Introduktion av ASP.NET MVC

Upphovsrätt för detta verk

Detta verk är framtaget i anslutning till kursen ASP.NET MVC vid Linnéuniversitetet.

Du får använda detta verk så här:

Det betyder att du i icke-kommersiella syften får:

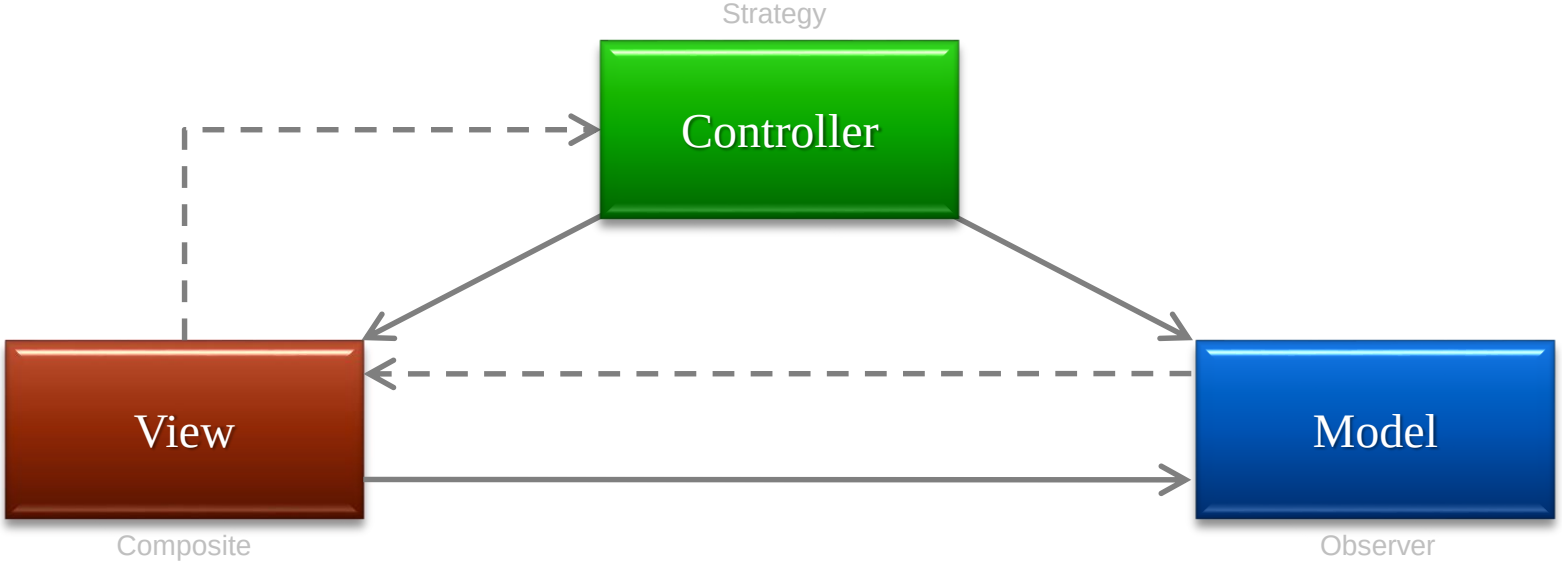
- kopiera hela eller delar av innehållet
- sprida hela eller delar av innehållet
- visa hela eller delar av innehållet offentligt och digitalt
- konvertera innehållet till annat format
- du får även göra om innehållet

Om du förändrar innehållet så ta inte med Linnéuniversitetets logotyp och symbol samt fotografier i din nya version!

Vid all användning måste du ange källan: ”Linnéuniversitetet – ASP.NET MVC” och en länk till https://coursepress.lnu.se/kurs/aspnet-mvc och till Creative Common-licensen här ovan.

Vad är MVC?

- ✓ MVC är ett designmönster där data och affärslogik separeras från presentationen och användarinteraktionen.
- ✓ Akronym för Model-View-Controller.
- ✓ ”Separation of Concerns” – modularisering.



