

<https://www.halvorsen.blog>

ASP.NET Core Basics



ASP.NET Core

Get Data from Database

Hans-Petter Halvorsen

Contents

This Tutorial are for rookies when it comes to Web Programming and ASP.NET Core

- ASP.NET Core
- SQL Server
- Practical Examples: Retrieving Data from Database
 - Example 1 (Part 1) – Show Company Information for a given Company
 - Example 2 (Part 2)– Show All Employees within the Company inside a Table
 - Example 3 (Part 3)– Show specific Database Information based on User Input
 - Example 4 (Part 4)– Show Employee Information based on selected Customer

ASP.NET Core Example

There you see the Final Solution. We will take it step by step and create this ASP.NET Web Application in Iterations

GetDatafromDatabase Home

Company

Webpage1

Company Name:

USN



Company Information:

Company Name: **USN**

Website: www.usn.no

Show Employees



© 2021 - GetDatafromDatabase - [Privacy](#)

GetDatafromDatabase Home

Employees

Webpage2

Employee Information for Customer: USN

EmployeeId	First Name	Last Name
1	Elvis	Presley
2	Knut	Hamsun

© 2021 - GetDatafromDatabase - [Privacy](#)



ASP.NET Core

Hans-Petter Halvorsen

[Table of Contents](#)

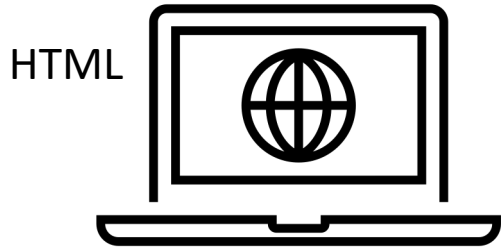
ASP.NET Core

- ASP.NET Core is an open-source web framework, created by Microsoft, for building web applications and web services
- You create ASP.NET Core Web Applications using Visual Studio and the C# Programming language
- ASP.NET Resources:

<https://www.halvorsen.blog/documents/programming/web/aspnet>

Web Fundamentals

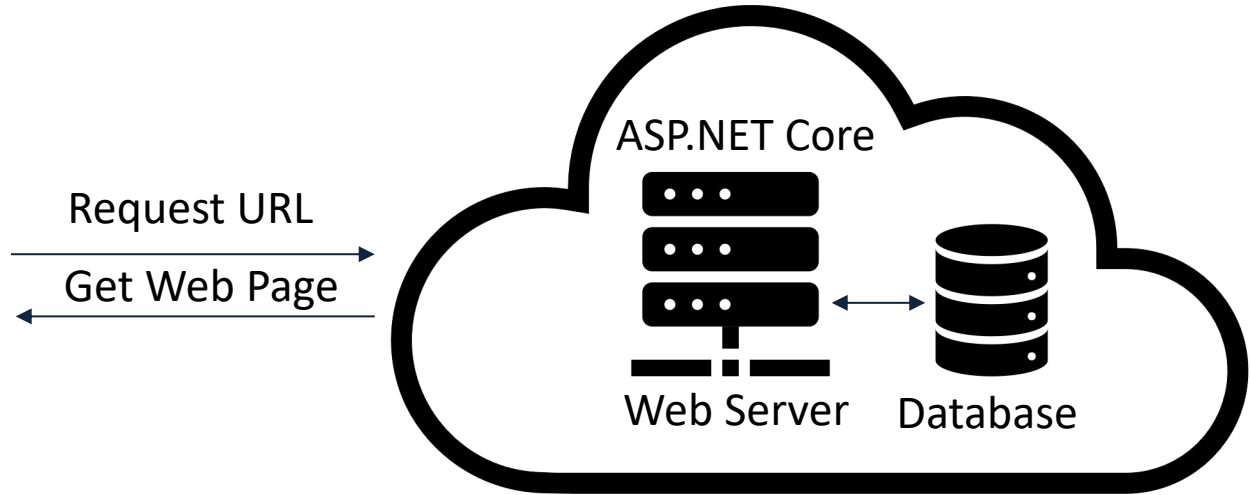
Client-side



Local Computer
with Web Browser

A pure HTML File is sent to
your local Web Browser

Server-side



The Cloud/Internet

Server-side Code (C#) is executed on the Server

Web Fundamentals

HTML

CSS

JavaScript

**Server-side
Framework**

Client-side

HTML is the standard markup language for creating Web pages

CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.

