

CompanyEmployee Manager Asp.Net Core MVC

Deel 3

We starten vanaf deze Source code: <https://github.com/CSharpSyntraWest/CompaniesEmployees5>

In deze les voegen we de CRUD operaties toe om een bestaande Employee te wijzigen en een Employee te verwijderen van de database

1 Wijzigen van een bestaande Employee

We gaan de volgende nieuwe pagina maken om een Employee te kunnen updaten.

The screenshot shows a web browser window with the title bar '- CompanyEmployees.MVC'. The address bar shows the URL 'localhost:44361/employeeManager/Update/9b206e8e-cd95-...'. The browser's navigation bar includes back, forward, and refresh buttons, along with search, star, and extension icons. The application's navigation bar is dark blue and contains the text 'CompanyEmployees.MVC' followed by links for 'Home', 'Privacy', 'Employees', and 'Companies'. The main content area has a white background and features the heading 'Update Existing Employee' in a large, bold, black font. Below the heading, the form displays the following fields: 'Id :' with the value '9b206e8e-cd95-45dd-4328-08d8da3845cf', 'Name:' with a text input containing 'José', 'Age:' with a text input containing '23', 'Position:' with a text input containing 'HR manager', and 'CompanyId:' with a dropdown menu showing 'MyCompany'. A 'Save' button is located below the 'CompanyId:' field. At the bottom of the form, there is a blue link that says 'Back to Employee Listing'. The Windows taskbar is visible at the bottom of the screen, showing icons for File Explorer, PowerPoint, a shopping bag, a folder, Google Chrome, Microsoft Word, and Visual Studio Code.

CompanyEmployees.MVC Home Privacy Employees Companies

Update Existing Employee

Id : 9b206e8e-cd95-45dd-4328-08d8da3845cf

Name:

Age:

Position:

CompanyId:

[Back to Employee Listing](#)

De Update Existing Employee lijkt op de Insert Employee pagina, maar er wordt nu een EmployeeID getoond. Deze is niet aanpasbaar. De huidige waarden van de fields worden getoond.

We gaan eerst 2 Update() action method voor HttpGet en HttpPost requests toevoegen aan de EmployeeManagerController

```
public IActionResult Update(Guid id)
{
    FillCompanies();

    Employee model = _repositoryManager.Employee.GetEmployee(id, false);

    return View(model);
}
[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult Update(Employee model)
{
    FillCompanies();

    if (ModelState.IsValid)
    {
        Employee employeeToUpdate =
_repositoryManager.Employee.GetEmployee(model.Id, true);
        employeeToUpdate.Name = model.Name;
        employeeToUpdate.Position = model.Position;
        employeeToUpdate.Age = model.Age;
        employeeToUpdate.CompanyId = model.CompanyId;
        _repositoryManager.Save();
        ViewBag.Message = "Employee updated successfully";
    }
    return View(model);
}
```

De eerste methode neemt een Employee id als invoer parameter. Deze wordt meegegeven met de url wanneer er op een update linkbutton wordt geklikt vanuit de Employee List pagina.

We zoeken eerst de employee op via zijn id in de database en geven het terug gegeven Employee object door naar de View, zodat deze de fields kan invullen met de huidige waarden.

De 2^{de} methode wordt aangeroepen wanneer op de Submit(Save) button in de form van de view wordt geklikt, Deze zal eerst nagaan of de validaties ok waren en gaat dan de Employee object updaten in de database

Maak nu een razor View onder Views/EmployeeManager met naam Update.csthml

En zet hierin de volgende code:

```
@model Employee
<h2>Update Existing Employee</h2>
<h3 class="message">@ViewBag.Message</h3>

<form asp-controller="EmployeeManager" asp-action="Update" method="post">

    <table border="0">
        <tr>
            <td class="right">
                <label asp-for="Id"></label> :
            </td>
            <td>
                <input type="hidden" asp-for="Id" />
                <span>@Model.Id</span>
            </td>
        </tr>
        <tr>
            <td class="right">
                <label asp-for="Name"></label>:
            </td>
            <td>
                <input type="text" asp-for="Name" />
                <span asp-validation-for="Name" class="message"></span>
            </td>
        </tr>
        <tr>
            <td class="right">
                <label asp-for="Age"></label>:
            </td>
            <td>
                <input type="number" asp-for="Age" />
                <span asp-validation-for="Age" class="message"></span>
            </td>
        </tr>
        <tr>
            <td class="right">
                <label asp-for="Position"></label>:
            </td>
            <td>
                <input type="text" asp-for="Position" />
                <span asp-validation-for="Position" class="message"></span>
            </td>
        </tr>
        <tr>
            <td class="right">
                <label asp-for="CompanyId"></label>:
            </td>
            <td>
                <select asp-for="CompanyId" asp-items="@ViewBag.Companies">
                    <option value="">Please select</option>
                </select>
                <span asp-validation-for="CompanyId" class="message"></span>
            </td>
        </tr>
        <tr>
            <td colspan="2">
```

```

        <button type="submit">Save</button>
      </td>
    </tr>
  </table>
</form>
<br />
<br />
<a asp-controller="employeemanager" asp-action="Index">Back to Employee Listing</a>

