MVC 5 (Model-View-Controller)

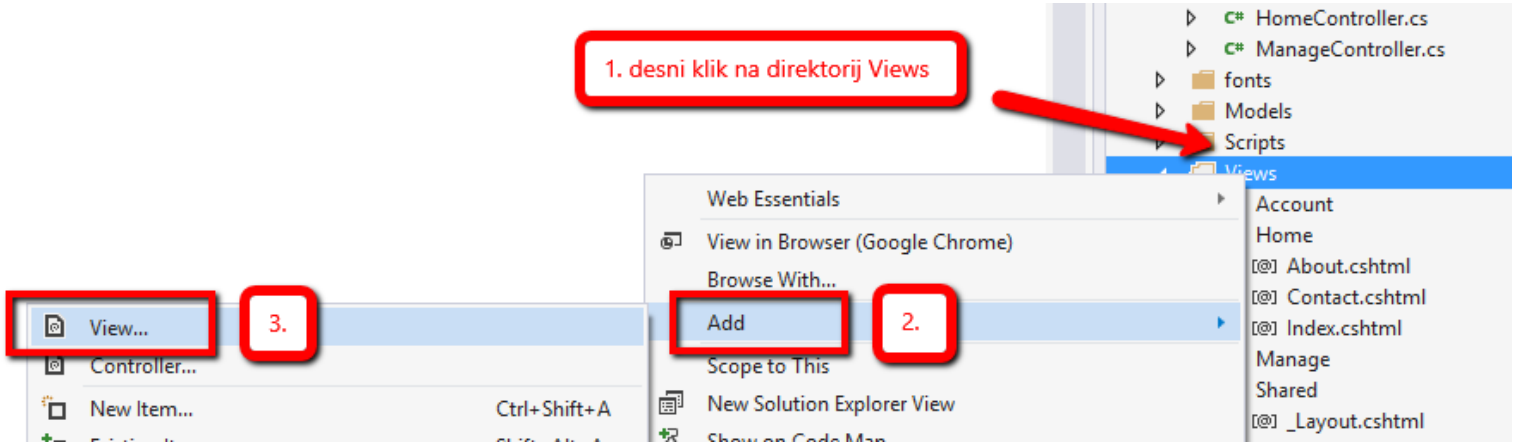
Pogledi (Views)

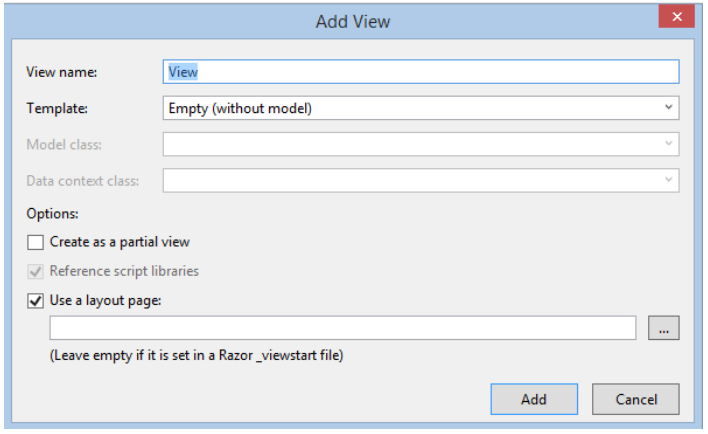
Što je Pogled (View)?

- Pogled (View) je datoteka na osnovu koje će generator pogleda generirati HTML i poslati ga pregledniku (Browser) tijekom obrade trenutno zahtjevane stranice
- View može biti i obična HTML datoteka, ali je u MVC-u ona najčešće .cshtml datoteka koja se uređuje pomoću Razor View Enginea

Generiranje Viewa (I)

