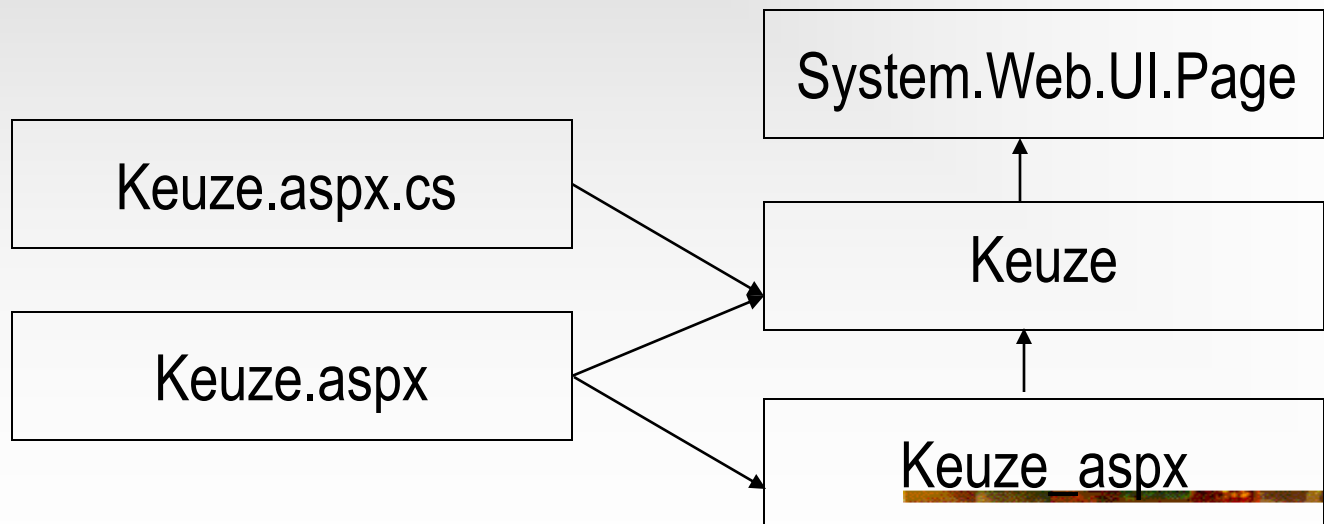
- Import

- Eén attribuut: Namespace

Webformulier

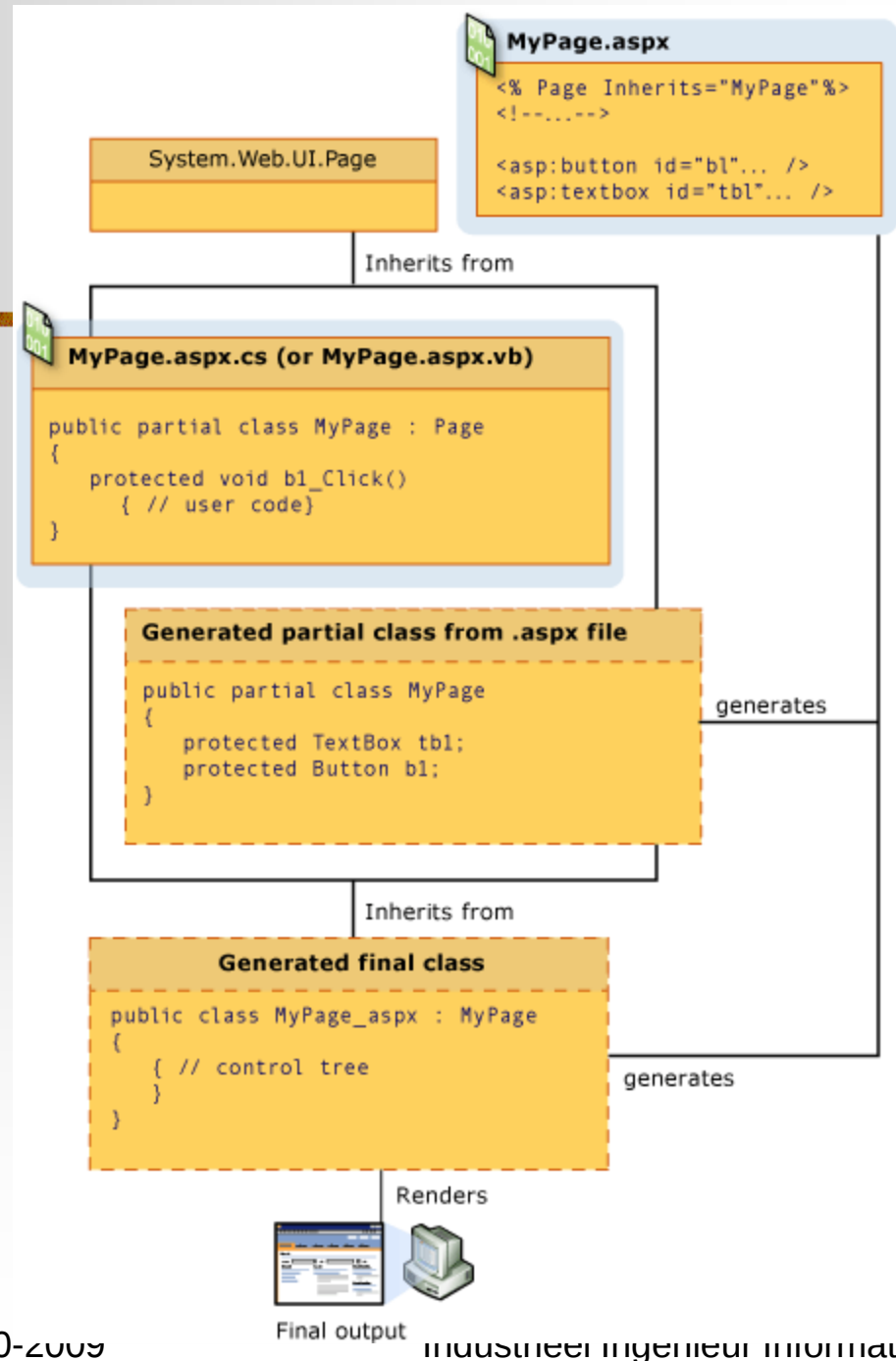
- Deel aspx-bestand (namen componenten) en codebehind (partiële klasse) worden gecombineerd tot één klasse afgeleid van de Page-klasse
- aspx-bestand wordt een klasse afgeleid van deze klasse
- Geladen door de server bij aanvraag

Code Behind



Klassemodel

- Bron:
 - Microsoft ASP.NET help
- Componenten worden niet expliciet vermeld in Code Behind



De klasse Page

- Code Behind is afgeleid van Page
 - Gebruik eigenschappen en methodes
- Logische eigenschap IsPostBack
 - `public bool IsPostBack`
 - Is deze pagina niet voor de eerste keer geladen?

De klasse Page

- Opgeroepen in de methode die de Load-event van het webformulier afhandelt
 - Page_Load
 - Het framework zoekt naar een methode met die naam
 - Wordt automatisch gekoppeld aan de Load-event
 - Eventueel ook mogelijk om zelf te koppeling te maken met een methode met een andere naam
 - Gebruik event-mechanisme
 - Schakel AutoEventWireUp uit

Load-event van Page

■ Event **Load**

- **public event EventHandler Load;**
- Wordt uitgevoerd bij het laden (één van de stadia van de levensloop van een Page-object)
- EventHandler-objecten toevoegen of verwijderen
 - **public delegate void EventHandler**
(object *sender*, EventArgs e);

Klikken op knop

- Methode in Code Behind

- Signatuur van een event

protected void **ToonKeuze**(object sender, System.EventArgs e)

- Koppelen aan Click-event

- Attribuuat OnClick

<asp:Button id="KiesKnop" runat="server" onclick="ToonKeuze"></asp:Button>

- Via eigenschappen-venster
 - In de methode Page_Init

KiesKnop.Click += new EventHandler(this.ToonKeuze);

Overzicht

- Wat is ASP.NET?
- Gebeurtenissen
- Servercomponenten
- Statusinformatie bewaren
- Editeerbare tabellen in ASP-pagina's
- Levensloop
- Beveiliging
- Sjablonen en Navigatie

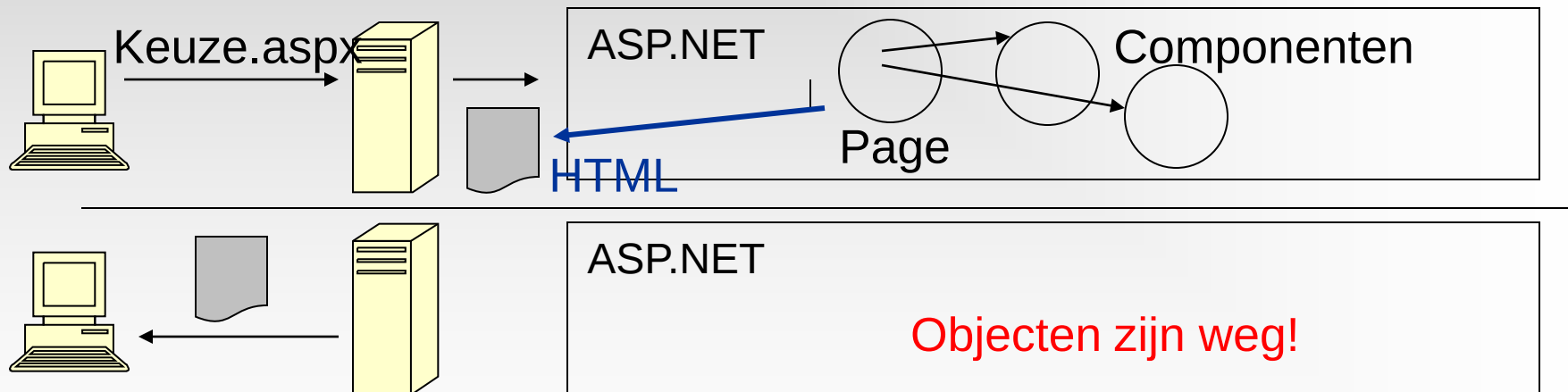
ASP.NET event model

■ Event driven model

- Webformulier bestaat uit webcomponenten
 - Bv. Button
- Een component heeft één of meerdere events
 - Bv. Button → Click
- Aan een event kan je een eventhandler toevoegen
 - Bv. Click → ToonKeuze
- Een eventhandler is een methode die uitgevoerd wordt als de event optreedt
 - Bv. ToonKeuze
 - Signatuur eventhandler bepaald door delegate-klasse
EventHandler