```

Test nu de Update Existing Employee pagina. Verander een paar waarden en klik op Save

The screenshot shows a web browser window with the address bar displaying 'localhost:44361/employeeManager/Update/9b206e8e-cd95-...'. The page title is 'CompanyEmployees.MVC'. The navigation bar includes links for 'Home', 'Privacy', 'Employees', and 'Companies'. The main content area is titled 'Update Existing Employee' and contains the following form:

Id : 9b206e8e-cd95-45dd-4328-08d8da3845cf

Name:

Age:

Position:

CompanyId:

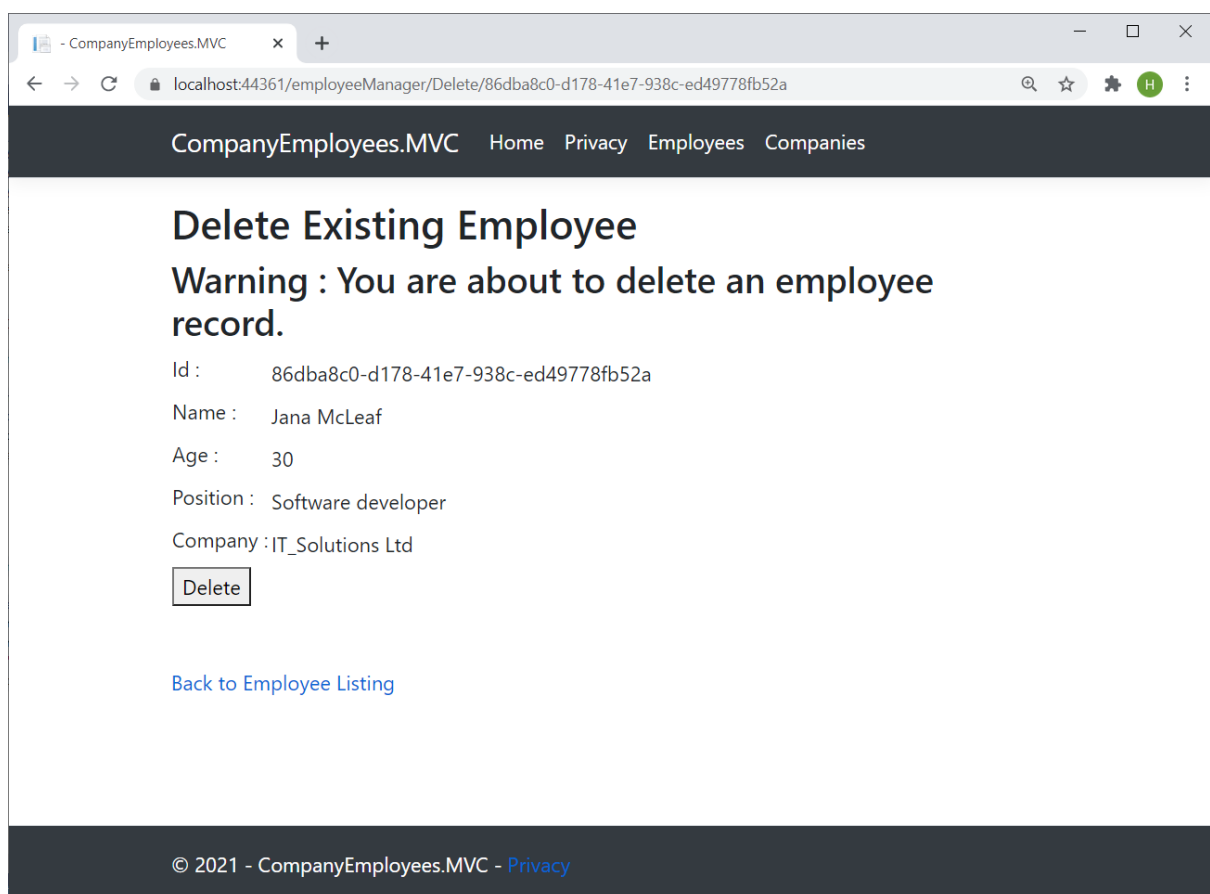
[Back to Employee Listing](#)