- Direktno na Views direktoriju

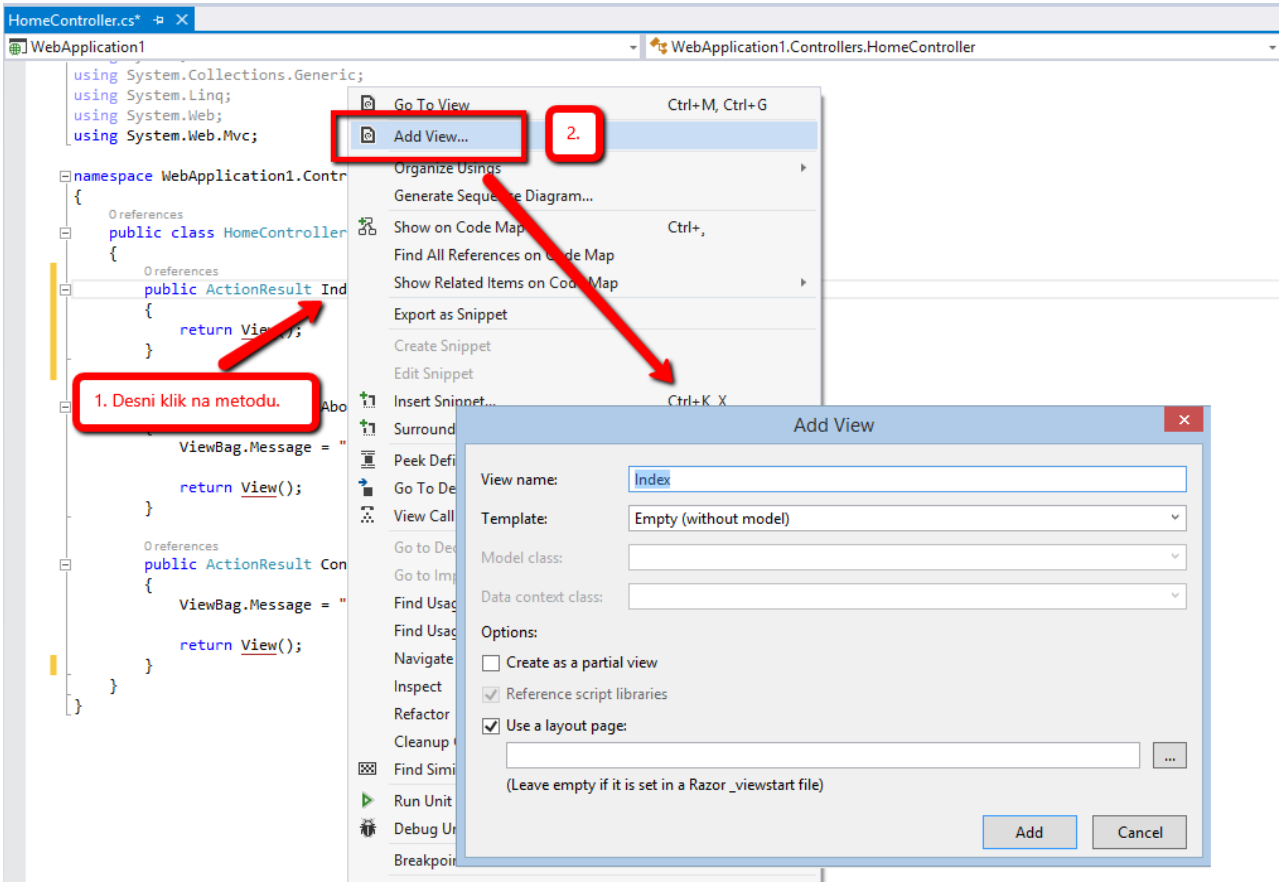




VERN'
VISOKO UČILIŠTE

Generiranje Viewa (II)

- Na akcijskoj metodi:



Što je strogo-tipizirani (Strongly-typed) View?

Strogo-tipizirani (strongly-typed) view sadrži podatke koji se baziraju na određenom modelu.

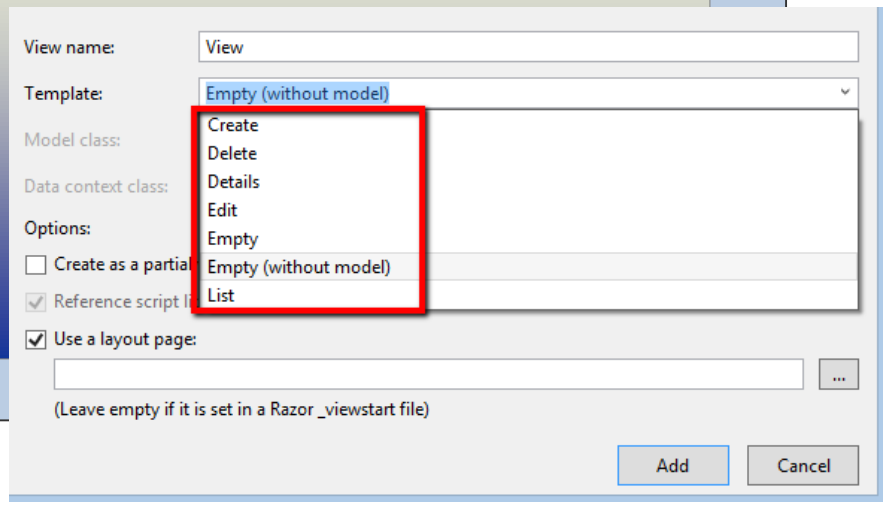
```
@model WebApplication1.Models.RegisterViewModel
@{
   ViewBag.Title = "Register";
}

<h2>@ViewBag.Title.</h2>

@using (Html.BeginForm("Register", "Account", FormMethod.Post, new { @class =
"form-horizontal", role = "form" }))
{
   @Html.AntiForgeryToken()
   <h4>Create a new account.</h4>
   <hr />
   @Html.ValidationSummary("", new { @class = "text-danger" })
   <div class="form-group">
      @Html.LabelFor(m => m.Email, new { @class = "col-md-2 control-label" })
      ...
```