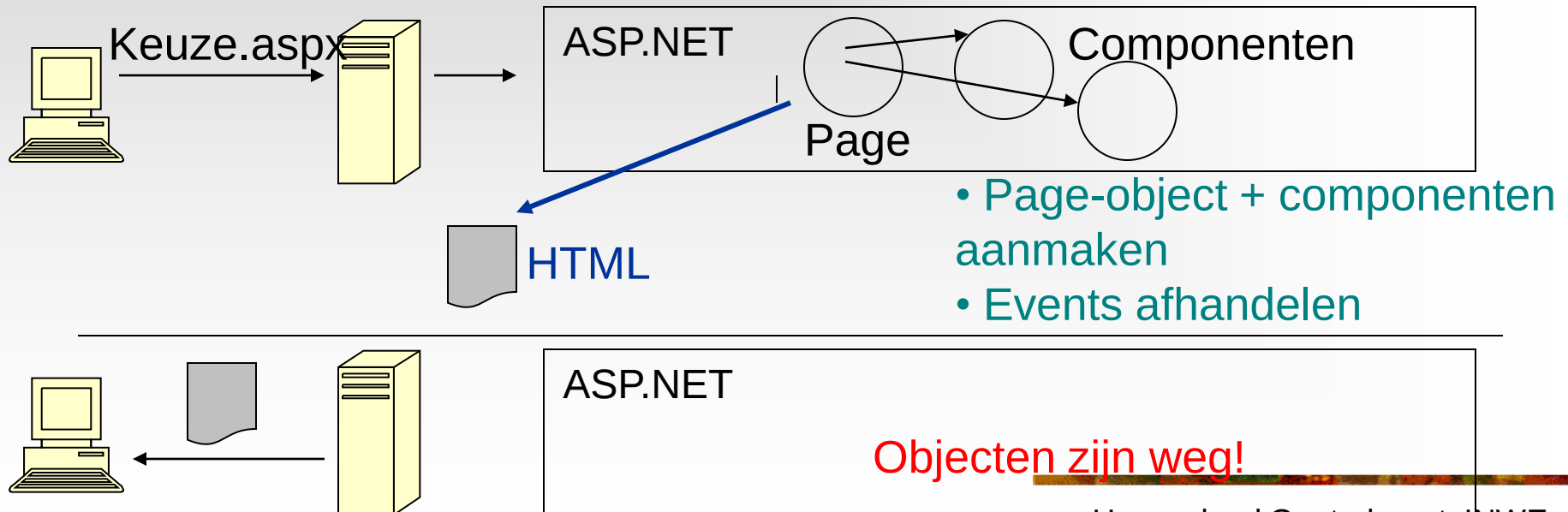
ASP.NET event model: realisatie

- Gebruiker roept voor de eerste maal de webpagina op



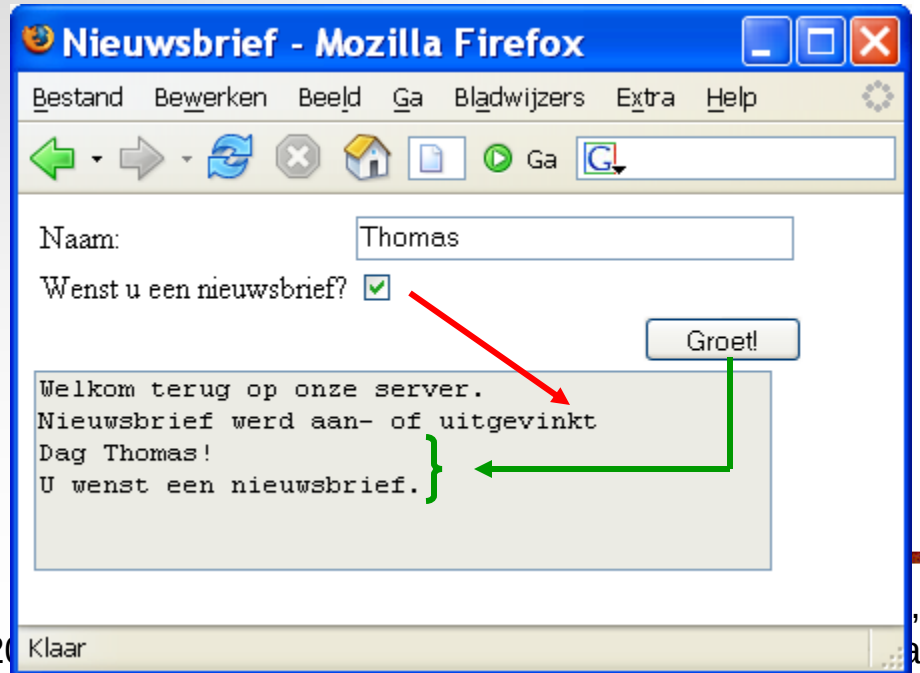
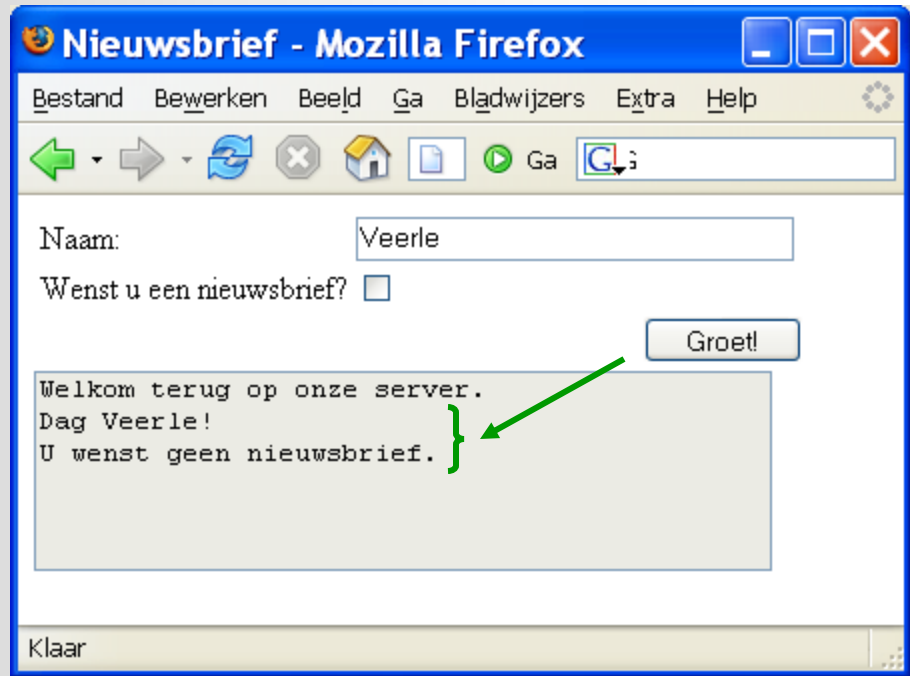
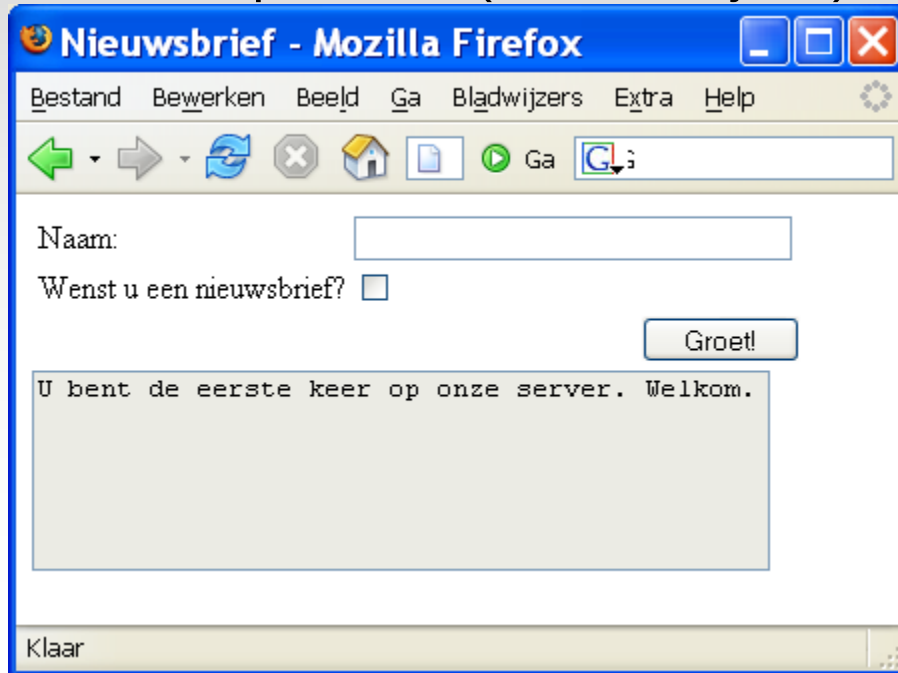
ASP.NET event model: realisatie