JavaScript is the programming language of the Web. JavaScript is probably the most used programming language in the world.

ASP.NET Core (C#), PHP or similar



SQL Server

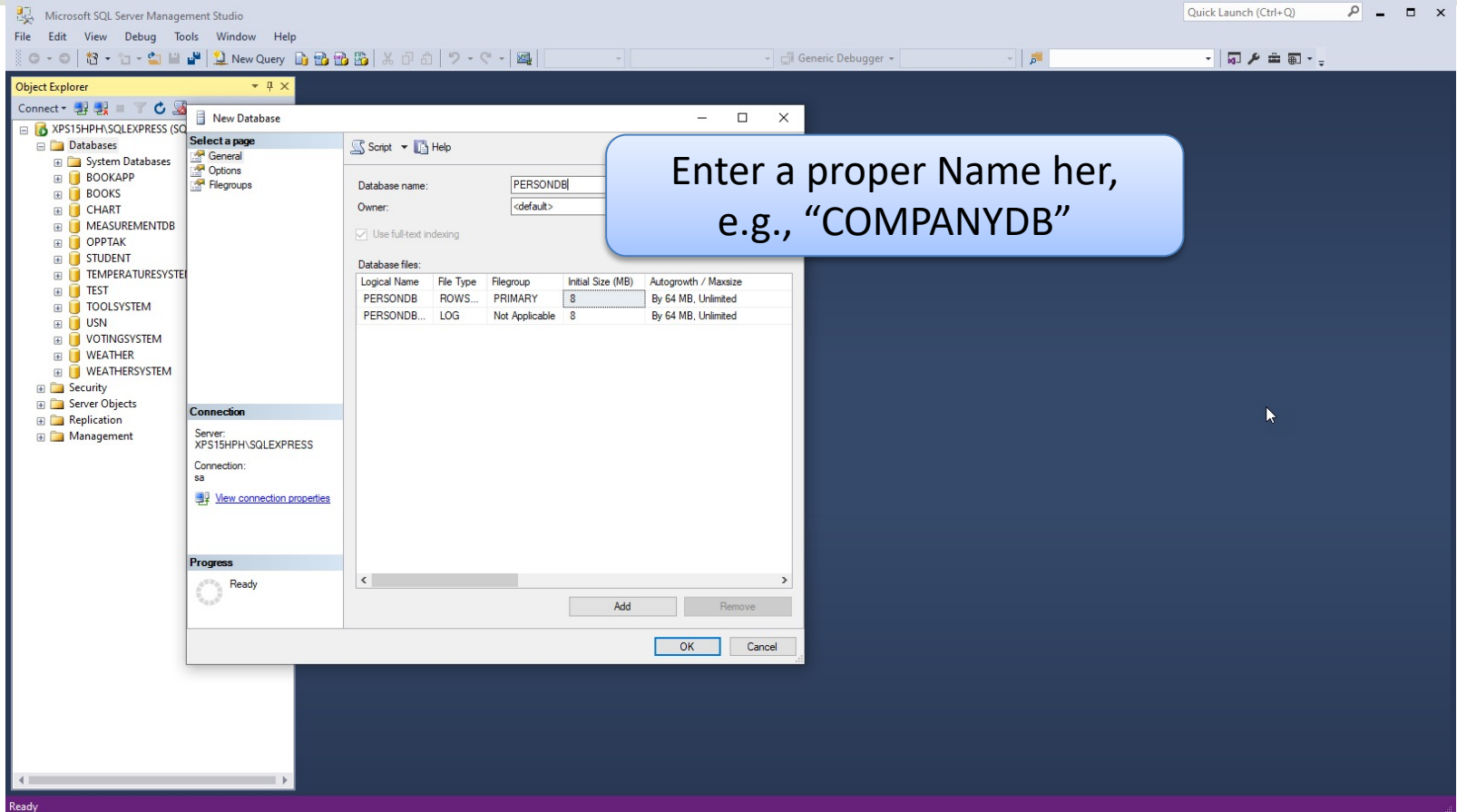
Hans-Petter Halvorsen

[Table of Contents](#)