CRUD MVC Views

Korištenjem Add View naredbe Visual Studio radne okoline, lako je kreirati strogo-tipizirane view-ove koji mogu služiti kreiranju, čitanju, ažuriranju ili brisanju podataka.

Ovo su tipovi view-ova koje Add View dialog box može kreirati:

- List
- Details
- Create
- Edit
- Delete
- Empty



Što je parcijalni MVC View?

Parcijalni View

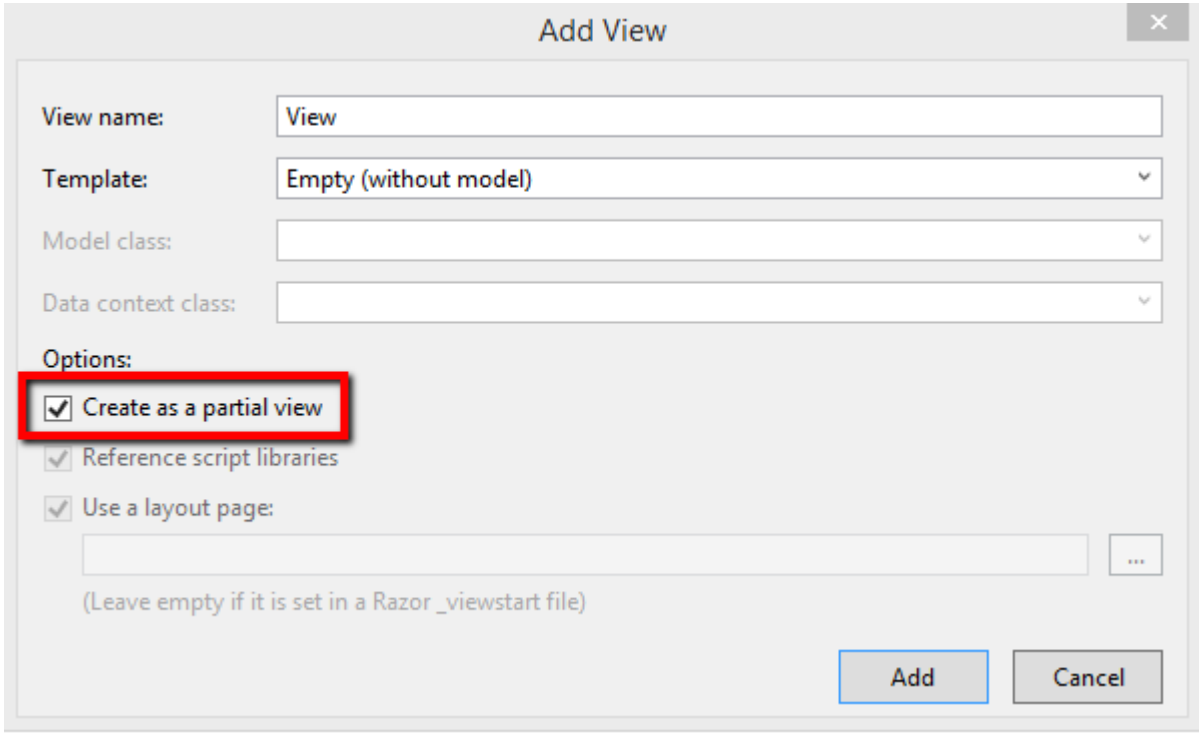
```
@model PartialViewDemo.Models.Proizvod
<!DOCTYPE html>
<html>
...
```

Stranica koja renderira parcijalni View

```
...
<head>
   <title>Novi proizvod</title>
</head>
<body>
...
...
   Html.RenderPartial("~/Views/Shared/_Proizvod.cshtml", proizvod);
...
</body>
</html>
```