Hur fungerar MVC? (1 av 5)

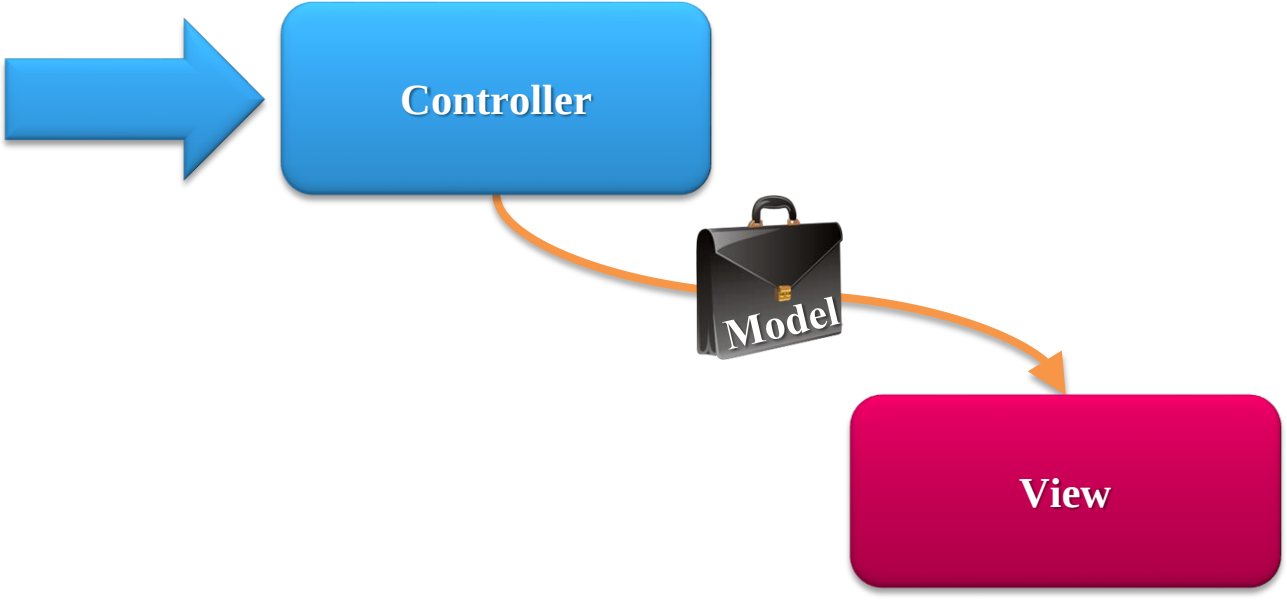
- ✓ En inkommande förfrågan går till controllern.

Hur fungerar MVC? (2 av 5)

- ✓ Controllern hanterar förfrågan genom att skapa en modell av datat som ska behandlas.

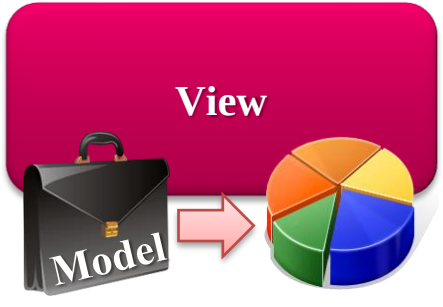
Hur fungerar MVC? (3 av 5)

- ✓ Modellen skickas till vyn.



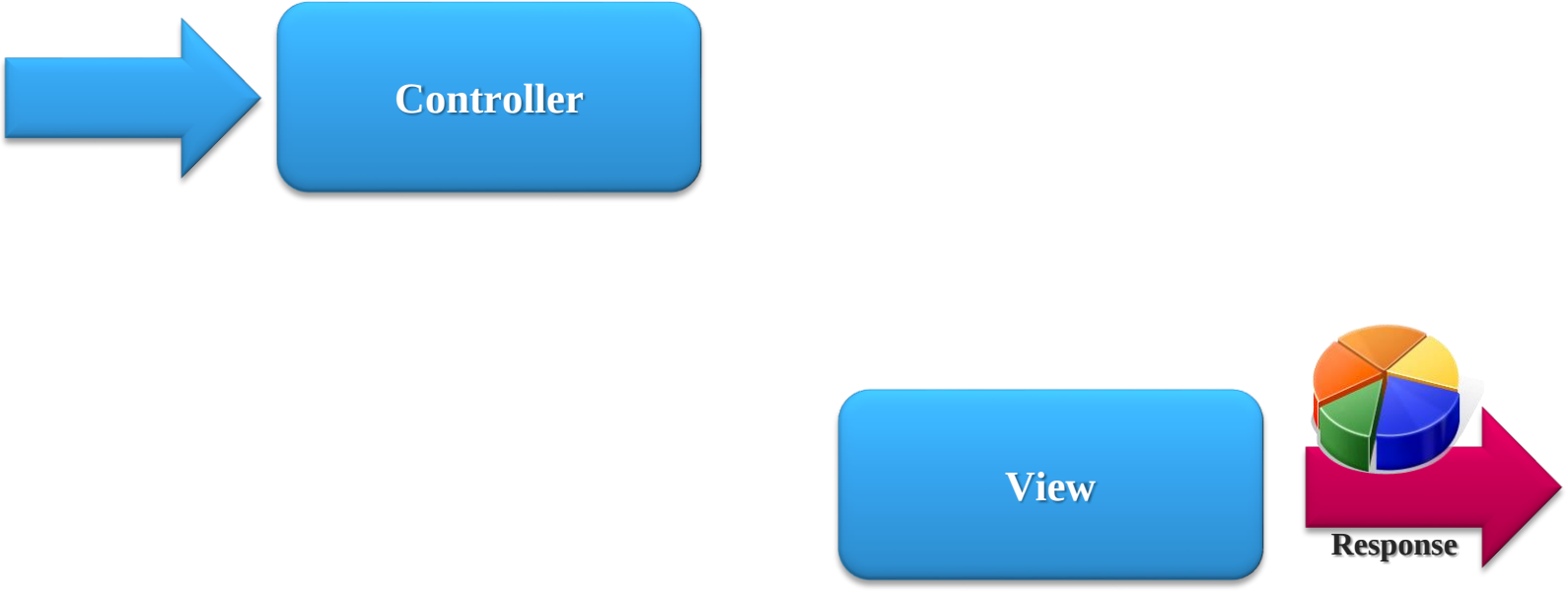
Hur fungerar MVC? (4 av 5)

