

CompanyEmployee Manager Asp.Net Core MVC

Deel 2

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

In dit document voegen we de functionaliteit toe om een Employee te creëren.

1 Creatie van een nieuwe Employee

We gaan de volgende nieuwe pagina maken om een Employee te kunnen aanmaken:

CompaniesEmployees.Mvc

Insert New Employee

Name :

Age :

Position :

CompanyId :

- Please select
- Admin_Solutions Ltd
- IT_Solutions Ltd

[Back to Employee Listing](#)

© 2021 - CompaniesEmployees.Mvc - [Privacy](#)

Wanneer we op de insert button link klikken, gaan we een andere pagina tonen in de browser waar we de gegevens voor een nieuwe employee kunnen invullen en deze kunnen toevoegen aan de database

De volgende gegevens hebben we nodig om een nieuwe Employee aan te maken:

Name, Age , Company en Position.

Employee ID is een Identity field in de database die wordt toegekend in de database, deze moeten we dus niet meegeven om een nieuwe Employee aan

te maken. De company zal in een dropdown lijst worden gekozen. Deze lijst van companies wordt uit de database gehaald.

2 Toevoegen van Insert action methods aan de EmployeeManagerController class

Voordat we de Insert Employee pagina aanmaken, gaan we 2 nieuwe action methods en een Helper methode toevoegen aan de

EmployeeManagerController

Action method Insert() voor HttpGet request: Deze zal bij het laden van de pagina een lijst van Companies uit de database halen, die dan in de pagina in een dropdown lijst zal worden getoond. Hiervoor gaan we een **Helper methode FillCompanies()** gebruiken.

Action method Insert() voor HttpPost request: Wanneer er op de form submit button wordt gedrukt, worden de waarden van de form fields doorgestuurd naar de server, die dan in deze methode in de database worden bewaard.

Open het bestand **EmployeeManagerController.cs**

En voeg de volgende code toe aan de class:

```
private void FillCompanies()
{
    List<SelectListItem> companies =
        (from c in _repositoryManager.Company.GetAllCompanies(false)
         orderby c.Name ascending
         select new SelectListItem() { Text = c.Name, Value = c.Id.ToString()
}).ToList();

    ViewBag.Companies = companies;
}

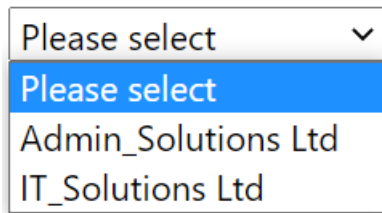
public IActionResult Insert()
{
    FillCompanies();
    return View();
}

[HttpPost]
public IActionResult Insert(Employee model)
{
    if (ModelState.IsValid)
    {
        _repositoryManager.Employee.CreateEmployeeForCompany(model.CompanyId,
model);
        _repositoryManager.Save();
        ViewBag.Message = "Employee inserted";
    }
    return View(model);
}
```

1. De helper methode **FillCompanies()**

Deze declareert een List van SelectListItem objecten. SelectListItem is een class die gedefinieerd is in de *Microsoft.AspNetCore.Mvc.Rendering* namespace.

en worden gebruikt om items in een dropdown lijst (<select> element van HTML) te kunnen tonen. We willen deze dropdown lijst tonen in een dropdown lijst op onze Insert EmployeeManager Web pagina:



In de **FillCompanies()** methode worden alle companies uit de database gehaald.

Merk op dat we de **Text en Value Properties van de SelectListItem** objecten worden ingevuld door de Name field van de Company objecten.

ToList() converteert de IQueryable naar een List van SelectListItem objecten

De lijst van companies zal worden doorgegeven naar de view om **een <select> html element** in te vullen. Om de data transfer te vergemakkelijken, wordt een **ViewBag** object gebruikt:

```
ViewBag.Companies = companies;
```

De ViewBag is een dynamisch type en laat toe om dynamisch waarden toe te voegen en uit te lezen via de object property syntax.

Hier worden de companies bijgehouden in het Companies property van ViewBag.

2. De action method **Insert()** voor **HttpGet requests**

Deze **Insert()** methode wordt aangeroepen wanneer er wordt geklikt op de Insert hyperlink van de EmployeeManager Index pagina.

Deze methode roept de FillCompanies() helper methode aan om de dropdown list van companies te kunnen invullen. (Deze companies worden via de ViewBag doorgegeven naar de view)

Daarna wordt de View voor de Insert Employee getoond.

3. De action method **Insert()** voor **HttpPost** requests

Deze action methode wordt aangeroepen wanneer de Insert Employee form wordt ge-submit via de POST method. Daarom wordt het attribuut **[HttpPost]** op de lijn boven de methode geplaatst.