- Gebruiker vult gegevens in en drukt op de knop
 - Er wordt opnieuw een HTTP-aanvraag gestuurd (postback naar Keuze.aspx)



Voorbeeld Events

- Nieuwsbrief (.aspx+.aspx.cs)
- Events
 - Pagina laden (1e lijn)
 - Aanvinken (eventueel 2e lijn)
 - Knop klikken (2 laatste lijnen)



Voorbeeld Events

- Events waarop onze applicatie reageert
 - Page
 - Load
 - CheckBox
 - CheckedChanged
 - Button
 - Click

Events afhandelen

■ EventHandler

- Methodes die events afhandelen hebben allemaal dezelfde signatuur

protected **void** Page_Load(object sender, EventArgs e)

protected **void** GroetKnop_Click(object sender, EventArgs e)

protected **void** KeuzeNieuwsbrief_CheckedChanged(object sender, EventArgs e)

- object: component waar de event optrad
- EventArgs: informatie over event

De klasse CheckBox



- Logische eigenschap **Checked**
 - Is het vakje aangevinkt?
- Het aanvinken veroorzaakt geen postback naar server
 - Indien nodig de logische eigenschap **AutoPostBack** op **true** instellen
 - Let op: mogelijk te veel heen en weer trips naar de server
 - Onmiddellijk reageren op aanvinken

Overzicht

- Wat is ASP.NET?
- Gebeurtenissen
- Servercomponenten
- Statusinformatie bewaren
- Editeerbare tabellen in ASP-pagina's
- Levensloop
- Beveiliging
- Sjablonen en Navigatie

Servercomponenten

- **HTML-elementen** met een attribuut `runat="server"` → HTML-servercomponenten
- Tags waarvan de naam begint met **asp:** → webservercomponenten
 - Hebben ID-attribuut → naam variabele code behind
 - Voorbeelden
 - Label
 - TextBox
 - Button
 - CheckBox
 - DropDownList

Servercomponenten

- Uitgevoerd op de server (steeds attribuut **runat="server"**)
- Gebruikersinterface
- Functionaliteit
- Herbruikbare componenten
- Bewaren client-informatie in verborgen veld
- Voorbeeld: tekstveld, label, ...
- Attribuut id: naam object (=variabele in code behind)

Servercomponenten: types

- HTML-servercomponenten (HTML server controls)
- Web-servercomponenten (Web server controls)
- Gebruikerscomponenten (User Controls)
- Custom controls
- Controlecomponenten (Validation Controls)

Html-servercomponenten

- HTML-tags met een attribuut **runat="server"**
 - De HTML-tag wordt niet naar de uitvoer geschreven
 - Een corresponderende object genereert uitvoer
- Namespace: **System.Web.UI.HtmlControls**
- Met HTML-tag correspondeert een klasse in **System.Web.UI.HtmlControls**
 - Attribuut ↔ eigenschap

form-tag

- runat="server"
- Geen action-attribuut
 - "post back" naar zelfde pagina
- Genereert een verborgen veld
 - __VIEWSTATE
 - Inhoud componenten bewaren

Servercomponenten: types

- HTML-servercomponenten (HTML server controls)
- Web-servercomponenten (Web server controls)
- Gebruikerscomponenten (User Controls)
- Custom controls
- Controlecomponenten (Validation Controls)

Web-servercomponenten

- Tags zijn van de volgende vorm

<asp:naamComponent runat="server"

</asp:naamComponent>

<asp:naamComponent runat="server" />

- Namespace **System.Web.UI.WebControls**

- Meer tags dan enkel corresponderende HTML-tags

- Overzicht:

<http://quickstarts.asp.net/QuickStartv20/aspnet/Default.aspx>

Web-servercomponenten

- Afgeleid van `System.Web.UI.Control`
 - Basiseigenschappen van alle controls
- Ook Page is afgeleid van `System.Web.UI.Control`

Web-servercomponenten

■ Label

- Eigenschap Text

■ TextBox

- Eigenschap Text
- Eigenschap TextMode
 - Één of meerdere lijnen, wachtwoord
- Eigenschap ReadOnly

■ Button

- Event Click
- Eigenschap Text

■ CheckBox

- Eigenschap AutoPostBack
- Eigenschap Checked
- Event CheckedChanged

ListControl

- ListControl
 - Basisklasse lijsten
- Afgeleide klassen
 - CheckBoxList
 - DropDownList
 - ListBox
 - RadioButtonList
- Lijsten
(.aspx+.aspx.cs)

Lijsten - Microsoft Internet Explorer

Bestand Bewerken Beeld Favorieten Extra Help

Vorige Zoeken Favorieten

Adres <http://localhost/vbn/Lijsten2.aspx> Ga naar Koppelingen

Land: Nederland

Categorie: Sport
Muziek
Actualiteit
Geschiedenis

Maaltijden: M V

☐ middagmaal
☐ ontbijt
☐ avondmaal
☐ vieruurtje
☐ tienuurtje

Gereed Lokaal intranet

Toon Events

```
Lijst KeuzeLand index 1 item Nederland waarde NL  
Lijst MVKeuze index 0 item M waarde M  
Lijst KeuzeCategorie item Muziek waarde Muziek  
Lijst KeuzeEten item tienuurtje waarde 10U  
Lijst KeuzeEten item middagmaal waarde MID
```

Overzicht

Land

- + Land (string code, string naam)
- + string Naam
- + string Code
- + string ToString()

GegevensFabriek

- + GegevensFabriek()
- + string[] GetCategorieen()
- + Dictionary<string,string> GetMaaltijden()
- + Dictionary<string,Land> GetLanden()

Dictionary<Tkey,Tvalue>

- Uit de namespace
System.Collections.Generic
- Bestaat uit sleutel/waarde-paren (type: KeyValuePair)
 - Type sleutel: Tkey
 - Type waarde: Tvalue
- Methode **public void Add(Tkey sleutel, Tvalue waarde)**

Principes gegevens binden

- Verwijzing maken tussen een servercomponent (bv. ListControl) en gegevensbron (DataSource)
 - Waarde geven aan de eigenschap DataSource
- Servercomponent wordt pas later opgevuld (expliciet oproepen methode DataBind)
 - DataBind
 - Eerste keer
 - Bij verandering
 - DataBind op component → DataBind op kindercomponenten

Voorwaarde DataSource

- Implementeert de interface IEnumerable
→ overlopen verschillende items
 - Methode
 - `IEnumerator GetEnumerator()` ;
- Mogelijke types van gegevensbronnen:
Array, ArrayList, Hashtable, DataView,
DataReader, ...

ListControl

■ Methodes ivm opvullen lijst

- `public object DataSource {get; set;}`
 - Link naar gegevens
- `public void DataBind();`
 - Lijst opvullen
- `public string DataTextField {get; set;}`
 - Naam eigenschap die getoond wordt
 - Bv. Dictionary → IEnumerator loopt over KeyValuePair → heeft eigenschappen Key en Value
- `public string DataValueField {get; set;}`
 - Naam eigenschap die de waarde voorstelt
 - Waarde = parameter HTML-formulier