- ✓ Vyn gör om modellen till lämpligt format.



Hur fungerar MVC? (5 av 5)

- ✓ Ett svar renderas.



Vad är ASP.NET MVC?

- ✓ Presenterades första gången juni 2007och första versionen släpptes mars 2009.
- ✓ ASP.NET MVC Framework…
  - …är ett tillägg till, och bygger på de bästa delarna av, ASP.NET.
  - …kopplar samman ”models”, ”views” och ”controllers” med hjälp av interface vilket gör det möjligt att helt, eller delvis, ersätta olika komponenter med egna implementeringar.
  - …gör det möjligt för dig att skapa applikationer som kan testas och underhållas.
  - …ger dig full kontroll över den HTML som renderas.
  - …ger dig kontroll över URL:erna. (ingen del av ASP.NET MVC egentligen, men…)

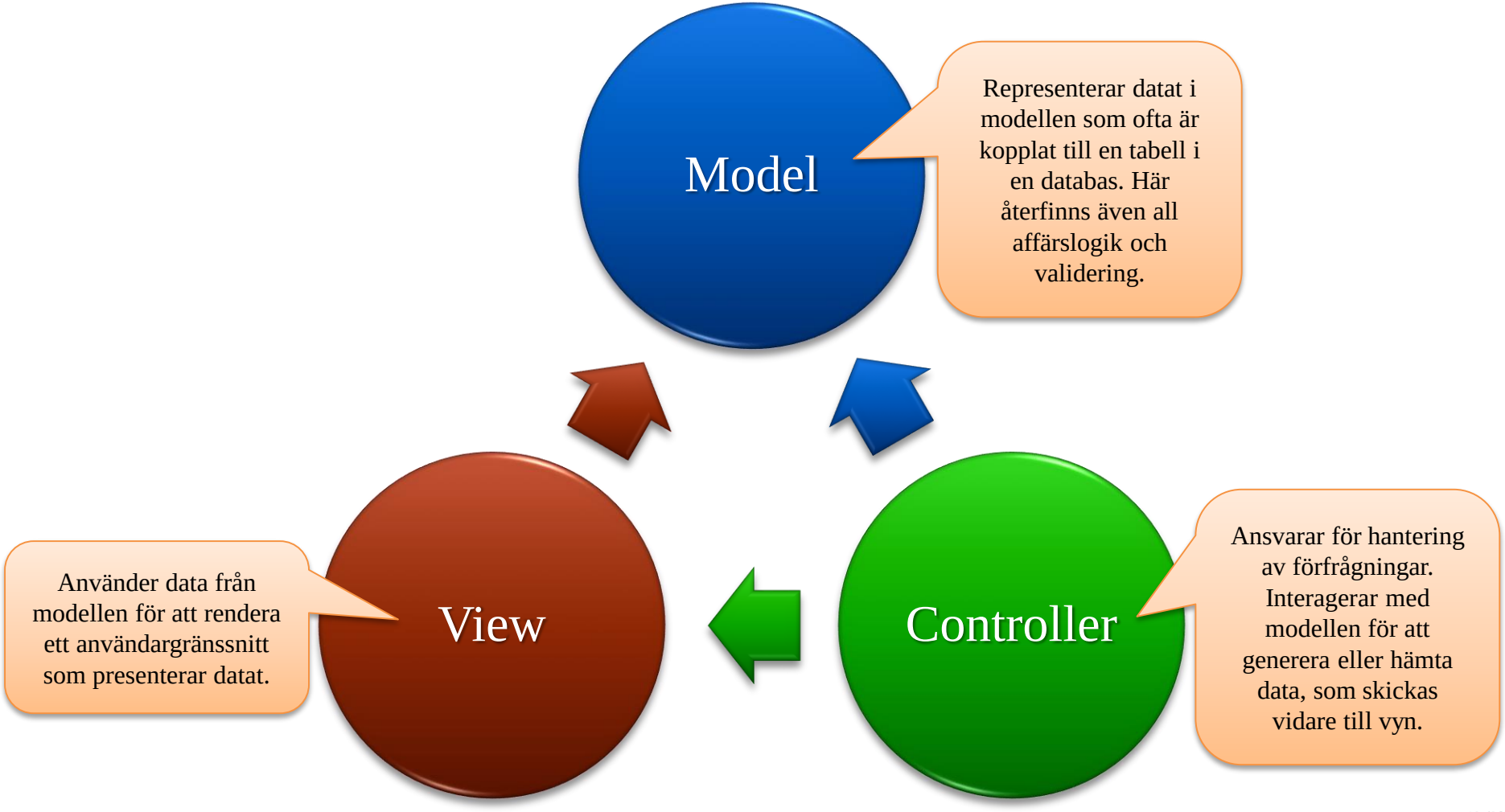