SQL Server

- SQL Server Express
 - Free version of SQL Server that has all we need for the examples in this Tutorial
- SQL Server Express consist of 2 parts (separate installation packages):
 - SQL Server Express
 - SQL Server Management Studio (SSMS) – This software can be used to create Databases, create Tables, Insert/Retrieve or Modify Data, etc.
- SQL Server Express Installation:
<https://youtu.be/hhhggAlUYo8>

Create Database



Database Table

```
CREATE TABLE COMPANY
(
    CompanyId          int    PRIMARY KEY IDENTITY (1,1),
    CompanyName        varchar(50) NOT NULL,
    WebSite             varchar(100) NOT NULL,
)
GO
```

This is just a basic example, here you can add more columns like Address, Phone, etc.

Create Table

The screenshot displays the Microsoft SQL Server Management Studio interface. The title bar indicates the current session is 'SQLQuery1.sql - XPS15HPH\SQLEXPRESS.PERSONDB (sa (53))* - Microsoft SQL Server Management Studio'. The menu bar includes File, Edit, View, Project, Debug, Tools, Window, and Help. The toolbar contains icons for file operations, query execution, and debugging. The Object Explorer on the left shows the database structure for 'XPS15HPH\SQLEXPRESS (SQL Server 13.0.1742 - ...)', with 'PERSONDB' expanded to show 'Tables' and 'dbo.PERSON'. The main query editor displays the following SQL code:

```
CREATE TABLE PERSON
(
    PersonId int NOT NULL IDENTITY (1,1),
    FirstName varchar(50) NOT NULL,
    LastName varchar(50) NOT NULL
)
GO
```

The Messages pane at the bottom shows the execution result: 'Command(s) completed successfully.' The status bar at the bottom indicates 'Query executed successfully.' and provides session details: 'XPS15HPH\SQLEXPRESS (13.0 RTM) | sa (53) | PERSONDB | 00:00:00 | 0 rows'.

Create Data

XPS15HPH\SQLEXPRESS.COMPANYDB - dbo.COMPANY - Microsoft SQL Server Management Studio

File Edit View Project Debug Query Designer Tools Window Help

New Query

Object Explorer

Connect

XPS15HPH\SQLEXPRESS (SQL Server 13.0.1742 - sa)

- Databases
 - System Databases
 - BOOKAPP
 - BOOKS
 - CHART
 - MEASUREMENTDB
 - OPPTAK
 - PERSONDATABASE
 - STUDENT
 - TEMPERATURESYSTEM
 - TEST
 - TOOLSYSTEM
 - USN
 - VOTINGSYSTEM
 - WEATHER
 - WEATHERSYSTEM
 - COMPANYDB
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - dbo.COMPANY
 - dbo.EMPLOYEE
 - Views

XPS15HPH\SQLEXP...B - dbo.COMPANY

	CompanyId	CompanyName	WebSite
▶	1	USN	www.usn.no
	2	Yara	www.yara.com
*	NULL	NULL	NULL



Practical Examples



Part 1

ASP.NET Core

Get Data from Database

Hans-Petter Halvorsen

[Table of Contents](#)



Code Example 1

Show Company Information for a given Company

ASP.NET Core Example

[GetDatafromDatabase](#) [Home](#) [Privacy](#)

Company

Company Information:

Company Name: **USN**

Website: www.usn.no

NuGet

The screenshot displays the Visual Studio IDE with the NuGet Package Manager window open. The 'Browse' tab is selected, showing a search for 'SQL'. The package 'Microsoft.Data.SqlClient' (version 3.0.1) is highlighted with a red box. The right pane provides details for this package, including its description, common types, and version information. The bottom pane shows an error list with two CS errors related to missing namespace 'Data' and 'SqlConnection'.

NuGet Package Manager: SaveDataDatabase

Package source: nuget.org

Microsoft.Data.SqlClient by Microsoft, 99M downloads, v3.0.1

Provides the data provider for SQL Server. These classes provide access to...