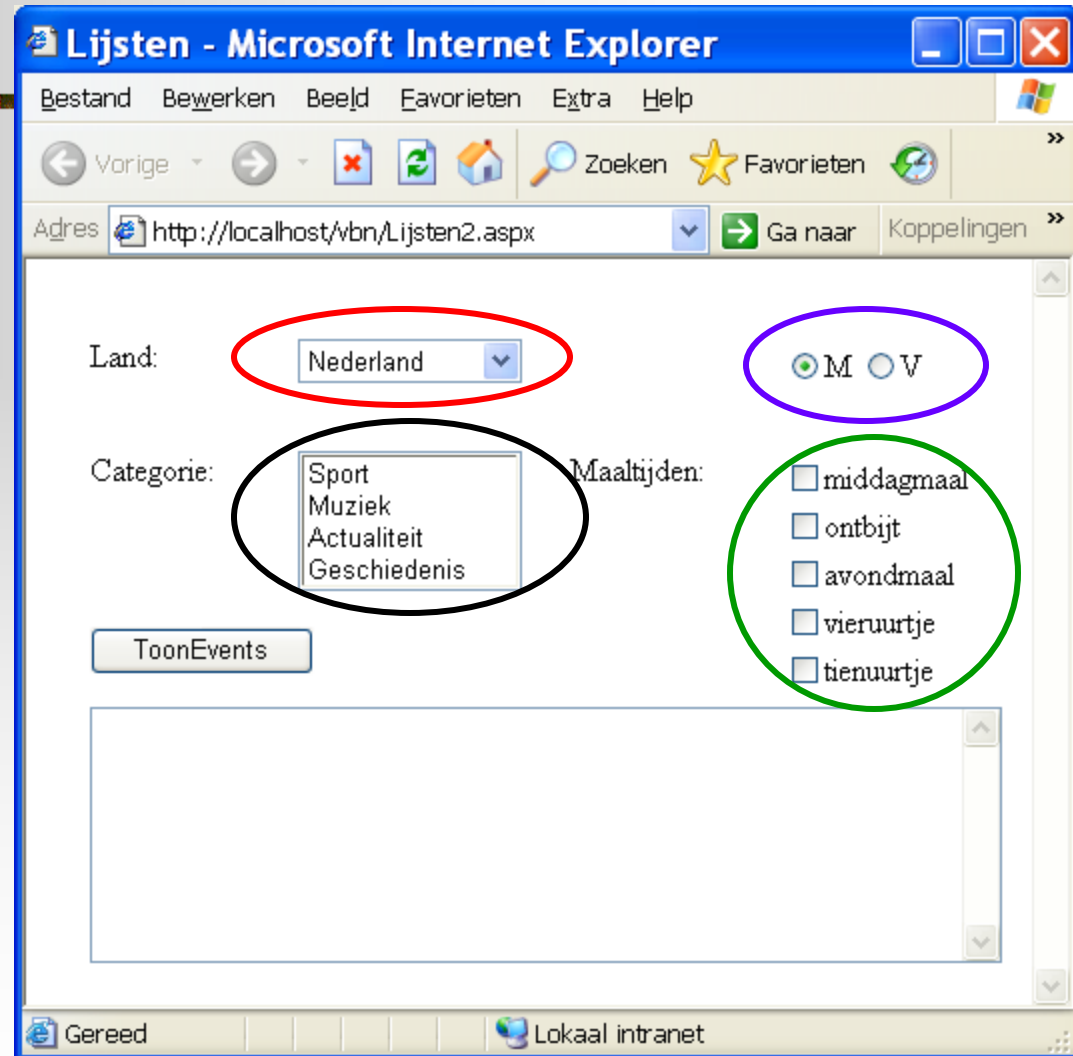
ListControl

- Methodes ivm selecteren lijstelementen
 - `public ListItemCollection Items {get;}`
 - `public int SelectedIndex {get; set;}`
 - `public ListItem SelectedItem {get;}`
 - Eigenschappen ListItem
 - `public bool Selected {get; set;}`
 - `public string Text {get; set;}`
 - `public string Value {get; set;}`
 - `public string SelectedValue {get; set;}`
 - `public event EventHandler
SelectedIndexChanged;`
 - `public bool AutoPostBack {get; set;}`

ListControl

■ Afgeleide klassen

- ListBox
- CheckBoxList
- RadioButtonList
- DropDownList



ListBox

- Instellen selectiemode
 - `public ListSelectionMode`
`SelectionMode {get; set;}`
- `ListSelectionMode (enum)`
 - `Single`
 - `Multiple`

CheckBoxList

- Aantal zichtbare kolommen

- `public int RepeatColumns {get; set;}`

- Horizontale of verticale lijst

- `public RepeatDirection RepeatDirection {get; set;}`

- `RepeatDirection (enum)`

- `Horizontal`

- `Vertical`

CheckBoxList

■ Layout bepalen

- `public virtual RepeatLayout RepeatLayout {get; set;}`

- `RepeatLayout(enum)`

- `Flow`

- `Met <br> ...`

- `Table`

- `In een HTML-tabel`

RadioButtonList

- Analooog CheckBoxList

Vershil Lijsten.aspx en Lijsten2.aspx

- Bij veranderen selectie worden onmiddellijk de events getoond
- Eigenschap **AutoPostBack** heeft de waarde True

Lijsten - Mozilla Firefox

Bestand Bewerken Beeld Ga Bladwijzers Extra Help

http://localhost/vbn/L

Land: Frankrijk

Categorie: Sport
Muziek
Actualiteit
Geschiedenis

Maaltijden: ☐ M ☐ V
☐ avondmaal
☒ middagmaal
☒ vieruurtje
☐ tienuurtje
☐ ontbijt

ToonEvents

Lijst KeuzeCategorie item Sport waarde Sport
Lijst KeuzeEten item middagmaal waarde MID
Lijst KeuzeEten item vieruurtje waarde 16U

Klaar

Web-servercomponenten

- Label
- TextBox
- Button
- CheckBox
- ListControl
 - DropDownList
 - ListBox
 - CheckBoxList
 - RadioButtonList

Web-servercomponenten

■ Data Web Server Controls

- Componenten die “veel” gegevens tonen, meestal in tabelvorm
- GridView
 - Tonen van tabellen en rijgeoriënteerde gegevens
- DataList
 - Meer invloed op de layout van de tabel

DataTable tonen in GridView



The screenshot shows a web browser window with the title 'Overzicht Bestelli...'. The menu bar includes 'Bestand', 'Bewerken', 'Beeld', 'Ga', and 'Bladwijzers'. The toolbar contains navigation icons (back, forward, refresh, stop, home, search) and a search box with the text 'Ga' and a magnifying glass icon. The main content area displays a table with 3 columns: 'BestelID', 'Hoeveelheid', and 'Firma'. The table contains 4 rows of data. At the bottom of the window, there is a status bar with the text 'Klaar'.

BestelID	Hoeveelheid	Firma
1	15	Mijn Firma
2	23	Firma om de hoek
3	67	Ik
4	12	Mijn Firma

- Bestellingen
(.aspx+.aspx.cs)

DataTable tonen in GridView

- In code (in methode Page_Load)

```
if (!IsPostBack) {  
    GegevensFabriek data = new GegevensFabriek();  
    OverzichtBestellingen.DataSource =  
        data.GetBestellingen(); —————→ DataTable  
    OverzichtBestellingen.DataBind();  
    }  
    ↘ GridView
```

- Kan ook met een DataSet

Principes gegevens binden

- Verwijzing maken tussen een servercomponent (bv. GridView) en gegevensbron (DataSource)
 - Waarde geven aan de eigenschap DataSource
- Servercomponent wordt pas later opgevuld (oproepen methode DataBind)
 - DataBind
 - Eerste keer
 - Bij verandering
 - DataBind op component → DataBind op kindercomponenten

Voorwaarde DataSource

- Implementeert de interface IEnumerable
→ overlopen verschillende items
 - Methode
 - `IEnumerator GetEnumerator()` ;
- Mogelijke types van gegevensbronnen:
Array, ArrayList, Hashtable, DataView,
DataReader, DataSet, DataTable, ...

Gegevens binden

■ GridView

- Gegevens voorstellen in tabelvorm
- Opvolger **DataGrid**
- Namespace:
System.Web.UI.WebControls
- Eigenschap: **object DataSource** (interface **System.Collections.IEnumerable** implementeren)
- Methode: **void DataBind()**

ListView

- Opvolger DataList
 - .NET 3.5
- Tonen van gegevens in tabelvorm
- Veel flexibeler layout dan een GridView
 - Layout wordt beschreven met templates
- Voorbeeld: BestellingenLijstView.aspx



ListView

■ Templates

■ LayoutTemplate

- Basislayout
- Beschrijving component die items zal bevatten (vb. table)
- Bevat object (placeholder) dat vervangen wordt voor elk item (vb. tr)

■ ItemTemplate

- Layout element in lijst

■ ItemSeparatorTemplate

- Scheiding tussen de elementen

■ ...

- GroupTemplate, GroupSeparatorTemplate, EmptyItemTemplate, EmptyDataTemplate, SelectedItemTemplate, AlternatingItemTemplate, EditItemTemplate, InsertItemTemplate

Te binden gegevens

- Beschrijven layout van een willekeurig element uit de lijst
 - Element is nog niet gekend
 - Speciale syntax om aan te duiden dat de waarde later nog ingevuld moet worden (tussen `<%#` en `%>`)
 - De gegevens worden later gebonden
 - Pas uitvoeren bij **DataBind()** van component of pagina

Gegevens binden API

■ **DataBinder**

- Namespace: **System.Web.UI**

- Statische methode:

object Eval(object, string);

- Vermijdt casten

DataBinder.Eval(Container.DataItem, “voornaam”)

- ipv

((DataListItem)Container.DataItem)[“voornaam”]

■ **Container.DataItem**

- Huidige element in lijst
- Container: eerste bovenliggende container

Servercomponenten: types

- HTML-servercomponenten (HTML server controls)
- Web-servercomponenten (Web server controls)
- Gebruikerscomponenten (User Controls)
- Custom controls
- Controlecomponenten (Validation Controls)

