C# Web – MVC

PRO - C# Web 1

DE HOGESCHOOL
MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

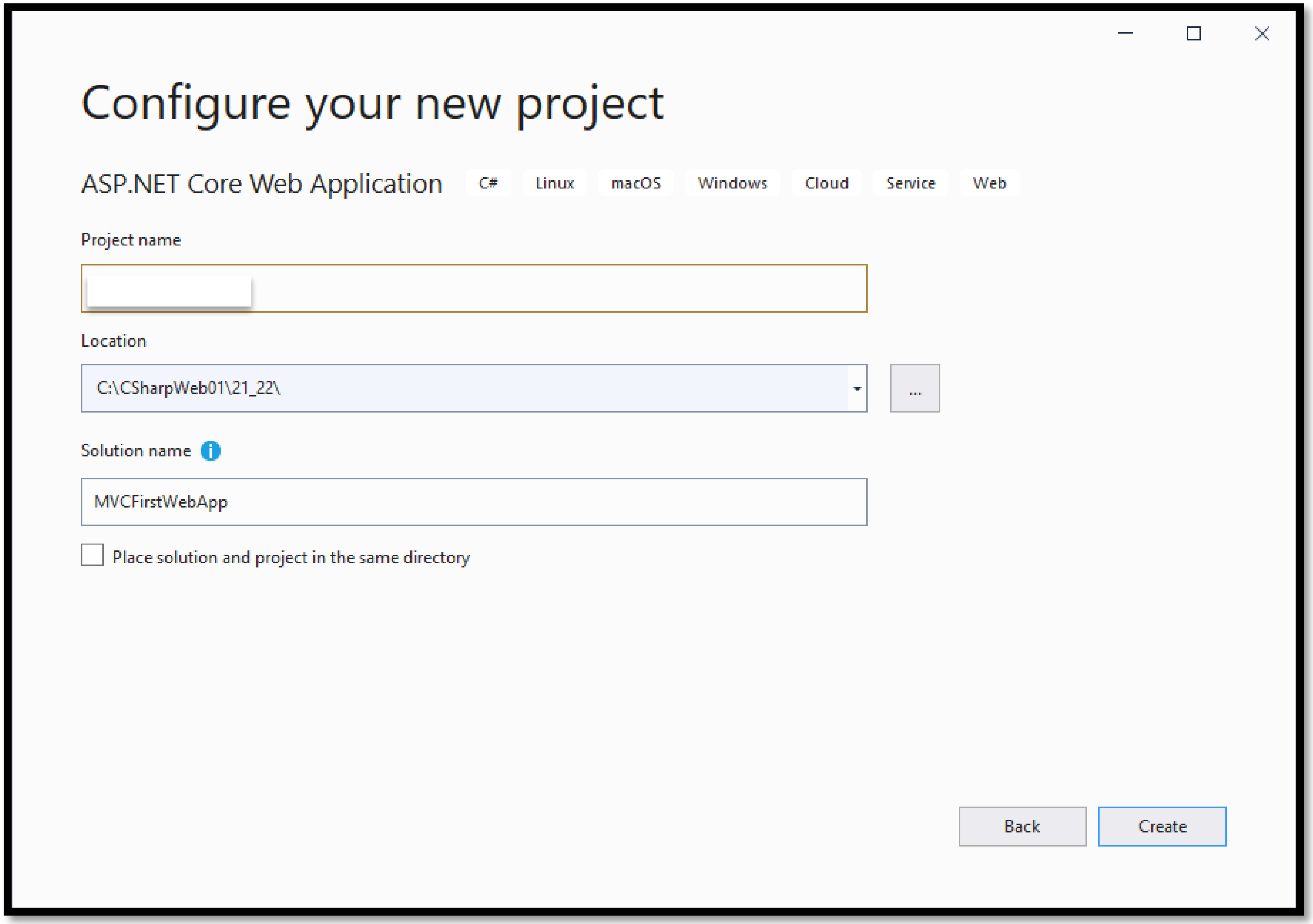
Doel

- ASP.Net Core toepassing
  - MVC Web Application
    - Models
    - Views
    - Controllers
  - Entity framework