Description

Provides the data provider for SQL Server. These classes provide access to versions of SQL Server and encapsulate database-specific protocols, including tabular data stream (TDS)

Commonly Used Types:

- Microsoft.Data.SqlClient.SqlConnection
- Microsoft.Data.SqlClient.SqlException
- Microsoft.Data.SqlClient.SqlParameter
- Microsoft.Data.SqlClient.SqlDataReader
- Microsoft.Data.SqlClient.SqlCommand
- Microsoft.Data.SqlClient.SqlTransaction
- Microsoft.Data.SqlClient.SqlParameterCollection
- Microsoft.Data.SqlClient.SqlClientFactory

When using NuGet 3.x this package requires at least version 3.4.

Version: 3.0.1

Author(s): Microsoft

License: MIT

Date published: Friday, September 24, 2021

Error List

Project	File	Line	Supp...
SaveData...	Index.csh...	2	Active
SaveData...	Index.csh...	25	Active

```
using Microsoft.AspNetCore.Mvc.RazorPages;  
using Microsoft.Data.SqlClient;
```

```
namespace GetDatafromDatabase.Pages
```

```
{
```

```
    public class IndexModel : PageModel
```

```
    {
```

```
        public string companyName;
```

```
        public string website;
```

```
        public void OnGet()
```

```
        {
```

```
            string connectionString = "Data Source=xxx\\SQLEXPRESS;Initial Catalog=COMPANYDB;Integrated Security=True";
```

```
            SqlConnection con = new SqlConnection(connectionString);
```

```
            con.Open();
```

```
            string sqlQuery = "select CompanyName, WebSite from COMPANY where CompanyId=1";
```

```
            SqlCommand cmd = new SqlCommand(sqlQuery, con);
```

```
            SqlDataReader dr = cmd.ExecuteReader();
```

```
            if (dr.Read())
```

```
            {
```

```
                companyName = dr["CompanyName"].ToString();
```

```
                website = dr["WebSite"].ToString();
```

```
            }
```

```
            con.Close();
```

```
        }
```

```
    }
```

```
}
```

When a user request a Webpage,
the code inside the **OnGet()** method
are executed on the Server before
the code is sent to the Client

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div>

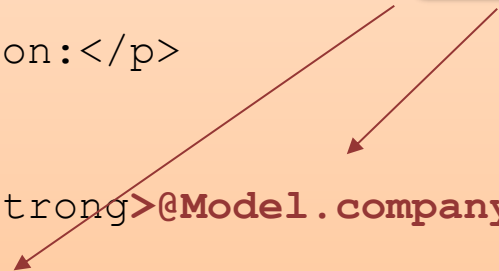
    <h1>Company</h1>

    <p>Company Information:</p>

    <p>
        Company Name: <strong>@Model.companyName</strong>
        <br />
        Website: @Model.website
    </p>

</div>
```

Razor Code (starts with "@")



The Razor Code (@) are executed on the Server and pure HTML is sent to the Client



Create Method

Hans-Petter Halvorsen

[Table of Contents](#)

```
using Microsoft.AspNetCore.Mvc.RazorPages;  
using Microsoft.Data.SqlClient;
```

```
namespace GetDatafromDatabase.Pages  
{
```

```
    public class IndexModel : PageModel  
    {
```

```
        public string companyName;  
        public string website;
```

```
        public void OnGet()  
        {  
            GetData();  
        }
```

```
        void GetData()  
        {
```

```
            string connectionString = "Data Source=XPS15HPH\\SQLEXPRESS;Initial Catalog=COMPANYDB;Integrated Security=True";  
            SqlConnection con = new SqlConnection(connectionString);  
            con.Open();
```

```
            string sqlQuery = "select CompanyName, WebSite from COMPANY where CompanyId=1";
```

```
            SqlCommand cmd = new SqlCommand(sqlQuery, con);
```

```
            SqlDataReader dr = cmd.ExecuteReader();
```

```
            if (dr.Read())  
            {  
                companyName = dr["CompanyName"].ToString();  
                website = dr["WebSite"].ToString();  
            }
```

