



UPPSALA  
UNIVERSITET

# . NET-programmering

View Component



UPPSALA  
UNIVERSITET

# Welcome to my demo!

## Student List

	<b>Id Code</b>	<b>Name</b>	<b>Enrollment No</b>
1	L0001	Amelia Gundersson	201804150001
2	L0002	Charlie Jansson	201804150002
3	L0003	Bertil Paulsson	201804150003

## Teacher List

	<b>Id Code</b>	<b>Name</b>
<u>1</u>	TT	Tomas Trivial
<u>2</u>	JC	Jonas Cluedo
<u>3</u>	RM	Rakel Monopol

## Course List

Choose Course ▼



UPPSALA  
UNIVERSITET

# Teacher Detail

1

Tomas Trivial

TT

@model Teacher

```
<h2>Teacher Detail</h2>
```

```
<div>
```

```
    <p>@Model.TeacherID </p>
```

```
    <p>@Model.Name </p>
```

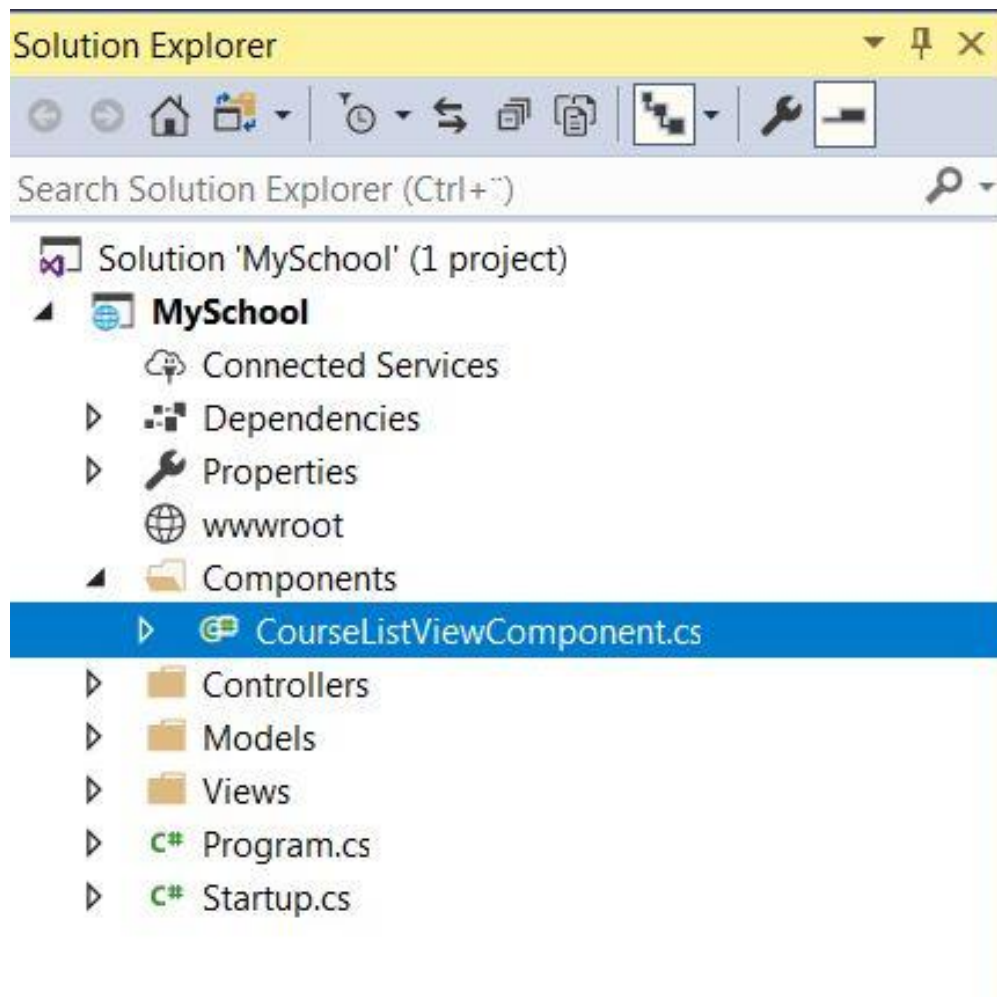
```
    <p>@Model.Code</p>
```

```
</div>
```

# View Component

- Vanlig C# klass
  - Ger Partial View data
  - Se den som en speciell Action
  - Kan inte ta emot HTTP requests
  - Datat inkluderas i föräldravyn
- Kan skapas på 3 olika sätt
  - En POCO View Component
  - Ärva av ViewComponent basklass
  - Använda ViewComponent Attribut