ASP.NET CORE

MVC Web Application

MVCFifa2021



Configure your new project

ASP.NET Core Web Application

Project name

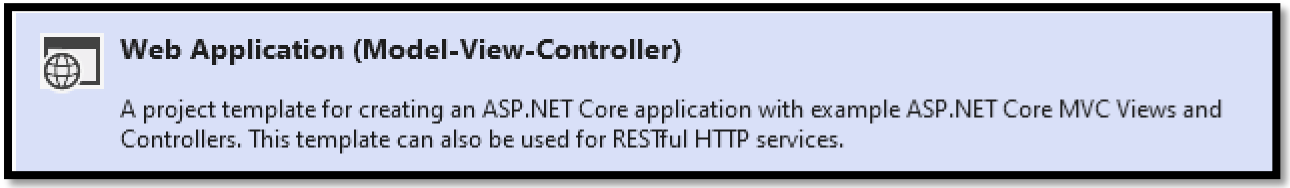
Location

C:\CSharpWeb01\21_22\

Solution name

MVCFirstWebApp

□ Place solution and project in the same directory



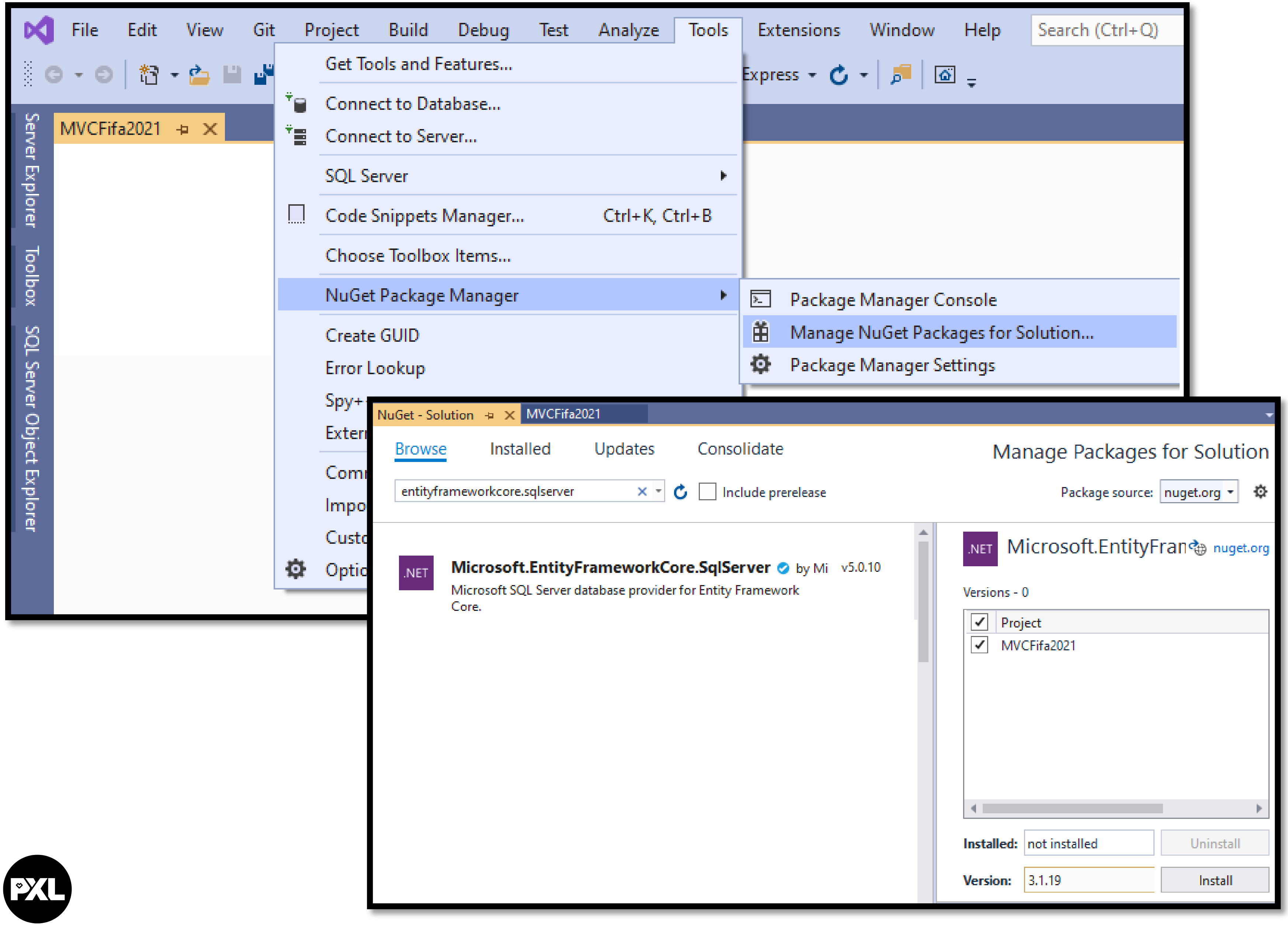
Web Application (Model-View-Controller)