Web Forms vs MVC

- En bild säger mer…

Web Forms vs MVC

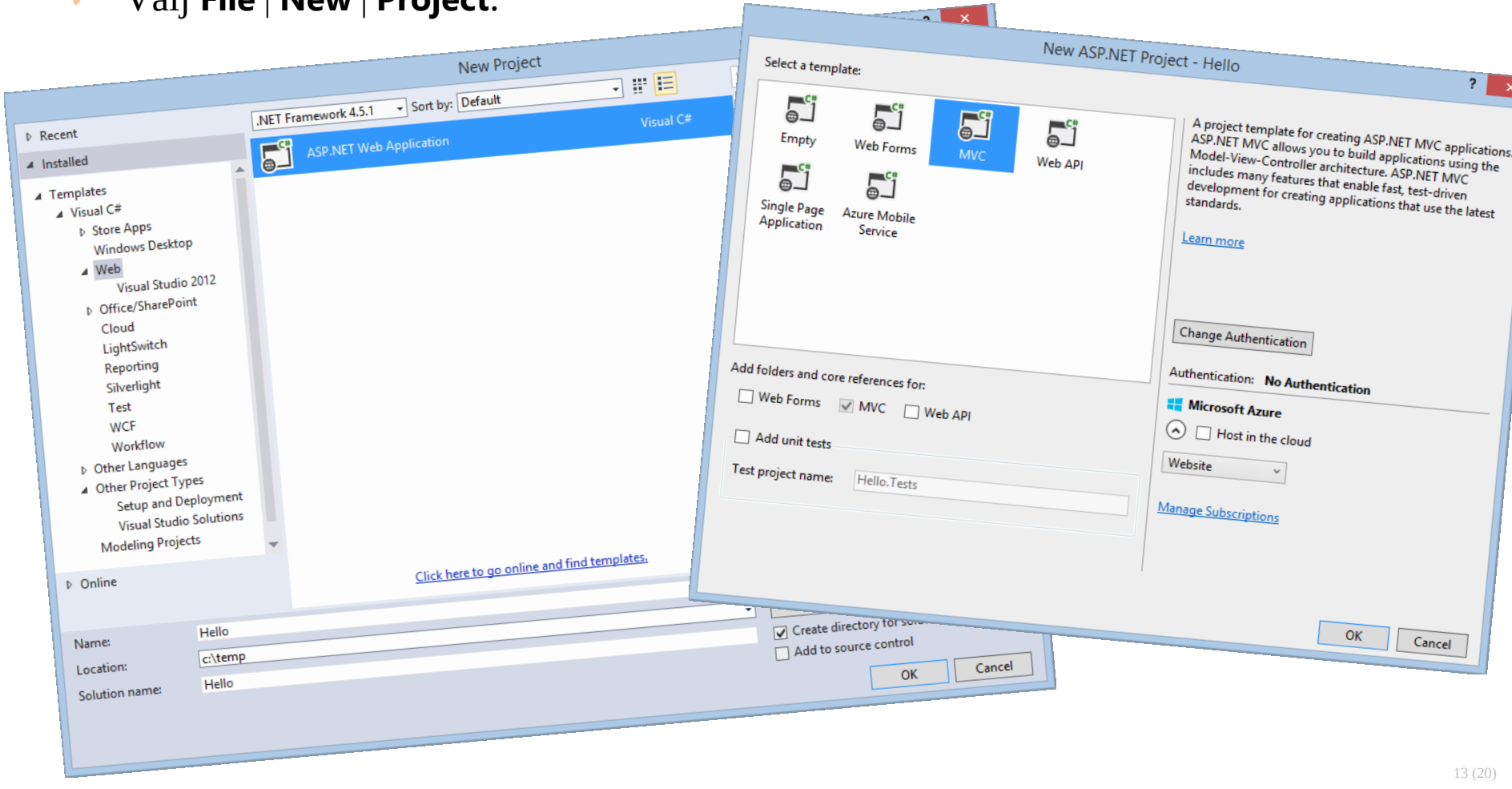
ASP.NET Web Forms	ASP.NET MVC
(Du kan detta redan!)	En helt ny modell du måste lära dig.
Stöder ASP.NET Server Controls.	Har preliminärt bara stöd för rå HTML och Javascript.
Automatisk hantering av kontrollers status mellan ”postbacks”.	Manuell hantering…
Funnits sedan 2002.	Version 1.0 släpptes den 9 april 2009.
Begränsade möjligheter för testdriven utveckling (Test-Driven Development, TDD).	Uppmuntrar och inkluderar TDD!
”Web Site” eller ”Project”.	Endast ”Project”!
Båda har tillgång till inbyggda objekt, t.ex.: Session, Cache och Application.	
Båda stöder användning av ”ASP.NET provider models”, t.ex.: (Membership), Profile och Sitemap	