# Exempel





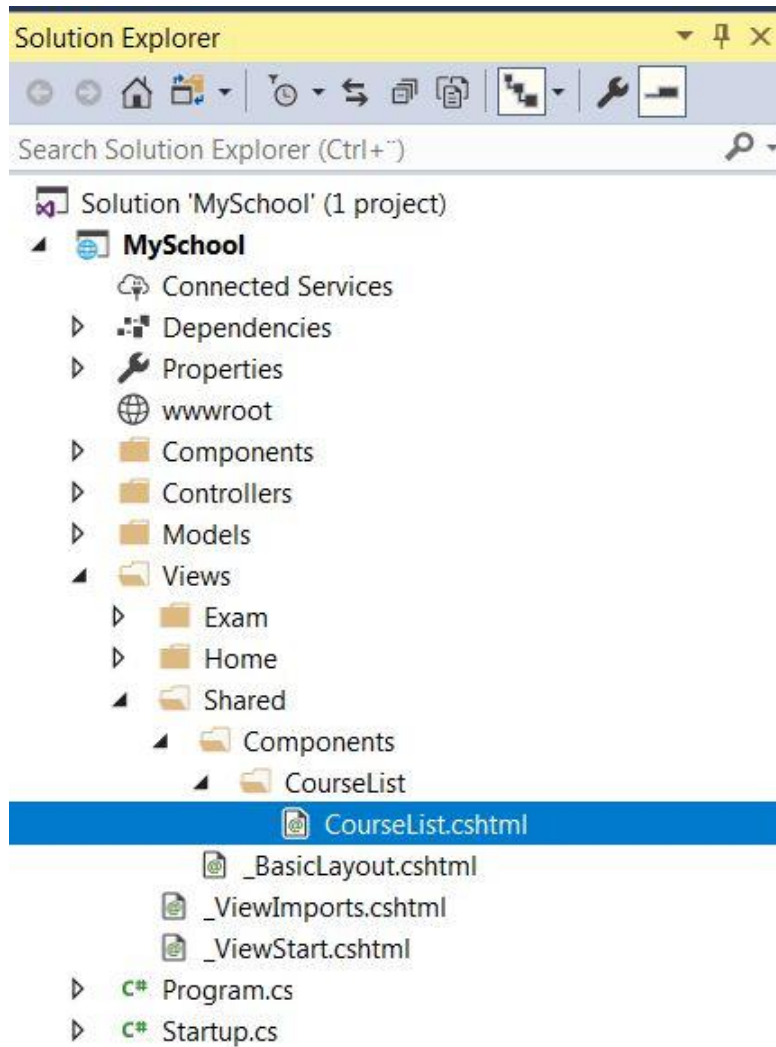
# ViewComponent (exempel)

```
CourseListViewComponent.cs  X
MySchool  MySchool.Components.Cour:
1  using Microsoft.AspNetCore.Mvc;
2  using MySchool.Models;
3
4  namespace MySchool.Components
5  {
6      public class CourseListViewComponent:ViewComponent
7      {
8          private ISchoolRepository repository;
9
10         public CourseListViewComponent(ISchoolRepository repo)
11         {
12             repository = repo;
13         }
14
15         public IViewComponentResult Invoke()
16         {
17             return View("CourseList", repository.Courses);
18         }
19     }
20 }
21
```



UPPSALA  
UNIVERSITET

# Exempel forts







# CourseList

```
CourseList.cshtml  CourseListViewComponent.cs
1  @model IEnumerable<Course>
2
3
4  <h3>Course List</h3>
5  <select>
6      <option>Choose Course</option>
7      @foreach (Course course in Model)
8      {
9          <option>@course.Name</option>
10     }
11 </select>
```