Kreiranje parcijalnog Viewa

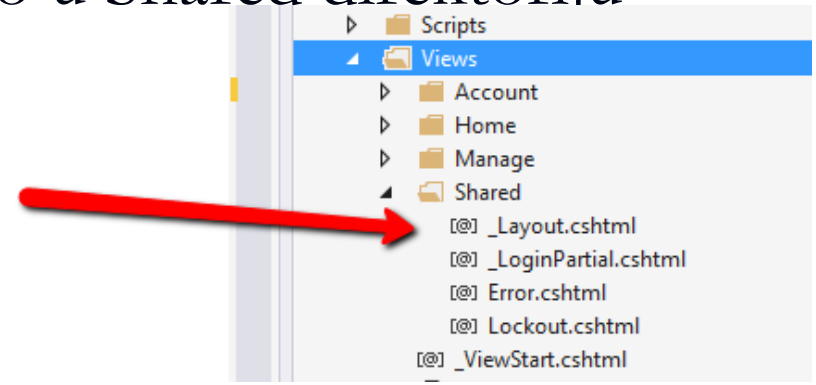
- Obavezno odabrati opciju – Create as partial view – tijekom kreiranja viewa:



Layout datoteke

- Koristimo ih za održavanje konzistentnog izgleda i dojma (look-and-feel) naše web aplikacije. Od prije taj koncept je poznat kao - Master Page.
- Njihov je kod sličan kao i kod View-ova
- Kada pogled naslijedi Layout naslijeđuje i sav izgled/dizajn koji ide s njim
- Layout datoteke obično držimo u Shared direktoriju

Shared direktorij je mjesto za Layout i Partial View datoteke



Pozivanje MVC View-a

ActionResult klasa enkapsulira rezultat akcijske metode koja može biti nešto od navedenog:

- ContentResult — vraća korisnički definiran rezultat.
- EmptyResult — vraća null rezultat.
- FileResult — vraća binarni file.
- JavaScriptResult — vraća JavaScript.
- JsonResult — vraća serijaliziran Json objekt.
- **PartialViewResult — vraća parcijalni view.**
- RedirectResult — preusmjerava na drugu akcijsku metodu korištenjem URL.
- RedirectToRouteResult — preusmjerava na drugu akcijsku metodu.
- **ViewResult — Renderira view. Ova klasa sadrži svojstva koja identificiraju View za prikazivanje, ime View-a, ime master view-a, podatke, privremene podatke, i kolekciju za aplikacijski generator view-ova (view engine).**

Za vraćanje Viewa koristi se ViewResult ili PartialViewResult.

Dohvaćanje podataka iz Request-a

- ✓ Korištenjem Request objekta.
- ✓ Korištenjem FormCollection kolekcije.
- ✓ Korištenjem Model Binder-a.

```
[HttpPost]
public ActionResult Create(Blog blog)
{
    if (ModelState.IsValid)
    {
        // TODO: programska logika
    };
    return RedirectToAction("Index");
}
```

Slanje podataka od Controllera do Viewa

- ✓ Korištenjem ViewBag objekta.
- ✓ Korištenjem ViewData objekta.
- ✓ Korištenjem TempData objekta.

Sličnosti ViewBag i ViewData objekata

- Cilj im je omogućiti komunikaciju između controllera i view-ova unutar serverskog poziva.
- Omogućuju prenošenje podataka od controllera do viewa tijekom trajanja requesta (zahtjeva).
- Kratkog su vijeka, tj. vrijednosti postaju – null – nakon redirekcije.

Različitosti ViewBag i ViewData objekata

- ViewData je Dictionary klasa. Njenim članovima (vrijednostima) pristupamo preko key stringova.
- ViewBag je dynamic objekt, to je nova karakteristika jezika C# od verzije 4.0. Svojstva tog objekta mogu biti dinamički određena tijekom run-timea.
- ViewData zahtijeva cast operaciju za kompleksne tipove tijekom korištenja. Također, prije korištenja treba provjeriti da li je vrijednost null da bi izbjegli greške.
- ViewBag ne zahtijeva cast operaciju za kompleksne tipove tijekom korištenja.

TempData objekt

- TempData je Dictionary klasa. Vrijednostima/objektima pristupamo preko key stringova.
- Spremljen je u privremenu Session varijablu.
- Razlika od prethodnika je što TempData čuva informacije tijekom HTTP Requesta. Znači, informacije prenesene sa jedne stranice na drugu.
- Informacije će biti sačuvane tijekom redirekcije sa jedne akcije na drugu unutar istog controllera, ili tijekom prelaska sa jednog na drugi controller.
- Zahtjeva cast operaciju za komplekne tipove. Zahtjeva provjeru null vrijednosti.