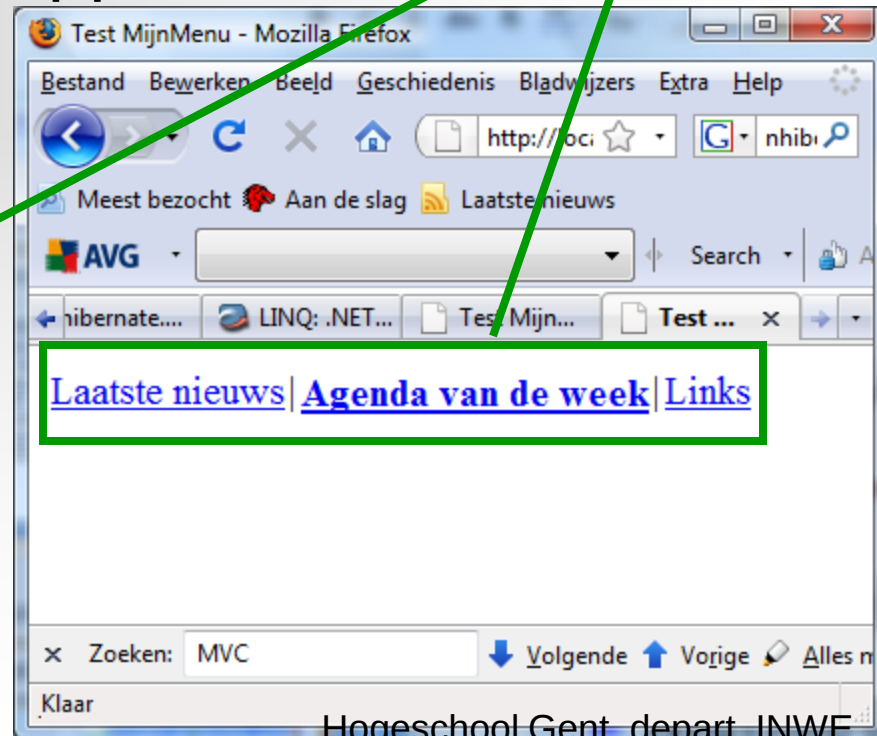
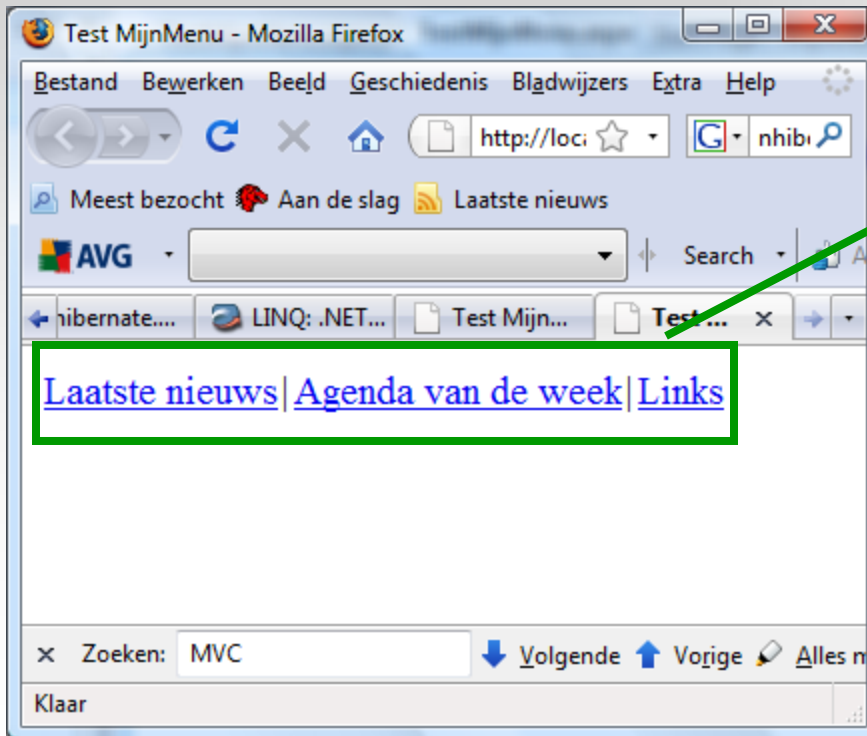
Zelf controls maken

- Visual Studio: basispakket controls
- Nood aan meer?
 - Zelfgemaakte controls
 - Web users controls
 - Vergelijkbaar met webformulier (web form)
 - Visueel opbouwen
 - Code behind
 - Web custom controls
 - Nieuwe te programmeren klasse → bibliotheek (DLL)
 - Kunnen opgenomen worden in de toolbox van Visual Studio

Web user control

- Voorbeeld TestMijnMenu.aspx
 - Menubalk van een webapplicatie

User control:
MijnMenu.ascx



Web user control

- Bestaat uit twee bestanden

- .ascx-bestand

- Layout

- Eerste lijn (richtlijn)

```
<%@ Control Language="C#" AutoEventWireup="true"  
CodeFile="MijnMenu.ascx.cs" Inherits="MijnMenu" %>
```

- .ascx.cs-bestand

- Functionaliteit

- Verband analoog zoals bij webformulier

Gebruik web user control

■ Gebruik user control

```
<%@ Register TagPrefix="uc1" TagName="Menu" Src="Menu.ascx" %>  
<uc1:MijnMenu id="MenuBalk" runat="server"></uc1:Menu>
```

Web user control

- Afgeleide klasse van `System.Web.UI.UserControl`
 - Eigenschappen
 - `bool IsPostBack`
 - Event
 - `EventHandler Load`

Custom controls

- Maken deel uit van een (zelf te schrijven) bibliotheek
- Kan je toevoegen aan “toolbox”

User controls ↔ custom controls

User control	Custom control
Eenvoudig	Moeilijk
Voor elke applicatie een kopie	Eén kopie voor verschillende applicaties
Geen deel van toolbox	Kan toegevoegd worden aan toolbox
Statische layout	Dynamische layout

Servercomponenten: types

- HTML-servercomponenten (HTML server controls)
- Web-servercomponenten (Web server controls)
- Gebruikerscomponenten (User Controls)
- Custom controls
- Controlecomponenten (Validation Controls)

Controlecomponenten

- Controleren van gebruikersinput
 - Controle waarde
 - Controle patroon
 - Is ingevuld?
- Tonen van foutboodschappen bij verkeerde invoer

Verschillende soorten

- RequiredFieldValidator
 - Waarde ingevuld?
- RangeValidator
 - Waarde in interval?
- CompareValidator
 - Waarden van twee controls vergelijken (<,>==,...)
- RegularExpressionValidator
 - Voldoet waarde aan een “regular expression”?
- CustomValidator
 - Zelf uit te voeren controle schrijven

Voorbeeld controle gegevens

■ KeuzeMetControle.aspx

Keuze - Microsoft Internet Explorer

Bestand Bewerken Beeld Favorieten Extra Help

Vorige Zoeken Favorieten

Adres <http://localhost/vbn/KeuzeMetControle.aspx> Ga naar Koppelingen

Naam:

Categorie: <!--Kies Categorie-->

Gelieve uw naam in te vullen

Gelieve een categorie te kiezen

Gereed Lokaal intranet

Controlecomponenten

- Alle controlecomponenten zijn afgeleide klassen van `System.Web.UI.WebControls.BaseValidator`
- Namespace: **`System.Web.UI.WebControls`**

Attributen/Eigenschappen

- **ControlToValidate**: naam van component die gecontroleerd moet worden
- **Display**: vaste plaats in pagina (**Static**) of aanpassen layout bij fout (**Dynamic**) of niet getoond (**None**)
- **ErrorMessage**: foutboodschap
- **EnableClientScript**: inschakelen validatie door client (=browser)

Principe controles

- Controles gebeuren indien mogelijk aan de “client side” (d.w.z. als de browser dit ondersteunt)
 - De component genereert daarvoor de nodige javascript
- Er is steeds controle op de server
- Controle wordt uitgevoerd bv. bij klikken knop
 - Eigenschap CausesValidation → ev. aanpassen
- Uitvoeren van alle controles op een pagina:
 - De logische eigenschap IsValid van Page

RequiredFieldValidator

- Namespace:
System.Web.UI.WebControls
- Extra attribuut
 - **InitialValue**: beginwaarde, mag niet de waarde bij controle zijn
 - Niet opgegeven → lege string
- Bij controle worden spaties voor en achteraan verwijderd

RangeValidator

- Invoer component vergelijken met twee grenswaarden
- Extra attributen
 - **MinimumValue**: ondergrens
 - **MaximumValue**: bovengrens
 - **Type**: datatype waarden
 - Waarden component worden naar dit type geconverteerd
 - Lege invoer → validatie gelukt !!
- Vaak gebruikt in combinatie met CompareValidator
 - Type invoer controleren

CompareValidator

- Invoer component vergelijken met invoer andere component of een constante
- Extra attributen
 - **ControlToCompare**: andere component
 - **ValueToCompare**: constante
 - **ValidationCompareOperator Operator**: type vergelijking
 - Equal, GreaterThan, ...
 - DataTypeCheck → controleert ALLEEN datatype
- Opm. invoer leeg → validatie gelukt!!

RegularExpressionValidator

- Controleren of invoer component voldoet aan een bepaald patroon bv. Datum, telefoonnummer, ...
- Extra attributen
 - **ValidationExpression**: reguliere uitdrukking
 - Syntax JScript
- Opm. invoer leeg → validatie gelukt!!

Voorbeeld Controls.aspx

Controls - Microsoft Internet Explorer

Bestand Bewerken Beeld Favorieten Extra Help

Vorige Zoeken Favorieten

Adres <http://localhost/vbn/Controls.aspx> Ga naar Koppelingen

Kleinste even getal tussen 0 en 1000

Grootste even getal tussen 0 en 1000

emailadres:

ControleerInvoer

Gereed Lokaal intranet

CustomValidator nodig

Kleinste even getal tussen 0 en 1000

Grootste even getal tussen 0 en 1000

emailadres:

Dit is geen mailadres

ControleerInvoer

- Grootste getal ligt niet tussen 0 en 1000
- Kleinste getal ligt niet tussen 0 en 1000
- Grootste getal is niet groter dan kleinste
- Dit is geen mailadres

ValidationSummary

Gereed Lokaal intranet

CustomValidator

- Functionaliteit controle is zelf te programmeren
- Validatie op server
 - Event **ServerValidate** van **CustomValidator**
 - Delegate toevoegen
- Validatie op client
 - Script toevoegen aan het aspx-bestand
 - Eigenschap **ClientValidationFunction**
 - Verwijzen naar scriptfunctie

Signatuur validatiemethodes

- Signatuur

- Delegate
- Scriptfunctie

void Methode(object,
System.Web.UI.WebControls.ServerValidateEventArgs)

- Eigenschappen **ServerValidateEventArgs**

- **Value**
 - Waarde invoer
- **IsValid**
 - Controle OK?