Så fungerar ASP.NET MVC



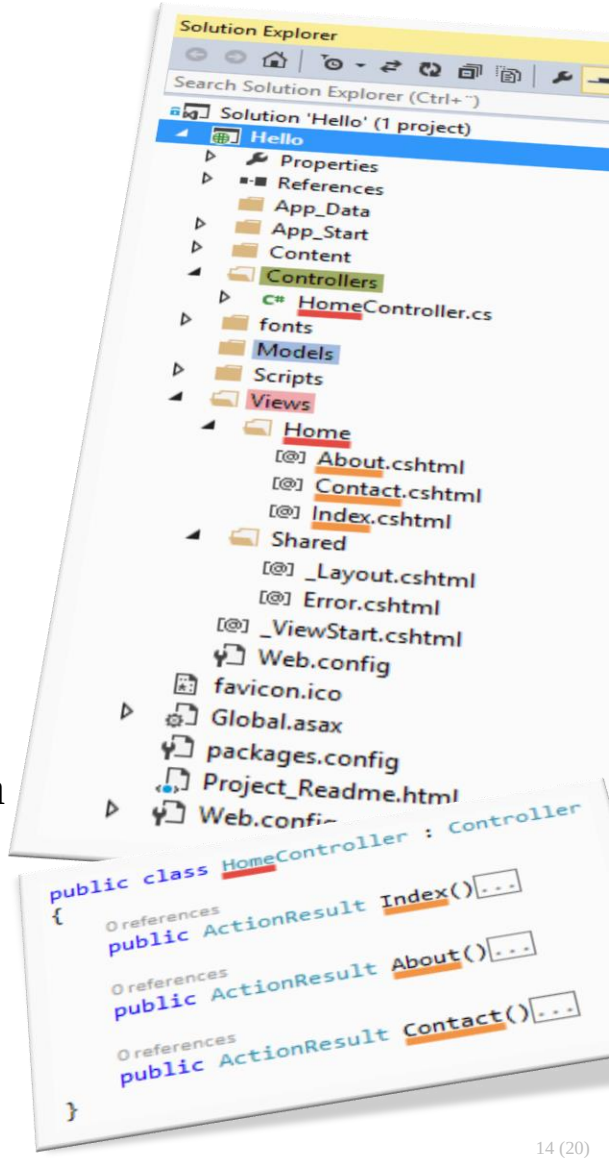
Ett ”ASP.NET MVC”-projekt

- Välj **File** | **New** | **Project**.