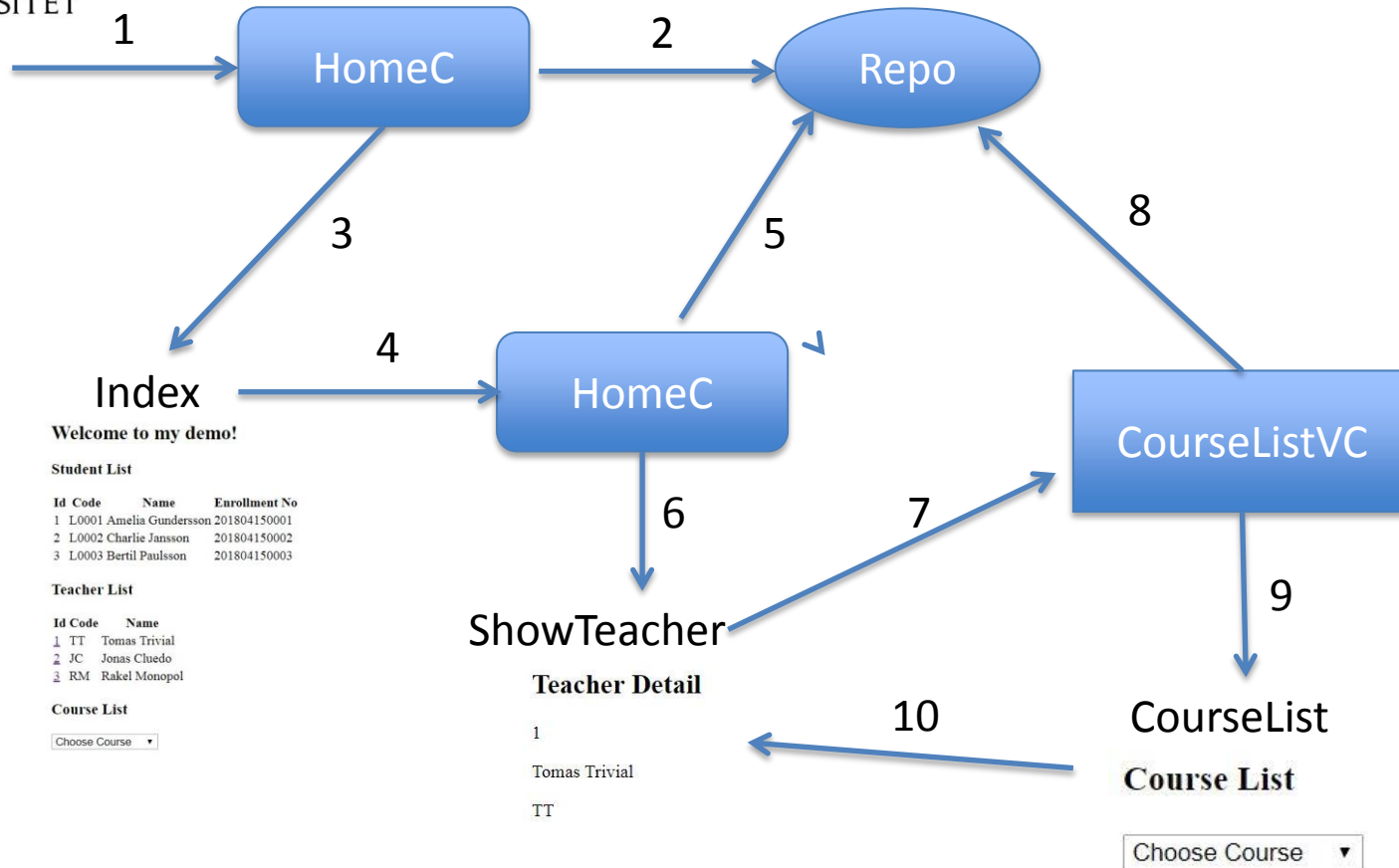


# ShowTeacher

```
ShowTeacher.cshtml  CourseList.cshtml  CourseListViewComponent.  
1  
2  @model Teacher  
3  
4  <h2>Teacher Detail</h2>  
5  
6  <div>  
7      <p>@Model.TeacherID</p>  
8      <p>@Model.Name</p>  
9      <p>@Model.Code</p>  
10 </div>  
11  
12 @await Component.InvokeAsync("CourseList")
```



UPPSALA  
UNIVERSITET





UPPSALA  
UNIVERSITET

# Slutresultatet

## Teacher Detail

2

Jonas Cluedo

JC

@model Teacher

## Course List

Choose Course ▼

@model IEnumerable<Course>

ViewComponent

# Parameter till ViewComponent

```
public ActionResult ShowTeacher(int id)
{
    var teachers = mymodel.teachers.Where
        (m => m.TeacherId == id).First();

    return View(teachers);
}
```



UPPSALA  
UNIVERSITET

# Interface och Repository

## **ISchoolRepository.cs**

```
Task<Teacher> GetTeacherDetail(int id);
```

## **FakeSchoolRepository.cs**

```
public Task<Teacher> GetTeacherDetail (int id)
{
    return Task.Run(() =>
    {
        var teacherDetail = Teachers.Where(td => td.TeacherID == id).First();
        return teacherDetail;
    });
}
```