ValidationSummary

- Overzicht van alle fouten op pagina.
- Lijst van foutboodschappen
 - Foutboodschappen van de controlecomponenten
- Controlecomponent kan eigenschap **Display** op **None** zetten

Overzicht

- Wat is ASP.NET?
- Gebeurtenissen
- Servercomponenten
- Statusinformatie bewaren
- Editeerbare tabellen in ASP-pagina's
- Levensloop
- Beveiliging
- Sjablonen en Navigatie

Statusinformatie bewaren

- HTTP is een statusloos protocol
 - Elke aanvraag staat “op zich”
 - Geen verband tussen twee aanvragen van een zelfde client
- Toch nodig
 - Vb. winkelwagentje
- Hoe informatie bewaren tussen twee HTTP-aanvragen van een zelfde client

Statusinformatie bewaren

- Bewaren informatie over verschillende aanvragen heen
 - Op de server
 - Sessies
 - Application State
 - Profielen (Profile properties)
 - Via de client
 - Cookies, Verborgene formulier velden, QueryString van een URL, ...
 - View State
 - Controle State: statusinformatie voor een component

Opvolging van sessies (Session Tracking)

- HTTP: geen statusinformatie
- Informatie over een aantal requests van één gebruiker (=browser) bewaren gedurende een bepaalde tijd
- Webapplicatie met verschillende ASP.NETs: gebruikersinformatie delen
 - Bestellen boeken
 - Boeken toevoegen aan winkelwagentje
 - Tonen winkelwagentje

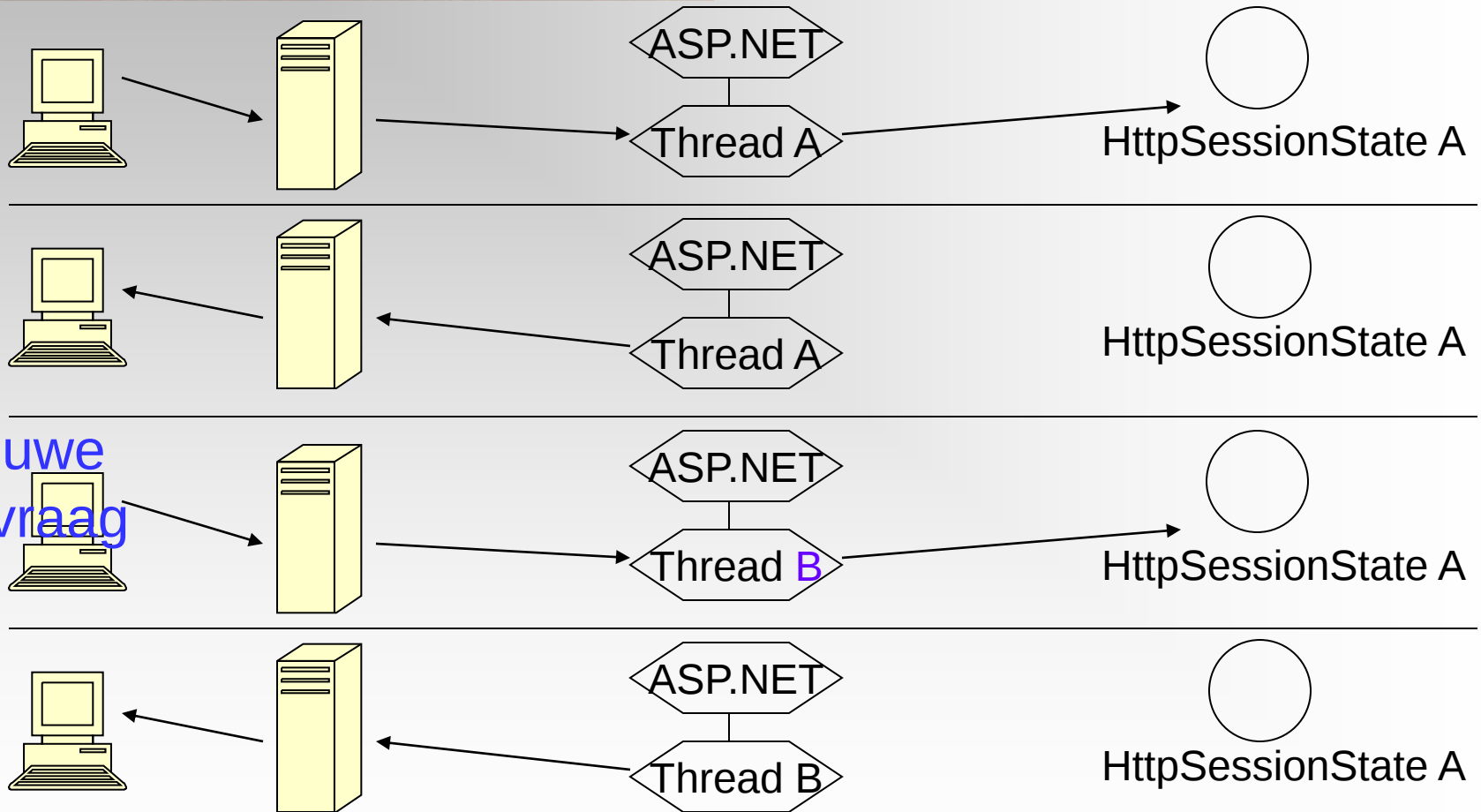
Principe sessies

- Uitwisselen kenteken tussen client en server
 - Het HTTP-antwoord van de eerste aanvraag krijgt een uniek kenteken mee
 - Bij elke volgende aanvraag wordt dit kenteken meegegeven
 - Kenteken is verbonden met een sessie-object op de server
- Mogelijke implementaties
 - Cookies
 - URL herschrijven
 - Verborgene formulierelementen

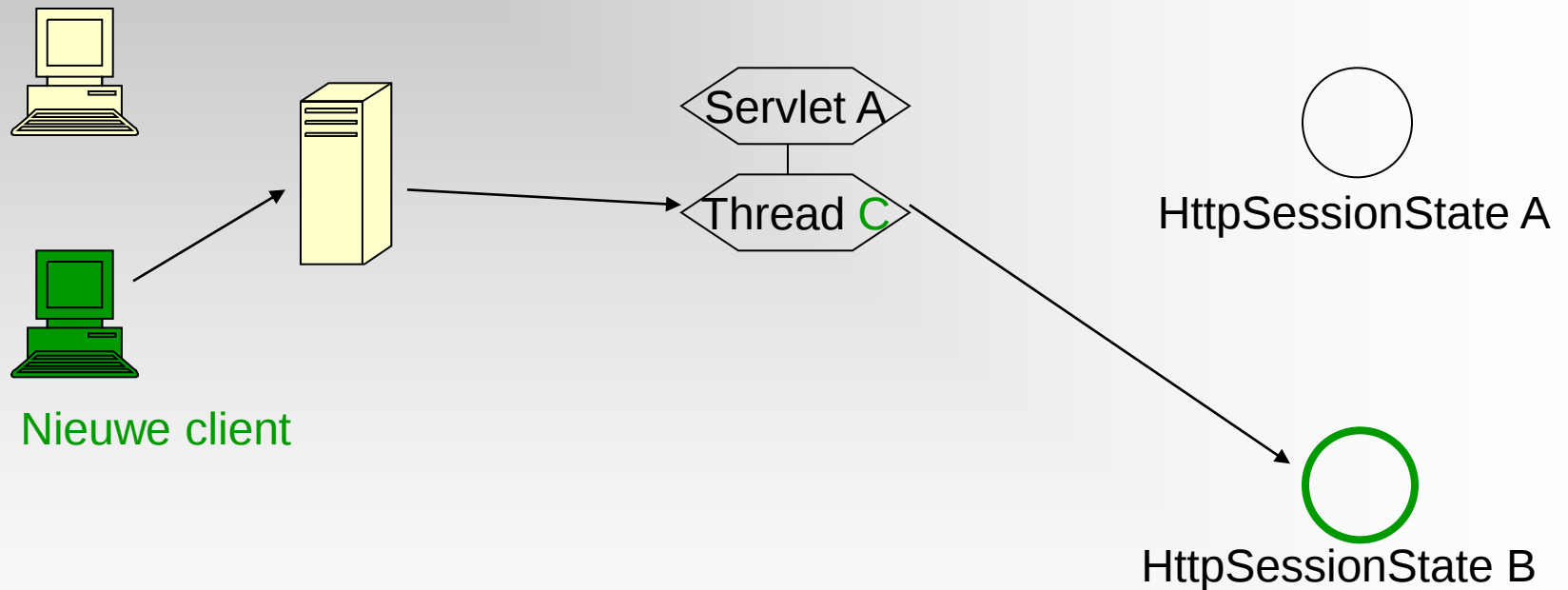
Principe sessies

- Sessie-object
 - Daaraan kunnen attributen toegekend worden
 - Attributen: sleutel-objectparen
- Type
 - `System.Web.SessionState.HttpSessionState`
- Sessie-object ophalen
 - Eigenschap `Session` van `Page`

Sessies



Sessies



Sessie

- Maakt gebruik van cookies
- Kan ook via URL herschrijven
 - Tag toevoegen aan Web.config

```
<sessionState timeout="40" cookieless="true"/>
```