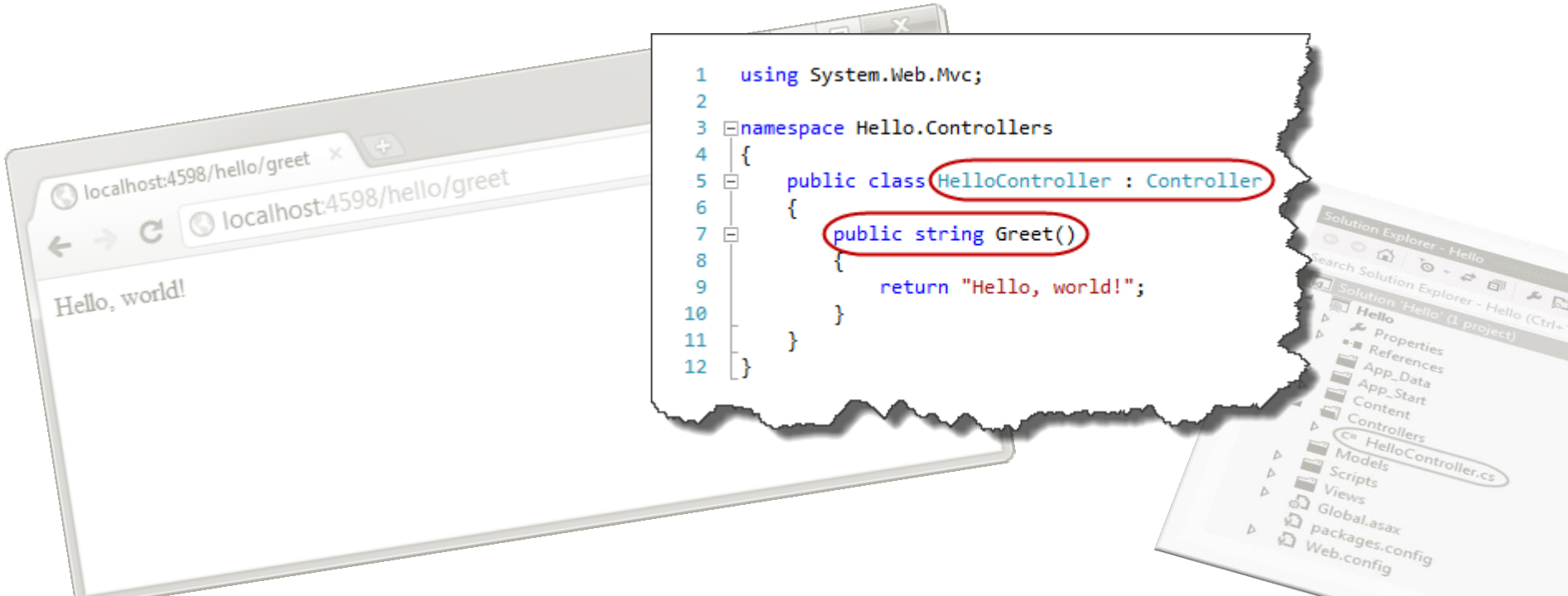
Projektet skapat!

- ✓ Namnen på katalogerna **Models**, **Views** och **Controllers** förklarar deras syfte och innehåll…
- ✓ Konventioner:
  - Namnet på en controllerklass måste avslutas med ”**Controller**” och vara placerad i katalogen **Controllers**.
  - Modellklasser kan placeras var som helst, men katalogen **Models** är ett bra ställe vid ett fåtal klasser.
  - Varje controller har en dedikerad katalog, med samma namn som controllern, under katalogen **Views** innehållande vyerna.
    - Underkatalogen **Shared** innehåller vyer samt partiella vyer som kan delas av flera vyer.



Controllrar

- ✓ En controller måste ärva från klassen `Controller` och dess namn måste avslutas med Controller.
- ✓ En controller kan direkt returnera en sträng med HTML till webbläsaren.
- ✓ Alla publika metoder kan efterfrågas av en webbläsare.