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

ASP.Net Core MVC

ENTITY FRAMEWORK

Entity Framework – SQL Server – Nuget Package Manager



entityframeworkcore.sqlserver

□ Include prerelease

Package source: nuget.org

Microsoft.EntityFrameworkCore.SqlServer by Mi v5.0.10

Microsoft SQL Server database provider for Entity Framework Core.

Microsoft.EntityFran nuget.org

Versions - 0

☑	Project
☑	MVCFifa2021

Installed: not installed

Version: 3.1.19

- Create Folder Data
  - Add class ApplicationDbContext

```
namespace MVCFifa2021.Data
{
  public class ApplicationDbContext : DbContext
  {
      public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
               : base(options)
     {
     }
  }
}
```

Startup.cs

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddControllersWithViews();
    var connString = "Server = (localdb)\\mssqllocaldb;";
    connString += "Database = fifa2021;";
    connString += "Trusted_Connection = True;";
    connString += "MultipleActiveResultSets = true";
    services.AddDbContext<ApplicationDbContext>(options =>
options.UseSqlServer(connString));
}
```

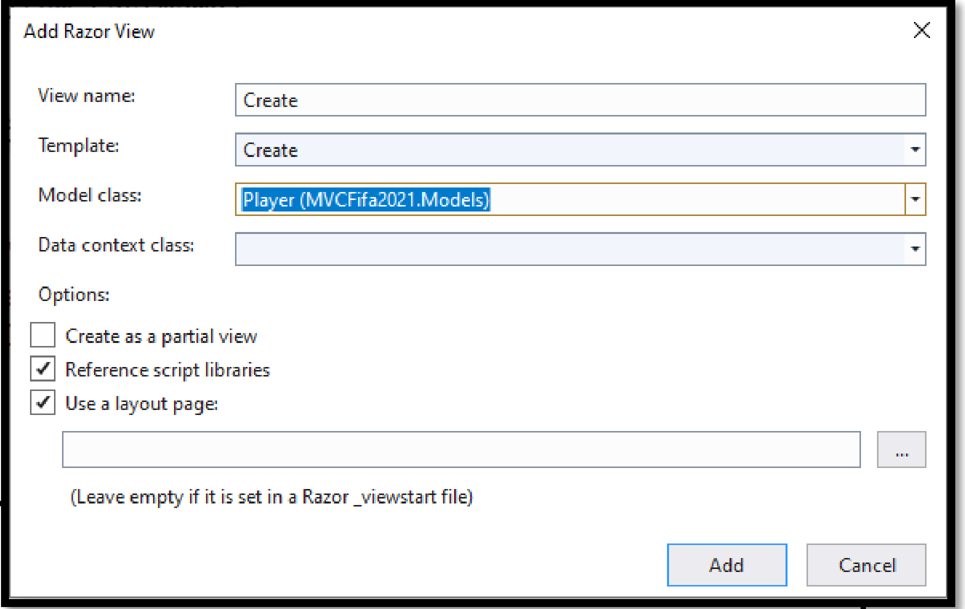
Models folder

- Add Player.cs