De **Insert()** methode neemt als parameter een object aan van het type Employee. Wanneer de form ge-submit wordt, vult ASP.NET Core automatisch de form fields values in een Employee object. Dit wordt **model binding** genoemd. ASP.NET Core zal de form field names matchen op de property names van het Employee object. Bv de form field met naam Name zal worden gematched met de property Name van Employee.

Via data annotaties zijn bepaalde attributen aan de properties van de Employee class gelinkt. Bv **[Required]** en **[MaxLength(20)]**. Deze worden gebruikt als validatie criteria. Indien een input veld niet is ingevuld voor een property data is aangeduid als **[Required]**, zal er een standaard foutboodschap worden getoond aan de gebruiker.

Om te verifiëren of de ModelState geldig is, wordt **IsValid property** gecontroleerd in de methode **Insert()**.

ModelState.IsValid property zal true teruggeven indien alle validaties succesvol waren. In dit geval wordt een nieuwe Employee naar de database geschreven. Dit gebeurt dmv de RepositoryManager.

We gaan nu een **Razor View** voor onze Insert Employee toevoegen.

3 Toevoegen van Insert Employee Razor View

Voeg een nieuwe Razor View toe onder de folder **Views/EmployeeManager**

met naam **Insert.cshtml**

Zet de volgende code in het bestand:

@model Employee

```
<h2>Insert New Employee</h2>
```

```
<h3 class="message">@ViewBag.Message</h3>
```

```
<form asp-controller="employeeManager" asp-action="Insert" method="post">
  <table border="0">
    <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>

```

Er zijn verschillende **Razor Tag Helpers** die in de code worden gebruikt:

Label, span, textarea, input, select en a (anchor)

Deze Tag helpers genereren de overeenkomstige HTML-tags

Het **attribuut** **asp-for** wordt gebruikt om de form fields te linken aan de properties van de model (hier een Employee object).

De **span-Tag Helpers** worden gebruikt om validatie-foutboodschappen te tonen.

De **select-Tag helper** wordt gebruikt om een drop-down lijst van companies te tonen. Via het attribuut **asp-items="@ViewBag.Companies"**.

De ViewBag.Companies property was in de controller gezet op een `List<SelectListItem>` met de namen van de companies. Deze worden dan omgezet door de Tag Helper naar `<option>` HTML tags.

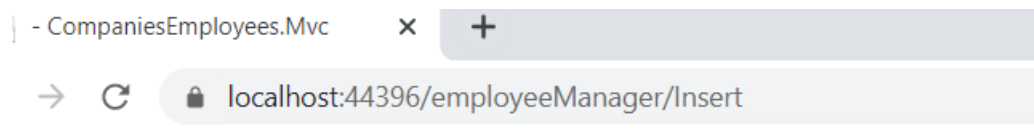
Via deze laatste lijn kan er worden teruggekeerd naar de Index van EmployeeManager:

```

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

```

Test nu eens de Insert Employee pagina:



CompaniesEmployees.Mvc Home Privacy

Insert New Employee

Name :

Age :

Position :

CompanyId :

Please select

Please select

Admin_Solutions Ltd

IT_Solutions Ltd

[Back to Employee Listing](#)

Indien verkeerde invoer werd gegeven, worden er **validatie**-foutboodschappen getoond wanneer op de Submit button (Save) wordt geklikt:

- CompaniesEmployees.Mvc x +

→ ↻ localhost:44396/employeeManager/Insert

CompaniesEmployees.Mvc Home Privacy

Insert New Employee

Name : Employee name is a required field.

Age : The value " " is invalid.

Position : Position is a required field.

CompanyId : The value " " is invalid.

[Back to Employee Listing](#)

Vul nu correct de invoer velden in en klik op Save button. Deze zal de form submitten met een post request naar de server.

CompaniesEmployees.Mvc

localhost:44396/employeeManager/Insert

CompaniesEmployees.Mvc

Insert New Employee

Name :

Joke

Age :

25

Position :

Analist

CompanyId :