Controllrar

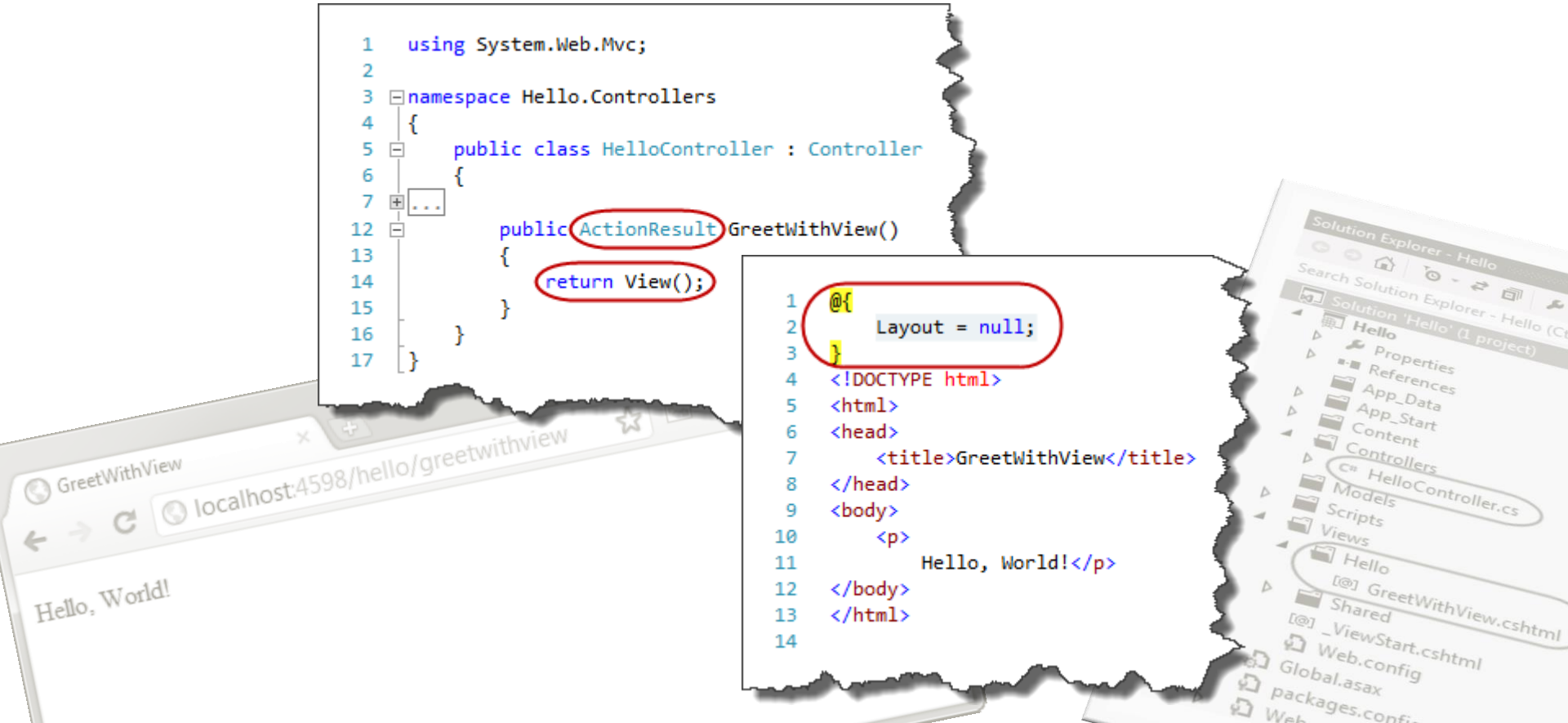
- Vanligast är att en controllermetod returnerar ett objekt av en typ som ärver från `ActionResult`.

Typer	Syfte	Exempel
ViewResult	Visar en vy (”view”).	return View();
PartialViewResult	Visar en partiell vy.	return PartialView();
RedirectToRouteResult	Omdirigerar till en rutt.	return RedirectToRote("demo");
RedirectResult	Enkel URL-omdirigering.	return Redirect("http://lnu.se");
ContentResult	Returnerar text.	return Content(rss, ”application/rss+xml");
FileResult	Returnera binärdata.	return File(@".\Document1.docx", ”application/vnd.ms-word");
JsonResult	Returnerar objekt serialiserade som Json.	return Json(myObject);
JavaScriptResult	Returnerar Javascript som ska köras av webbläsaren. Används med Ajax.	return JavaScript(”$(#elementName).hide();”);
HttpUnauthorizedResult	Returnerar koden 401 – inte autentiserad.	return new HttpUnauthorizedResult();
EmptyResult	Returnerar inget.	return new EmptyResult();

Fler typer hittar du på http://msdn.microsoft.com/en-us/library/system.web.mvc.actionresult(v=vs.118).aspx

Controllermetod som returnerar ViewResult

- Genom att låta controllermetoden returnera ett `ViewResult`-objekt meddelas ramverket att rendera en specifik vy.