```
namespace MVCFifa2021.Models
{
    public class Player
    {
        public int PlayerId { get; set; }
        [Required]
        public string FirstName { get; set; }
        [Required]
        public string LastName { get; set; }
        public string ImageLink { get; set; }
    }
}
```

Controllers folder

- Add PlayerController.cs



Add Razor View

View name: Create

Template: Create

Model class: Player (MVCFifa2021.Models)

Data context class:

Options:

☐ Create as a partial view

☑ Reference script libraries

☑ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

Views folder

- Add folder Player
  - Scaffold Razor View - Index – List Template – Player Model
  - Scaffold Razor View - Create – Create Template – Player Model
  - Geen Data context class selecteren

Data folder

- Add new Table in database file - ApplicationDbContext

```
public class ApplicationDbContext : DbContext
{
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options)
     {
     }
    public DbSet<Player> Players { get; set; }
}
```

Models folder

- Add Player.cs

```
namespace MVCFifa2021.Models
{
    public class Player
    {
        public int PlayerId { get; set; }
        [Required]
        public string FirstName { get; set; }
        [Required]
        public string LastName { get; set; }
        public string ImageLink { get; set; }
    }
}
```

Dependency Injection

- PlayerController
    - Add Constructor
        - PlayerController(DbContext Service)

```
public class PlayerController : Controller
{
    ApplicationDbContext _context;
    public PlayerController(ApplicationDbContext context)
    {
        _context = context;
        _context.Database.EnsureCreated();
    }
        // … Code not shown for clarity
```

- PlayerController
  - Read data
    - Index

```
public IActionResult Index()
{
        var players = _context.Players;
        return View(players);
}
```

- PlayerController
  - Insert data
    - Create

```
[HttpGet]
public IActionResult Create()
{
    var player = new Player();
    return View(player);
}
[HttpPost]
public IActionResult Create(Player player)
{
    if (ModelState.IsValid)
    {
        AddPlayer(player);
        return RedirectToAction("Index");
    }
    return View(player);
}
private void AddPlayer(Player player)
{
    _context.Players.Add(player);
    _context.SaveChanges();
}
```

- Styling the content
    - Views/Shared
        - _Layout.cshtml

- wwwroot folder
    - Add folder images
        - Add image fifa…

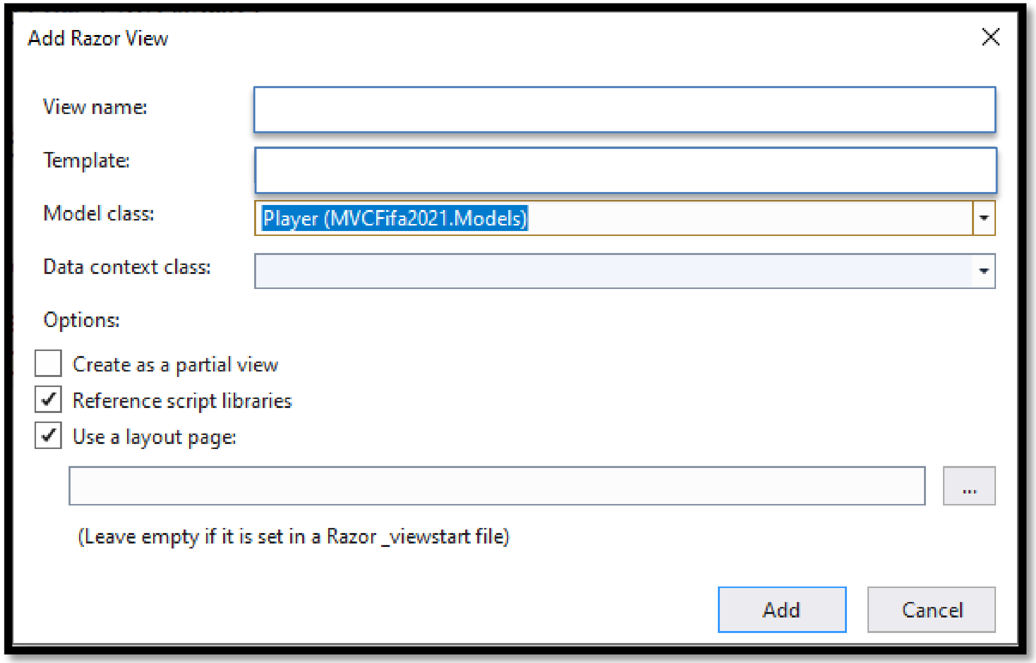
```
<header>
    <div class="text-center">
        <img src="~/images/fifa_header2.png" />
    </div>
</header>
<div class="row">
    <div class="col-1"></div>
    <div class="col-3">
        <br />
        <a asp-controller="Player" asp-
action="Index">Players</a>
    </div>
    <div class="col-8"> @RenderBody()</div>
</div>
<footer class="border-top footer text-muted">
    <div class="container">
        &copy; 2021 - MVCFifa2021
    </div>
</footer>
```

<u>Samenvatting Entity Framework</u>

- Install Nuget package
- Update Startup class
  - Service DBContext
- Create DbContext class file in Data folder
- Create Model -> Table -> DbSet
- Dependency Injection
  - Context class -> Controller
    - Context.SaveChanges()

ASP.NET CORE

MVC Web Application – Entity Framework - Oefening07
Medewerkers – Index/Create



Add Razor View

View name:

Template:

Model class: Player (MVCFifa2021.Models)

Data context class:

Options:

☐ Create as a partial view

☑ Reference script libraries

☑ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

Views folder

- Scaffold razor views in Player folder
  - Scaffold Razor View - Details – Details Template – Player Model
  - Scaffold Razor View - Edit – Edit Template – Player Model
  - Scaffold Razor View - Delete – Delete Template – Player Model
  - Geen Data context class selecteren

- PlayerController
    - Details