```
            con.Close();
```

```
        }
```

```
    }
```

```
}
```

A separate Method
"GetData()" has been created



Create Class

```
using Microsoft.Data.SqlClient;

namespace GetDatafromDatabase.Models
{
    public class Company
    {
        public string companyName { get; set; }
        public string website { get; set; }

        public Company GetData()
        {
            string connectionString = "Data Source=xxx;Initial Catalog=COMPANYDB;Integrated Security=True";
            SqlConnection con = new SqlConnection(connectionString);
            con.Open();

            string sqlQuery = "select CompanyName, WebSite from COMPANY where CompanyId=1";

            SqlCommand cmd = new SqlCommand(sqlQuery, con);

            SqlDataReader dr = cmd.ExecuteReader();

            Company company = new Company();

            if (dr.Read())
            {
                company.companyName = dr["CompanyName"].ToString();
                company.website = dr["WebSite"].ToString();
            }
            con.Close();
            return company;
        }
    }
}
```

A separate Class “Company”
has been created in a new
Folder named “Models”


```
using Microsoft.AspNetCore.Mvc.RazorPages;  
using GetDatafromDatabase.Models;
```

```
namespace GetDatafromDatabase.Pages  
{  
    public class IndexModel : PageModel  
    {  
        public Company companyData = new Company();  
  
        public void OnGet()  
        {  
            Company company = new Company();  
            companyData = company.GetData();  
        }  
    }  
}
```

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div>

    <h1>Company</h1>

    <p>Company Information:</p>

    <p>
        Company Name: <strong>@Model.companyData.companyName</strong>
        <br />
        Website: @Model.companyData.website
    </p>

</div>
```



Part 2

ASP.NET Core

Get Data from Database

Hans-Petter Halvorsen

[Table of Contents](#)

ASP.NET Core Example

There you see the Final Solution. We will take it step by step and create this ASP.NET Web Application in Iterations

GetDatafromDatabase Home

Company

Webpage1

Company Name:

USN



Company Information:

Company Name: **USN**

Website: www.usn.no

Show Employees



© 2021 - GetDatafromDatabase - [Privacy](#)

GetDatafromDatabase Home

Employees

Webpage2

Employee Information for Customer: USN

EmployeeId	First Name	Last Name
1	Elvis	Presley
2	Knut	Hamsun

© 2021 - GetDatafromDatabase - [Privacy](#)

Contents

This Tutorial are for rookies when it comes to Web Programming and ASP.NET Core

- ASP.NET Core
- SQL Server
- Practical Examples: Retrieving Data from Database
 - Example 1 (Part 1) – Show Company Information for a given Company
 - **Example 2 (Part 2) – Show All Employees within the Company inside a Table**
 - Example 3 (Part 3) – Show specific Database Information based on User Input
 - Example 4 (Part 4) – Show Employee Information based on selected Customer



Code Example 2

Show All Employees within the Company inside a Table

ASP.NET Core Example

- In Addition to Showing Company Information, we also want to show Employee Information
- We create a second Web page where we put all Employees inside a Table

ASP.NET Core Example

[GetDatafromDatabase](#)

[Home](#)

[Employees](#)

Company

Company Information:

Company Name: **USN**

Website: www.usn.no

[GetDatafromDatabase](#)

[Home](#)

[Employees](#)

Employees

Employee Information:

EmployeeId	First Name	Last Name
1	Elvis	Presley
2	Knut	Hamsun

Updated Database

```
CREATE TABLE COMPANY
(
    CompanyId          int    PRIMARY KEY IDENTITY (1,1),
    CompanyName        varchar(50)    NOT NULL,
    WebSite             varchar(100)   NOT NULL,
)
GO
```

```
CREATE TABLE EMPLOYEE
(
    EmployeeId int    PRIMARY KEY IDENTITY (1,1),
    FirstName   varchar(50)    NOT NULL,
    LastName    varchar(50)    NOT NULL,
    CompanyId   int NOT NULL FOREIGN KEY REFERENCES [COMPANY] ([CompanyId])
)
GO
```

Create Data

Microsoft SQL Server Management Studio interface showing the Object Explorer and a query result.