UPPSALA  
UNIVERSITET

# HomeController

```
public ActionResult ShowTeacher(int id)
{
    ViewBag.Tid = id;

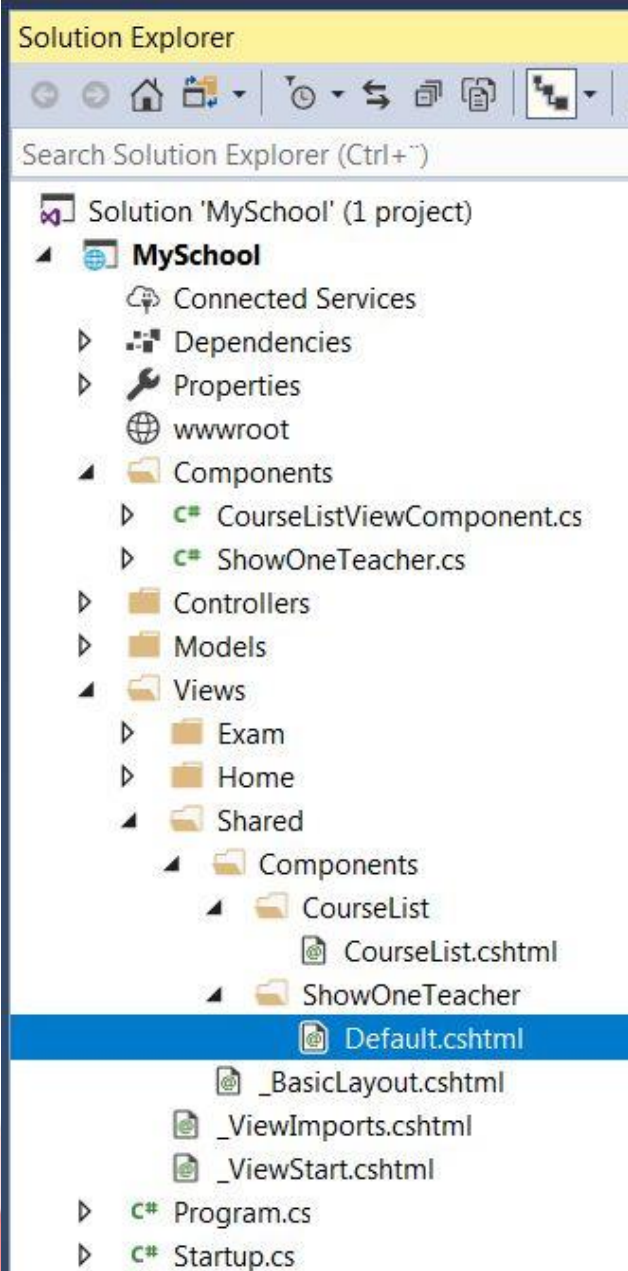
    return View();
}
```

# ShowOneTeacher - komponent

```
ShowOneTeacher.cs  + X
MySchool  MySchool.Components.ShowOneTea

1  using System.Threading.Tasks;
2  using MySchool.Models;
3  using Microsoft.AspNetCore.Mvc;
4
5  namespace MySchool.Components
6  {
7      public class ShowOneTeacher : ViewComponent
8      {
9          private ISchoolRepository repository;
10
11         public ShowOneTeacher(ISchoolRepository repo)
12         {
13             repository = repo;
14         }
15
16         public async Task<IViewComponentResult> InvokeAsync(int id)
17         {
18             var objectOfTeacher = await repository.GetTeacherDetail(id);
19             return View(objectOfTeacher);
20         }
21     }
22 }
23
```





# ShowOneTeacher - vy

```
ShowTeacher.cshtml  Deafult.cshtml + X
1  @model Teacher
2
3  <h2>Teacher Detail (från komponent</h2>
4
5  <div>
6      <p>@Model.TeacherID</p>
7      <p>@Model.Name</p>
8      <p>@Model.Code</p>
9  </div>
```



UPPSALA  
UNIVERSITET

# ShowTeacher-vy

```
@await Component.InvokeAsync("ShowOneTeacher", ViewBag.Tid)
```

```
@await Component.InvokeAsync("CourseList")
```

```
<vc:show-one-teacher id="@ViewBag.Tid"></vc:show-one-teacher>
```

```
<vc:course-list></vc:course-list>
```



# Tag Helper

```
_ViewImports.cshtml  CourseList.cshtml  CourseListViewComponent.cs
1  @using MySchool.Models
2  @addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
3  @addTagHelper *, MySchool|
```