```
public IActionResult Details(int id)
{
        var player = _context.Players.Where(x=>x.PlayerId == id).FirstOrDefault();
        return View(player);
}
```

Views/Player/Index.cshtml

```
…
@Html.ActionLink("Edit", "Edit", new { id = item.PlayerId }) |
@Html.ActionLink("Details", "Details", new { id=item.PlayerId}) |
@Html.ActionLink("Delete", "Delete", new { id = item.PlayerId })
<br>
<a asp-action="Details" asp-route-id="item.PlayerID">Details</a>
```

- PlayerController
  - Details
    - Upgrade -> show image in details page if exists

```
private IWebHostEnvironment _environment;
```

```
public IActionResult Details(int id)
{
    var player = _context.Players.Where(x=>x.PlayerId == id).FirstOrDefault();
   var path = _environment.WebRootPath;
   var file = Path.Combine($"{path}\\images", player.ImageLink);
   var fileExist = System.IO.File.Exists(file);
   ViewBag.FileExist = fileExist;
   return View(player);
}
```

Views/Player/Details.cshtml

```
@{
    ViewData["Title"] = "Details";
    var fileExist = Convert.ToBoolean(ViewBag.FileExist);
    var imgLink = $"/images/{Model.ImageLink}";
}
@if (fileExist)
{<img src="@imgLink" width="200" />}
```

- PlayerController
    - Delete

```
[HttpGet]
public IActionResult Delete(int id)
{
    var player = _context.Players.Where(x => x.PlayerId == id).FirstOrDefault();
    return View(player);
}
[HttpPost]
public IActionResult DeletePost(int id)
{
    var player = _context.Players.Where(x => x.PlayerId == id).FirstOrDefault();
    _context.Players.Remove(player);
    _context.SaveChanges();
    return RedirectToAction("Index");
}
```

Views/Player/Delete.cshtml

```
<form asp-action="DeletePost" asp-route-id="@Model.PlayerId">
        <input type="submit" value="Delete" class="btn btn-danger" /> |
        <a asp-action="Index">Back to List</a>
</form>
```

- PlayerController
    - Update data
        - Edit

```
[HttpGet]
public IActionResult Edit(int id)
        {
var player = _context.Players.Where(x => x.PlayerId == id).FirstOrDefault();
            return View(player);
        }
[HttpPost]
public IActionResult Edit(Player player)
{
    if (ModelState.IsValid)
    {
        UpdatePlayer(player);
        return RedirectToAction("Index");
    }
    return View(player);
}
private void UpdatePlayer(Player player)
{
    _context.Players.Update(player);
    _context.SaveChanges();
}
```

ASP.NET CORE

MVC Web Application – Entity Framework - Oefening07
Medewerkers – Details/Edit/Delete