Object Explorer: XPS15HPH\SQLEXPRESS (SQL Server 13.0.1742 - sa)

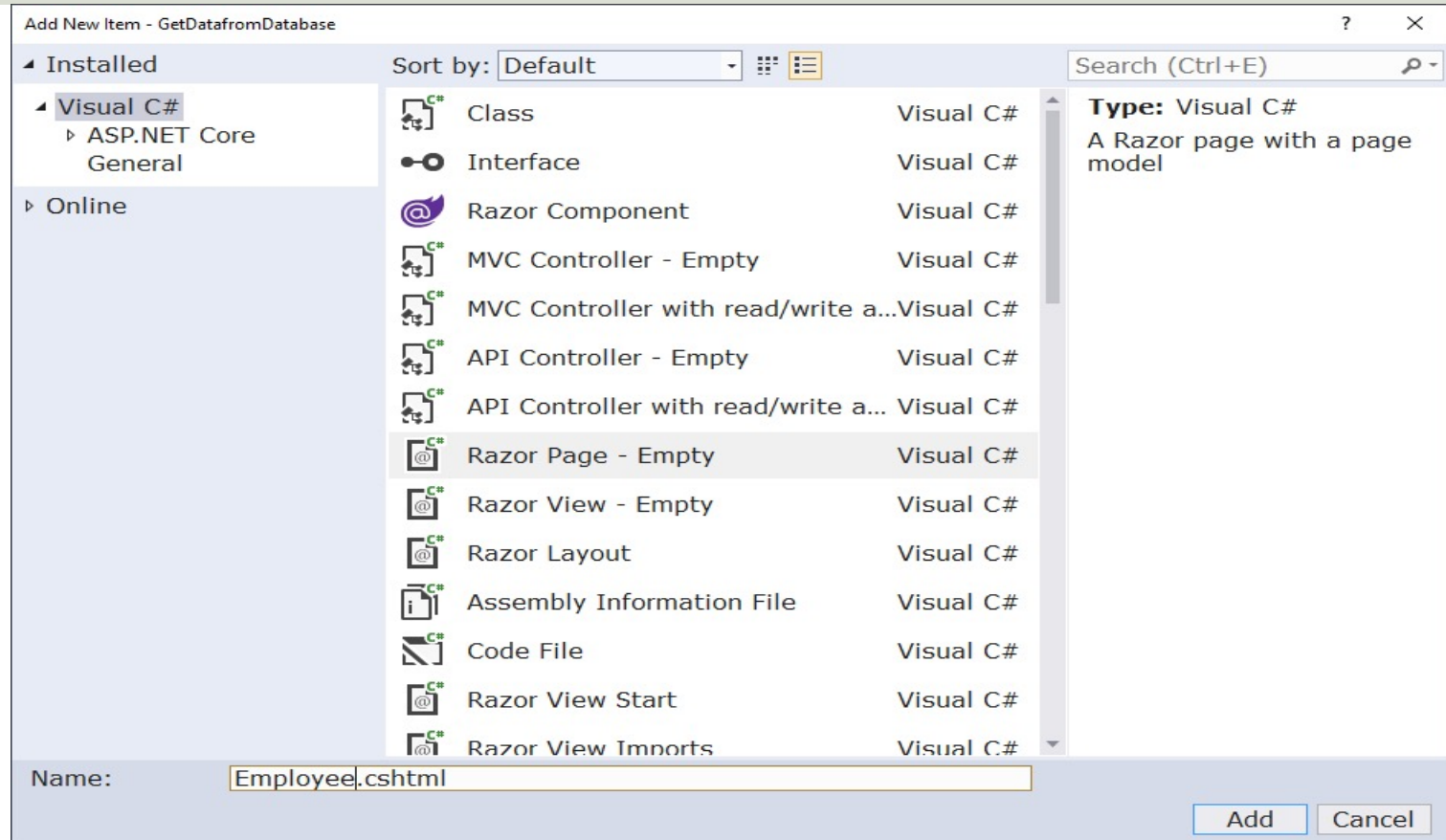
- Databases
 - System Databases
 - BOOKAPP
 - BOOKS
 - CHART
 - MEASUREMENTDB
 - OPPTAK
 - PERSONDATABASE
 - STUDENT
 - TEMPERATURESYSTEM
 - TEST
 - TOOLSYSTEM
 - USN
 - VOTINGSYSTEM
 - WEATHER
 - WEATHERSYSTEM
 - COMPANYDB
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - dbo.COMPANY
 - dbo.EMPLOYEE
 - Views
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
 - Security
 - Server Objects
 - Replication