Statusinformatie bewaren

- HTTP is een statusloos protocol
- Bewaren informatie over verschillende aanvragen heen
 - Op de server
 - Sessies
 - Application State
 - Profielen (Profile properties)
 - Via de client
 - Cookies, Verborgene formulier velden, QueryString van een URL, ...
 - View State
 - Controle State: statusinformatie voor een component

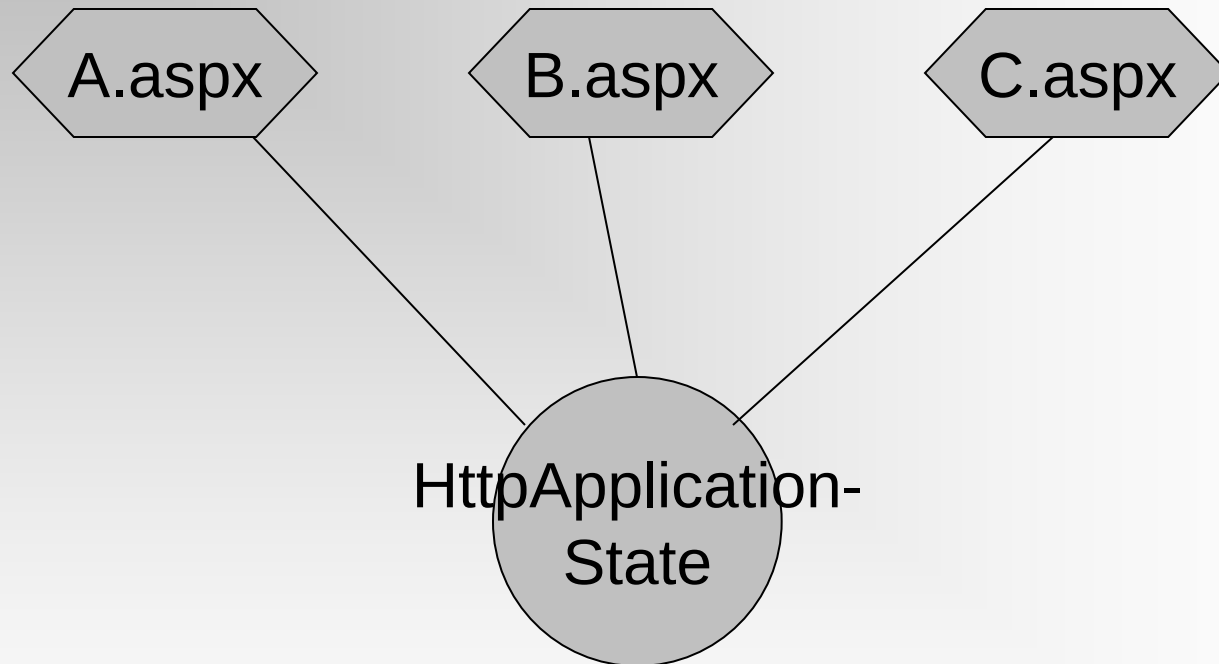
Application

- Eén Application-object per webapplicatie
- Bewaren sleutel/object-paren
 - Beschikbaar voor alle ASP.NETs van de webapplicatie
 - Synchronisatie!
- Informatie delen tussen verschillende ASP.NETs (van één applicatie): logbestanden, naam server, ...

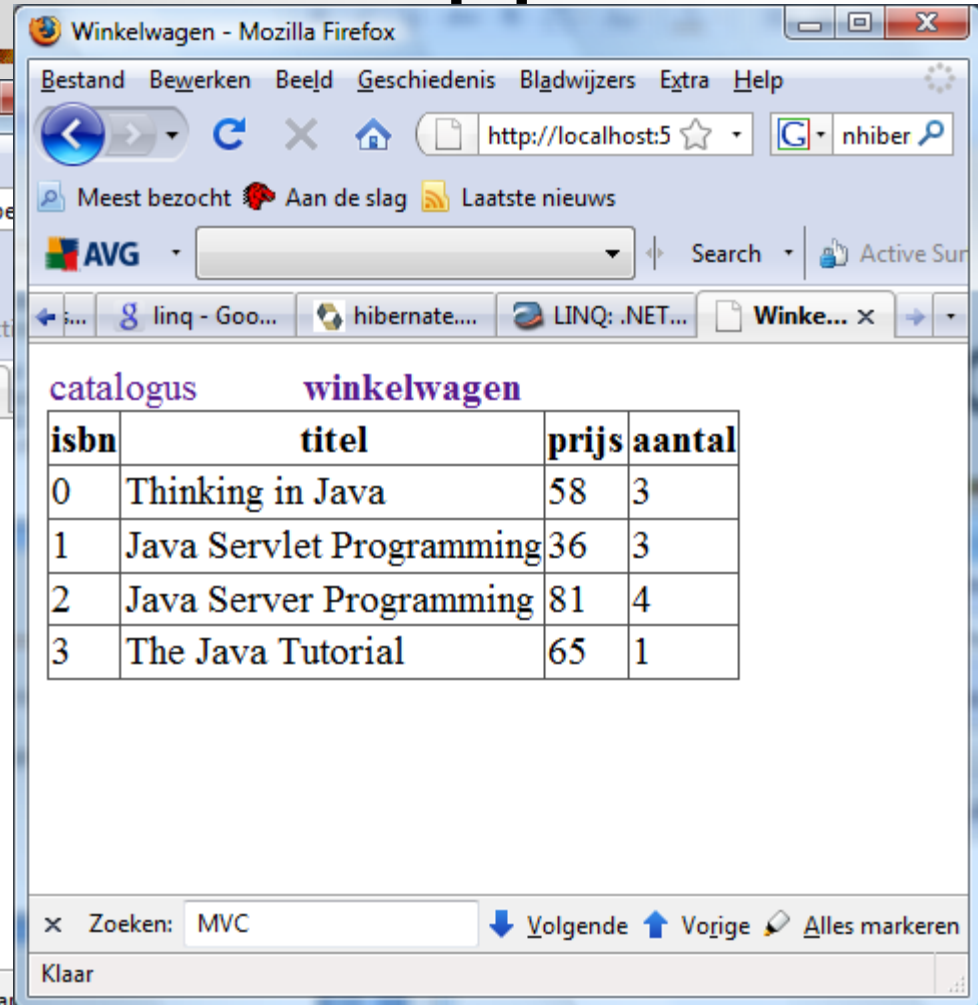
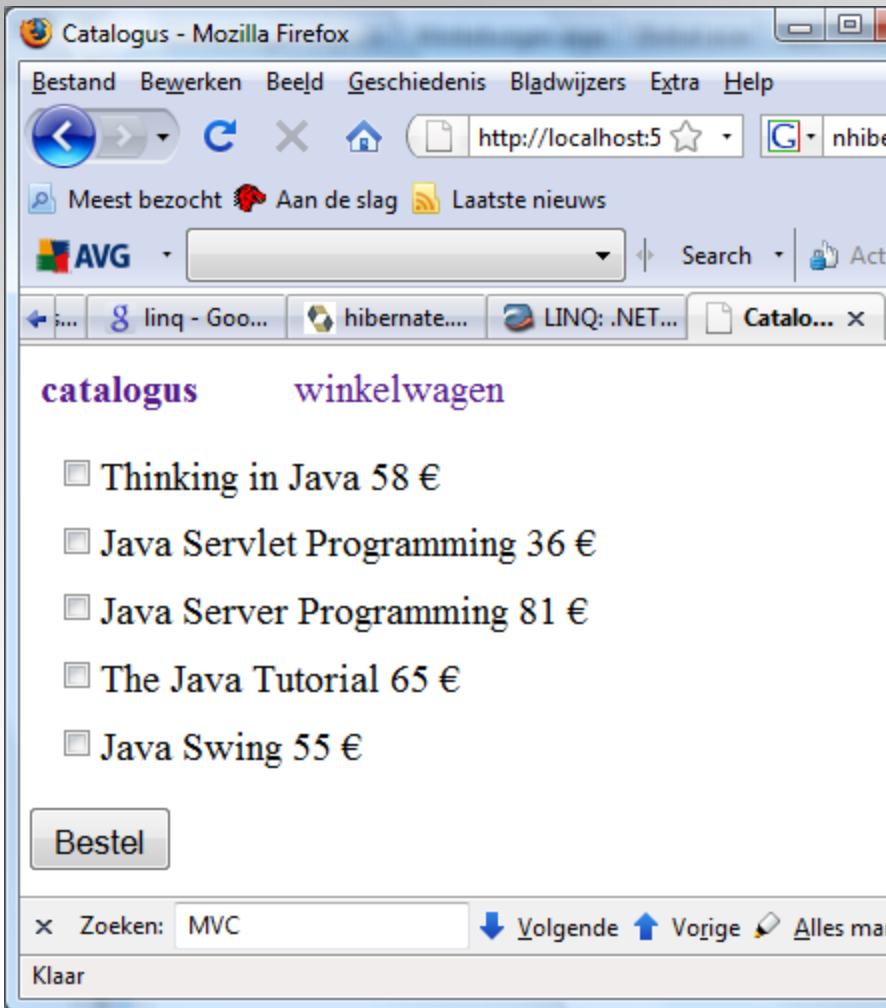
Application

- Type
 - `System.Web.HttpApplicationState`
- Ophalen
 - Eigenschap `Application` van `Page`

Application



Voorbeeld Session en Application



Voorbeeld Session en Application

- ASP.NET pagina's

- catalogus.aspx

- Toont een lijst van boeken
 - Link om boeken toe te voegen aan winkelwagentje

- winkelwagen.aspx

- Toont boeken in winkelwagen

- Global.asax

- Aanmaak Dictionary (~winkelwagen) op sessie en DataTable (boeken) op applicatieniveau

Business objecten

Boek

- + Boek (string isbn, string titel, decimal prijs)
- + string Isbn
- + string Titel
- + decimal Prijs
- + string TitelEnPrijs

BoekFabriek

- + BoekFabriek()
- + Boek[] Boeken

De klasse Page

- Eigenschappen Page
 - **HttpRequest** **Request** (As HttpRequest)
 - **HttpResponse** **Response** (As HttpResponse)
 - **HttpSessionState** **Session** (As SessionState)
 - **HttpApplicationState** **Application** (As HttpApplicationState)
 - ...