Als laatste taak gaan we een Delete Existing Employee pagina en action methods voorzien

2 Verwijderen van een bestaande Employee

We gaan de volgende nieuwe pagina maken om een Employee te kunnen verwijderen.

We gaan een Delete-Confirmatiepagina tonen en daarin moet de gebruiker nogmaals op de knop delete drukken om de Employee uit de database te verwijderen.



Open **EmployeeManagerController.cs** en voeg de volgende action methods toe:

```
[ActionName("Delete")]
public IActionResult ConfirmDelete(Guid id)
{
    Employee model = _repositoryManager.Employee.GetEmployee(id, false);
    Company company =
_repositoryManager.Company.GetCompany(model.CompanyId, false);
    ViewBag.Company = company;
    return View(model);
}

[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult Delete(Guid id)
{
    Employee employeeToDelete =
_repositoryManager.Employee.GetEmployee(id, false);
    _repositoryManager.Employee.DeleteEmployee(employeeToDelete);
    _repositoryManager.Save();
    TempData["Message"] = "Employee deleted successfully";
    return RedirectToAction(nameof(Index));
}
```

Boven de **HttpGet** action Method **ConfirmDelete** staat er een attribuut

```
[ActionName("Delete")]
```

Dankzij deze kan de methode een andere naam hebben dan meegegeven in de url. Deze methode kan dan worden aangeroepen met de url
/EmployeeManager/Delete/xxx-xxx-xxxx-xxx

Waarbij **xxx-xxx-xxxx-xxx** een geldig Guid id is voor de employee

De **Delete** methode met attribuut **HttpPost** zal worden aangeroepen wanneer er op de delete button wordt geklikt vanuit de **Delete Existing Employee** Pagina.

Voeg een Razor View toe onder Views/EmployeeManager met als naam Delete.cshtml en zet de volgende inhoud:

```
@model Employee

<h2>Delete Existing Employee</h2>

<h3 class="message">
    Warning : You are about to delete an employee record.
</h3>

<form asp-controller="EmployeeManager" asp-action="Delete" method="post">

    <input type="hidden" asp-for="Id" />

    <table border="0">
        <tr>
            <td class="right">
                <label asp-for="Id"></label> :
            </td>
            <td>
                @Model.Id
            </td>
        </tr>
        <tr>
            <td class="right">
                <label asp-for="Name"></label> :
            </td>
            <td>
                @Model.Name
            </td>
        </tr>
        <tr>
            <td class="right">
                <label asp-for="Age"></label> :
            </td>
            <td>
                @Model.Age
            </td>
        </tr>
        <tr>
            <td class="right">
                <label asp-for="Position"></label> :
            </td>
            <td>
                @Model.Position
            </td>
        </tr>
        <tr>
            <td class="right">
                <label asp-for="Company"></label> :
            </td>
            <td>
                @ViewBag.Company.Name
            </td>
        </tr>
        <tr>
            <td colspan="2">
                <button type="submit">Delete</button>
            </td>
        </tr>
    </table>
</form>
```



```

    </table>
</form>

<br />
<br />

<a asp-controller="EmployeeManager" asp-action="Index">Back to Employee Listing</a>

```

Test nu de Delete pagina en functionaliteit:

The screenshot shows a web browser window with the title 'CompanyEmployees.MVC'. The address bar displays 'localhost:44361/EmployeeManager'. The page content includes a navigation bar with links: 'CompanyEmployees.MVC', 'Home', 'Privacy', 'Employees', and 'Companies'. The main heading is 'List of Employees'. Below the heading, a message states 'Employee deleted successfully' with a blue 'Insert' link. A table lists three employees:

Name	Position	Age	Actions
Jana McLeaf	Software developer	30	Update Delete
Kane Miller	Administrator	35	Update Delete
Jos De Klos	Developer	45	Update Delete

The footer of the page contains the text: '© 2021 - CompanyEmployees.MVC - [Privacy](#)'.