Query Result: XPS15HPH\SQLEXP...B - dbo.EMPLOYEE

	EmployeeId	FirstName	LastName	CompanyId
▶	1	Elvis	Presley	1
	2	Knut	Hamsun	1
	3	Morten	Harket	2
*	NULL	NULL	NULL	NULL

Ready

Create New Web Page



Add Web Page to Menu

The screenshot shows the Visual Studio IDE with the following components:

- File Explorer:** Shows the project structure. The file `_Layout.cshtml` is highlighted under the `Pages` folder.
- Code Editor:** Displays the content of `_Layout.cshtml`. The code is an ASP.NET Razor view that defines the layout of the application, including a navigation bar and a footer. The `asp-page="/Employee">Employees</a>` line is highlighted with a red box.
- Properties Window:** Shows the properties of the selected file, `_Layout.cshtml`.
- Output Window:** Shows the output of the build process, indicating that the application was built successfully.

The code in the `_Layout.cshtml` file is as follows:

```
<!-- _Layout.cshtml -->
<html>
<head>
</head>
<body>
<header>
<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
<div class="container">
<a class="navbar-brand" asp-area="" asp-page="/Index">GetDatafromDatabase</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-controls="navbarSupport"
aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
<ul class="navbar-nav flex-grow-1">
<li class="nav-item">
<a class="nav-link text-dark" asp-area="" asp-page="/Index">Home</a>
</li>
<li class="nav-item">
<a class="nav-link text-dark" asp-area="" asp-page="/Employee">Employees</a>
</li>
</ul>
</div>
</div>
</nav>
</header>
<div class="container">
<main role="main" class="pb-3">
@RenderBody()
</main>
</div>
<div class="border-top footer text-muted">
<div class="container">
&copy; 2021 - GetDatafromDatabase - <a asp-area="" asp-page="/Privacy">Privacy</a>
</div>
</div>
</body>
</html>
<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
<script src="~/js/site.js" asp-append-version="true"></script>
```

Add Web Page to Menu

[GetDatafromDatabase](#)

[Home](#)

[Employees](#)

Company

Company Information:

Company Name: **USN**

Website: www.usn.no

Employee.cs

```
using System;
using System.Collections.Generic;
using Microsoft.Data.SqlClient;

namespace GetDatafromDatabase.Models
{
    public class Employee
    {
        public int employeeId { get; set; }
        public string firstName { get; set; }
        public string lastName { get; set; }

        public List<Employee> GetEmployeeData()
        {
            string connectionString = "Data Source=XPS15HPH\\SQLEXPRESS;Initial Catalog=COMPANYDB;Integrated Security=True";
            SqlConnection con = new SqlConnection(connectionString);
            con.Open();

            string sqlQuery = "select EmployeeId, FirstName, LastName from EMPLOYEE where CompanyId=1";

            SqlCommand cmd = new SqlCommand(sqlQuery, con);

            SqlDataReader dr = cmd.ExecuteReader();

            List<Employee> employeeList = new List<Employee>();

            while (dr.Read())
            {
                Employee employee = new Employee();

                employee.employeeId = Convert.ToInt32(dr["EmployeeId"]);
                employee.firstName = dr["FirstName"].ToString();
                employee.lastName = dr["LastName"].ToString();

                employeeList.Add(employee);
            }
            con.Close();
            return employeeList;
        }
    }
}
```

```
using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc.RazorPages;
using GetDatafromDatabase.Models;

namespace GetDatafromDatabase.Pages
{
    public class EmployeeModel : PageModel
    {
        public List<Employee> employeeList = new List<Employee>();

        public void OnGet()
        {
            Employee employee = new Employee();
            employeeList = employee.GetEmployeeData();
        }
    }
}
```