HttpResponse

- Eigenschappen
 - **string** **ContentType**
 - **HttpCookieCollection** **Cookies**
- Methodes
 - **AppendHeader**(string naam, string waarde)
 - **Clear**()
 - **Flush**()
 - **Write**(object obj)
 - **Redirect**(string url)
 - ...

Voorbeeld Session en Application

■ Session

■ Dictionary<string,int>

- boek_id: unieke identificatie boek
- Aantal: aantal te bestellen exemplaren

■ Application

■ Boek[]

- boeken

Afhalen/toevoegen objecten

- Toevoegen objecten aan session/application

`Session["winkelwagen"] = winkelwagen;`

`Application["boeken"] = boeken;`

Eigenschap
Page

- Referentie naar objecten op session/application

`winkelwagen =`

`(Dictionary<string,int>)Session["winkelwagen"];`

`boeken = (Boek[])Application["boeken"];`

Applicatie

- Virtuele map
 - Oude versie: map in \inetpub\wwwroot
- Bij de eerste aanvraag naar een pagina van de applicatie wordt een pool van applicatie-objecten aangemaakt
- Global.asax
 - Beschrijft een reeks gebeurtenissen i.v.m. sessies en applicaties
 - In de root van de virtuele map
- web.config
 - Configuratie applicatie: authenticatie, sessies, ...

Gebruik bibliotheken

- In C#-bestand: using VbnBo
- In aspx-bestand of asax-bestand:
`<%@ Import Namespace="VbnBo" %>`
- In map bin van virtuele map applicatie (bv.
 \inetpub\wwwroot\bin)
- Aanmaken dll-bestand
`csc /t:library /out:VbnBo.dll *.cs`

Global.asax.cs

■ Mogelijke gebeurtenissen

- `void Application_Start(object,EventArgs)`
- `void Application_End (object,EventArgs)`
- `void Session_Start (object,EventArgs)`
- `void Session_End (object,EventArgs)`
- `void Application_BeginRequest (object,EventArgs)`
- `void Application_EndRequest (object,EventArgs)`
- `void Application_Error (object,EventArgs)`

Global.asax

```
<%@ Application Language="C#" %>
<%@ Import Namespace="System.Collections.Generic" %>

<script runat="server">
    void Application_Start(object sender, EventArgs e)
    { ... }
</script>
```

- Omgezet naar een klasse afgeleid van `HttpApplication`

Voorbeeld Global.asax

- Aanmaak Dictionary
 - Te gebruiken om aankopen bij te houden
- Aanmaak DataTable met boeken

Statusinformatie bewaren

- HTTP is een statusloos protocol
- Bewaren informatie over verschillende aanvragen heen
 - Op de server
 - Sessies
 - Application State
 - Profielen (Profile properties)
 - Via de client
 - Cookies, Verborgene formulier velden, QueryString van een URL, ...
 - View State
 - Controle State: statusinformatie voor een component

Profielen

- Analooq aan sessie
- MAAR
 - Bij afsluiten blijft gebruikersinformatie bewaard
 - Database
 - Bestand

Statusinformatie bewaren

- HTTP is een statusloos protocol
- Bewaren informatie over verschillende aanvragen heen
 - Op de server
 - Sessies
 - Application State
 - Profielen (Profile properties)
 - Via de client
 - Cookies, Verborgene formulier velden, QueryString van een URL, ...
 - View State
 - Controle State: statusinformatie voor een component

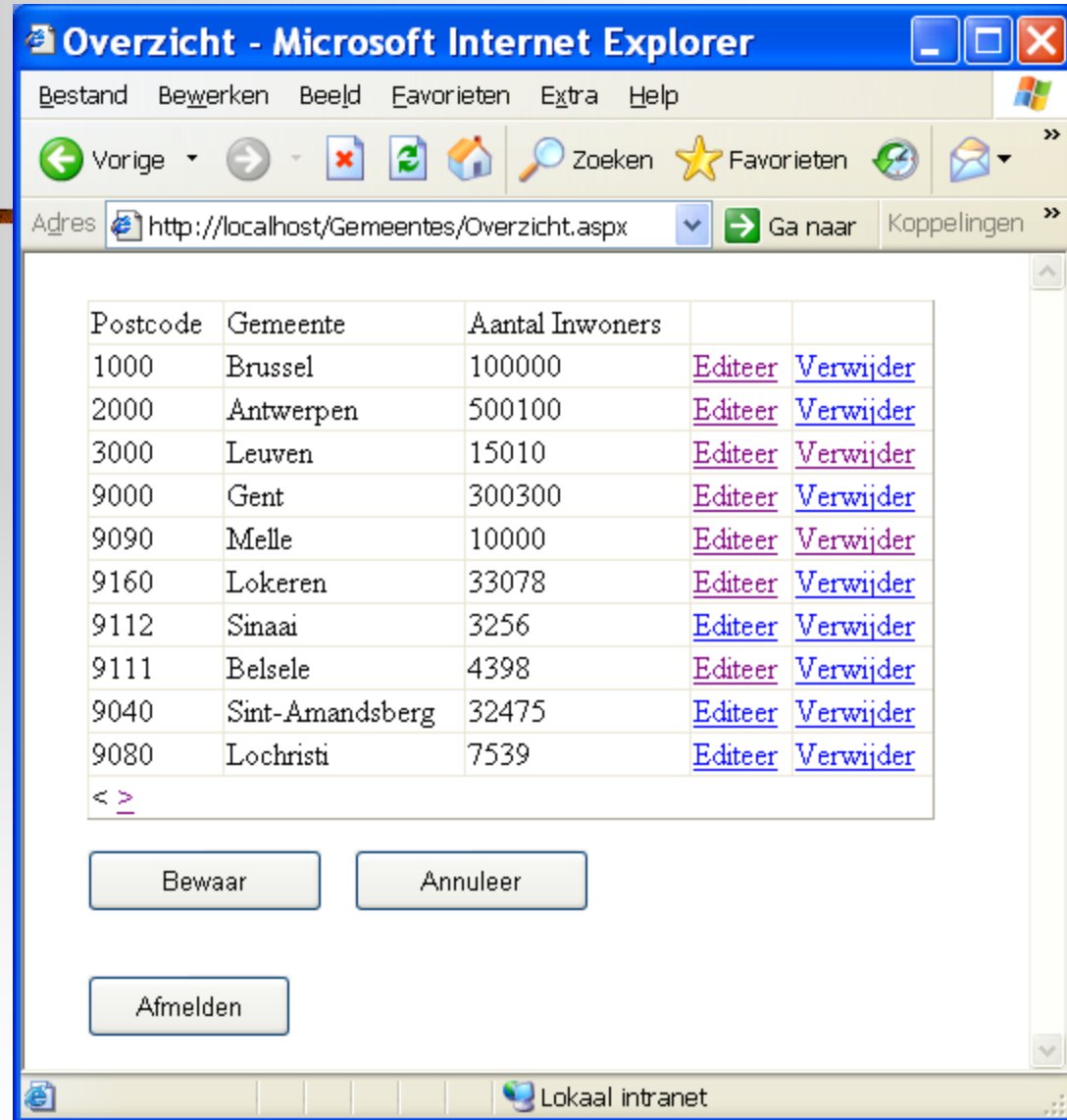
ViewState

- Bewaren
 - kleine hoeveelheden data
 - tussen verschillende aanvragen naar één pagina
- Data horende bij een request
 - Verschillende vensters = verschillende aanvragen
- Opslag in verborgen velden → form-tag moet attribuut runat="server" hebben

ViewState

- Sleutel-objectparen bewaren
 - Enkel serialiseerbare objecten mogelijk
 - Object → string (in verborgen veld)
- Elke component (en ook Page-object) heeft een eigenschap ViewState (type: StateBag)
 - Wordt gebruikt om de status van de component bij te houden
- ViewState (van Page) expliciet gebruiken
 - Bewaren inhoud variabelen/leden van de pagina

ViewState: voorbeeld



ViewState: voorbeeld

- Webapplicatie gemeente
 - overzicht.aspx
- gemeenteDS
 - Attribuu/veld van de klasse
 - Verwijzing nodig omdat de dataset aangepast wordt
 - Rijen aanpassen
 - Rijen verwijderen
 - Voorkomen telkens opnieuw inlezen DataSet

```
ViewState["gemeentes"] = gemeenteDS;  
gemeenteDS = (DataSet)ViewState["gemeentes"];
```

ViewState syntax

■ Algemeen

String sleutel;

type waarde;

ViewState[sleutel] = waarde;

waarde = (type)ViewState[sleutel];