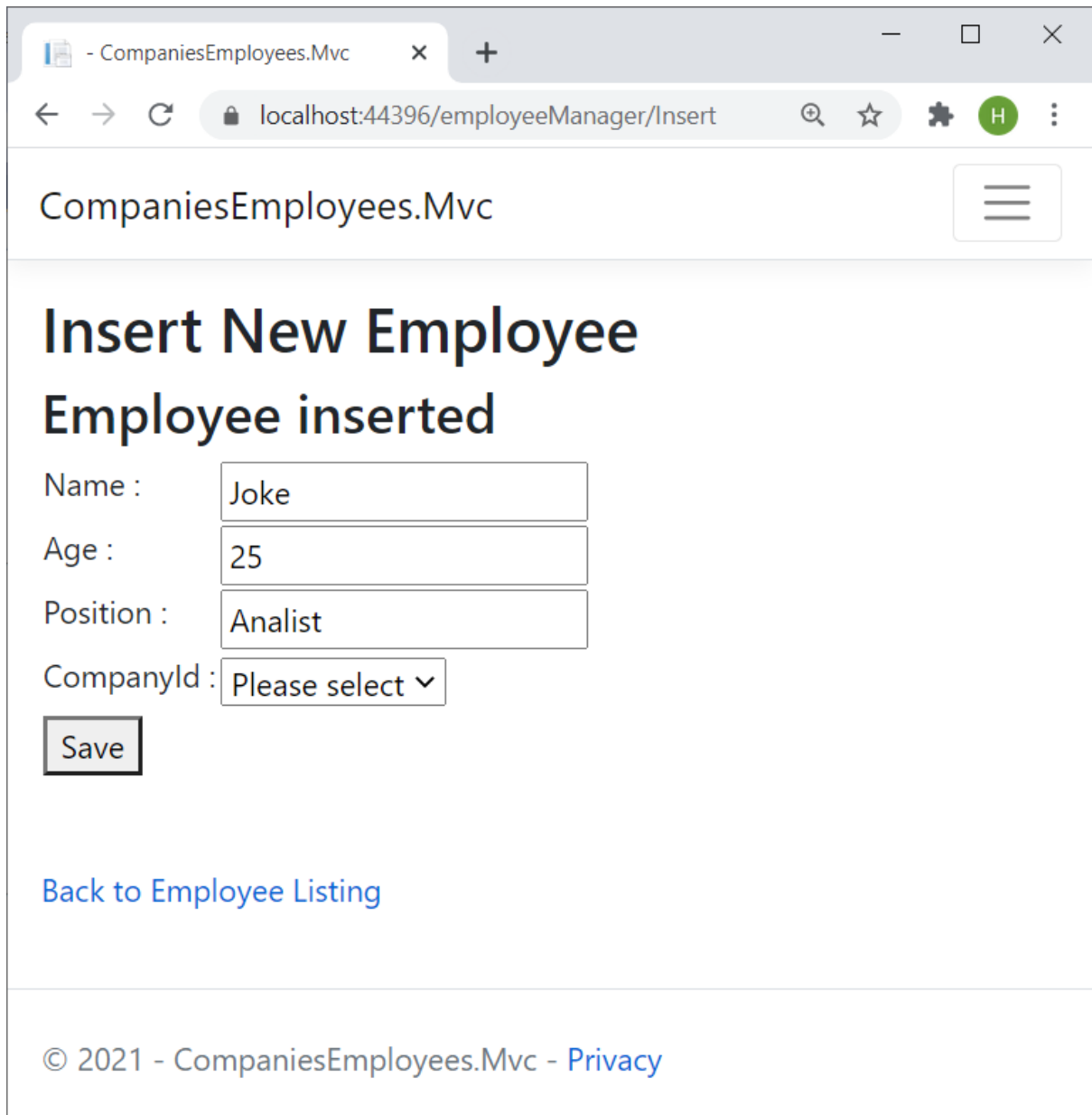
Admin_Solutions Ltd ▾

Save

[Back to Employee Listing](#)

© 2021 - CompaniesEmployees.Mvc - [Privacy](#)

Je krijgt de boodschap (via `ViewBag.Message`): “Employee inserted” te zien:



The screenshot shows a web browser window with the title "CompaniesEmployees.Mvc". The address bar displays "localhost:44396/employeeManager/Insert". The page content includes a header "CompaniesEmployees.Mvc" with a hamburger menu icon. The main heading is "Insert New Employee". Below this, a message "Employee inserted" is displayed. The form contains four input fields: "Name :" with the value "Joke", "Age :" with the value "25", "Position :" with the value "Analist", and "CompanyId :" with a dropdown menu showing "Please select". A "Save" button is located below the form fields. At the bottom of the form area, there is a link "Back to Employee Listing". The footer of the page shows "© 2021 - CompaniesEmployees.Mvc - Privacy".