Controllermetoden skickar data till vyn

- ✓ Data överförs till vyn med hjälp av egenskapen `ViewBag` (av typen `dynamic`).

```
1  using System;
2  using System.Web.Mvc;
3
4  namespace Hello.Controllers
5  {
6      public class HelloController : Controller
7      {
8          ...
18         public ActionResult GreetWithViewBag()
19         {
20             int hour = DateTime.Now.Hour;
21             ViewBag.Greeting = hour < 12 ?
22                 "Good morning" : "Good afternoon";
23             return View();
24         }
25     }
26 }
```

```
1  @{
2      ViewBag.Title = "GreetWithViewBag";
3  }
4  <h2>
5      GreetWithViewBag</h2>
6  <p>
7      @ViewBag.Greeting, World!</p>
8
```

GreetWithViewBag

localhost:4598/hello/g

GreetWithViewBag

Good morning, World!

dynamic (C# Referen

The **dynamic** type enables the operations in which it occurs to bypass compile-time type checking. Instead, these operations are resolved at run time. The **dynamic** type simplifies access to COM APIs such as the Office

Solution Explorer - Hello

Search Solution Explorer - Hello (Ctrl+

Solution 'Hello' (1 project)

Hello

Properties

References

App_Data

App_Start

Content

Controllers

HelloController.cs

Models

Scripts

Views

Hello

GreetWithView.cshtml

GreetWithViewBag.cshtml

Shared

_ViewStart.cshtml

Web.config

Global.asax

…controllern kan även skicka ett datamodellobjekt.

```
using System;

namespace Hello.Models
{
    public class Salutation
    {
        public string Greeting
        {
            get
            {
                return DateTime.Now.Hour < 12 ?
                    "Good morning" : "Good afternoon";
            }
        }
    }
}
```

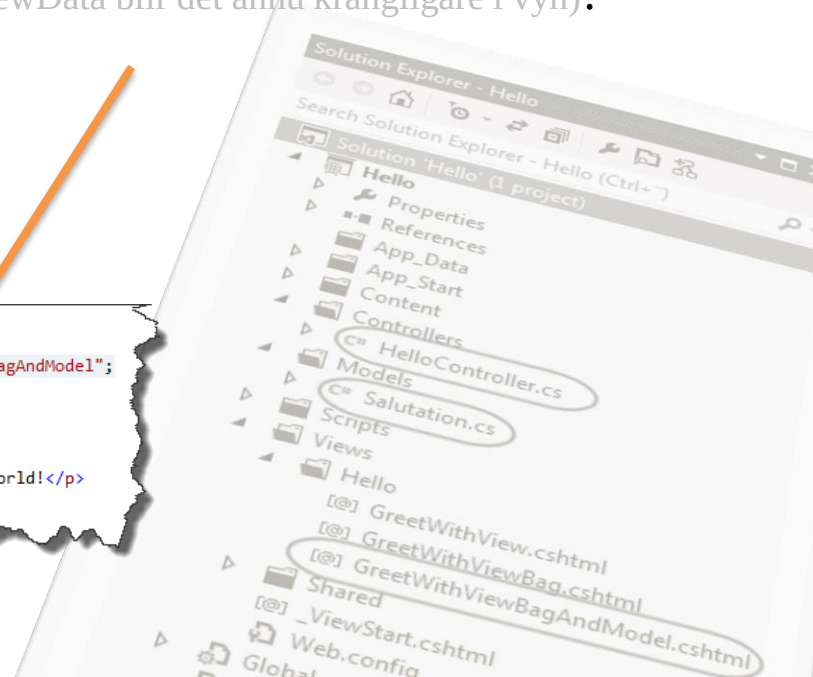
```
using System;
using System.Web.Mvc;
using Hello.Models;

namespace Hello.Controllers
{
    public class HelloController : Controller
    {
        ...

        public ActionResult GreetWithViewBagAndModel()
        {
            ViewBag.Salutation = new Salutation();
            return View();
        }
    }
}
```

```
@{
    ViewBag.Title = "GreetWithViewBagAndModel";
}
<h2>
    GreetWithViewBagAndModel</h2>
<p>
    @ViewBag.Salutation.Greeting, World!</p>
```

- Controllermetoden skickar med ett modellobjekt till vyn med hjälp av egenskapen `ViewBag`…
- …varför Visual Studio inte kan ”hjälpa till” med IntelliSense (används ViewData blir det ännu krångligare i vyn)!



Vyn kan vara starkt typad

- Controllermetoden skickar med modellen till vyn med hjälp egenskapen `Model`.
- Att en vy är starkt typad innebär att vyns datamodellobjekt måste vara av specificerad typ.

```
1  using System;
2  using System.Web.Mvc;
3  using Hello.Models;
4
5  namespace Hello.Controllers
6  {
7      public class HelloController : Controller
8      {
9          ...
32         public ActionResult GreetWithModel()
33         {
34             var model = new Salutation();
35             return View(model);
36         }
37     }
38 }
```

```
1  @model Hello.Models.Salutation
2
3  @{
4      ViewBag.Title = "GreetWithModel";
5  }
6
7  <h2>GreetWithModel</h2>
8  <p>
9      @Model.Greeting, World!</p>
```