UPPSALA  
UNIVERSITET

# Resultatet

## Teacher Detail (från component)

2

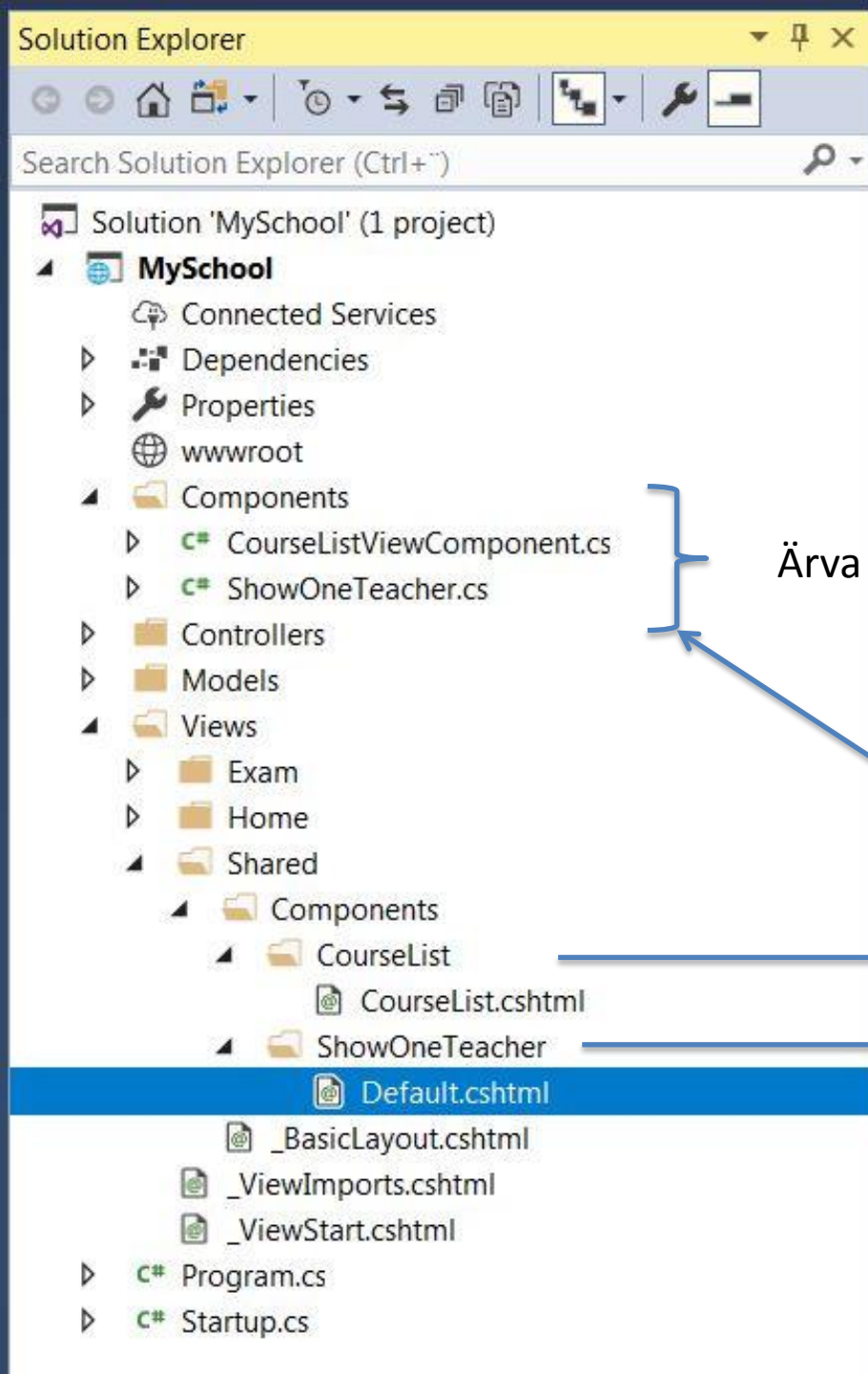
Jonas Cluedo

JC

## Course List (från component)

Choose Course





Ärva av ViewComponents

Samma namn som komponenterna



UPPSALA  
UNIVERSITET

## CourseListViewComponent

```
public IViewComponentResult Invoke()  
{  
    return View("CourseList", repository.Courses);  
}
```

## ShowOneTeacher

```
public async Task<IViewComponentResult> InvokeAsync(int id)  
{  
    var objectOfTeacher = await repository.GetTeacherDetail(id);  
    return View(objectOfTeacher);  
}
```



UPPSALA  
UNIVERSITET

# **OBS!!!!!!**

## **LÄS NÄSTA UPPGIFT NOGA**

Ni ska **INTE** skapa massor av  
Partial Views och View Component  
i nästa uppgift