CompaniesEmployees.Mvc

Insert New Employee

Employee inserted

Name :

Age :

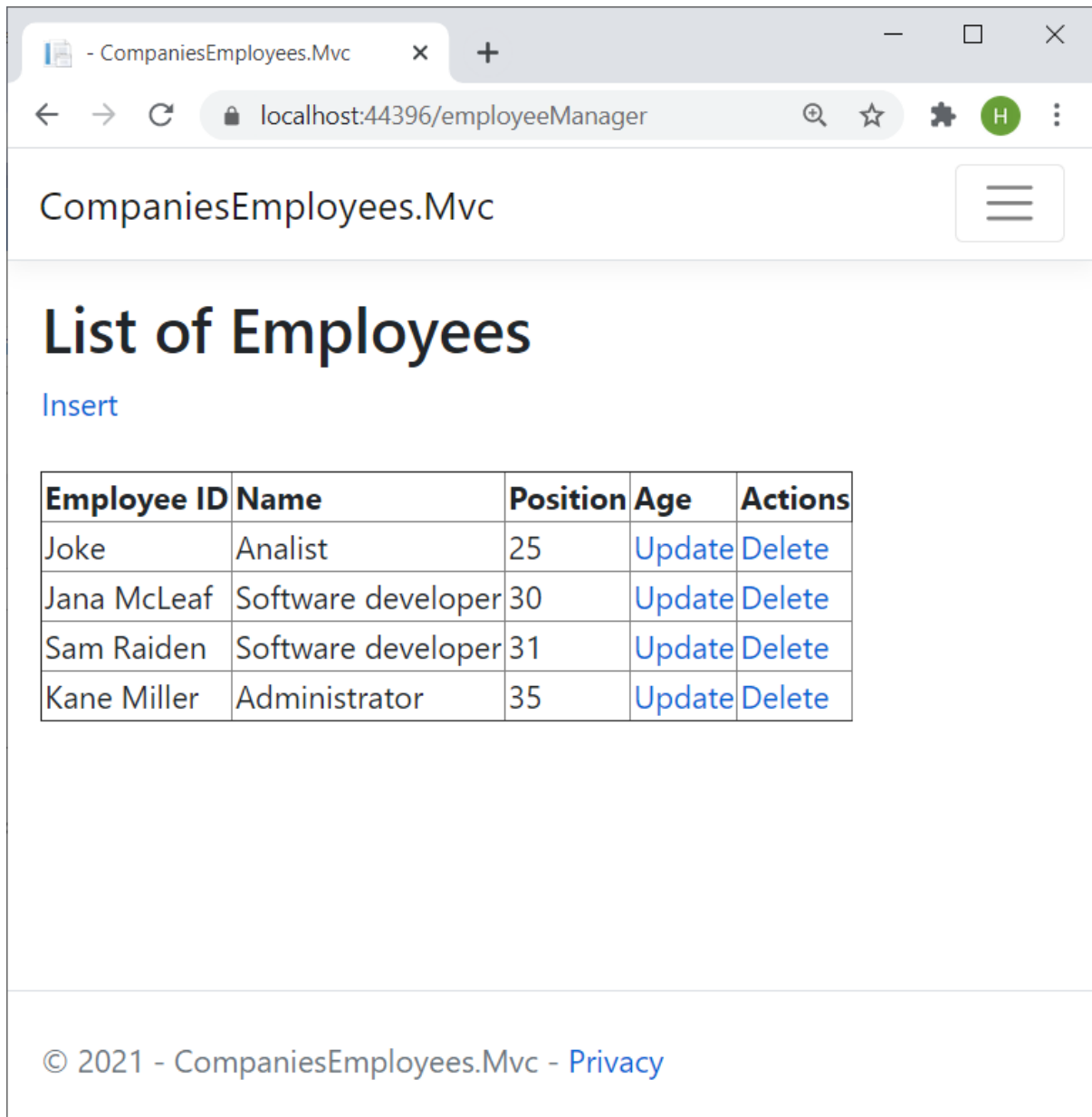
Position :

CompanyId :

[Back to Employee Listing](#)

© 2021 - CompaniesEmployees.Mvc - [Privacy](#)

Klik op “Back to Employee Listing” en je ziet dat employee Joke is toegevoegd:



The screenshot shows a web browser window with the address bar displaying 'localhost:44396/employeeManager'. The page title is 'CompaniesEmployees.Mvc'. The main heading is 'List of Employees'. Below the heading is a blue link labeled 'Insert'. A table lists four employees: Joke, Jana McLeaf, Sam Raiden, and Kane Miller. Each row includes columns for Employee ID, Name, Position, Age, and Actions (Update, Delete). The footer shows '© 2021 - CompaniesEmployees.Mvc - Privacy'.

CompaniesEmployees.Mvc

List of Employees

[Insert](#)

Employee ID	Name	Position	Age	Actions
Joke	Analist	25	Update	Delete
Jana McLeaf	Software developer	30	Update	Delete
Sam Raiden	Software developer	31	Update	Delete
Kane Miller	Administrator	35	Update	Delete

© 2021 - CompaniesEmployees.Mvc - [Privacy](#)