Employee.cshtml

```
@page
@model GetDatafromDatabase.Pages.EmployeeModel
@{
}
<div>
    <h1>Employees</h1>
    <p>Employee Information:</p>

    <table class="table">
        <thead>
            <tr>
                <th>EmployeeId</th>
                <th>First Name</th>
                <th>Last Name</th>
            </tr>
        </thead>
        <tbody>
            @foreach (var employee in Model.employeeList)
            {
                <tr>
                    <td> @employee.employeeId</td>
                    <td> @employee.firstName</td>
                    <td> @employee.lastName</td>
                </tr>
            }
        </tbody>
    </table>
</div>
```




Part 3

ASP.NET Core

Get Data from Database

Hans-Petter Halvorsen

[Table of Contents](#)

ASP.NET Core Example

There you see the Final Solution. We will take it step by step and create this ASP.NET Web Application in Iterations

GetDatafromDatabase Home

Company

Webpage1

Company Name:

USN



Company Information:

Company Name: **USN**

Website: www.usn.no

Show Employees



© 2021 - GetDatafromDatabase - [Privacy](#)

GetDatafromDatabase Home

Employees

Webpage2

Employee Information for Customer: USN

EmployeeId	First Name	Last Name
1	Elvis	Presley
2	Knut	Hamsun

© 2021 - GetDatafromDatabase - [Privacy](#)

Contents

This Tutorial are for rookies when it comes to Web Programming and ASP.NET Core

- ASP.NET Core
- SQL Server
- Practical Examples: Retrieving Data from Database
 - Example 1 (Part 1) – Show Company Information for a given Company
 - Example 2 (Part 2) – Show All Employees within the Company inside a Table
 - Example 3 (Part 3) – **Show specific Database Information based on User Input**
 - Example 4 (Part 4) – Show Employee Information based on selected Customer



Code Example 3

Show specific Database Information based on User Input

ASP.NET Core Example

- Here we will add a **Textbox** on the webpage where the user can enter the Company Name
- Then by clicking on a **Button** the Company Information for the specific Company will be presented

ASP.NET Core Example

GetDatafromDatabase [Home](#) [Employees](#)

Company

Company Name:

OK



© 2021 - GetDatafromDatabase - [Privacy](#)

GetDatafromDatabase [Home](#) [Employees](#)

Company

Company Name:

OK

Company Information:

Company Name: **USN**

Website: www.usn.no

© 2021 - GetDatafromDatabase - [Privacy](#)

```
using Microsoft.Data.SqlClient;

namespace GetDatafromDatabase.Models
{
    public class Company
    {
        public string companyName { get; set; }
        public string website { get; set; }

        public Company GetCompanyData(string companyName)
        {
            string connectionString = "Data Source=xxx\\SQLEXPRESS;Initial Catalog=COMPANYDB;Integrated Security=True";
            SqlConnection con = new SqlConnection(connectionString);
            con.Open();

            string sqlQuery = "select CompanyName, WebSite from COMPANY where CompanyName='" + companyName + "'";

            SqlCommand cmd = new SqlCommand(sqlQuery, con);

            SqlDataReader dr = cmd.ExecuteReader();

            Company company = new Company();

            if (dr.Read())
            {
                company.companyName = dr["CompanyName"].ToString();
                company.website = dr["WebSite"].ToString();
            }
            con.Close();
            return company;
        }
    }
}
```

```
using Microsoft.AspNetCore.Mvc.RazorPages;
using GetDatafromDatabase.Models;

namespace GetDatafromDatabase.Pages
{
    public class IndexModel : PageModel
    {
        public Company companyData = new Company();

        public void OnGet()
        {

        }

        public void OnPost()
        {
            string companyName = Request.Form["CompanyName"];

            Company company = new Company();
            companyData = company.GetCompanyData(companyName);
        }
    }
}
```



```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div>

    <h1>Company</h1>

    <form method="post">

        Company Name:
        <br />
        <input name="CompanyName" type="text" class="form-control input-lg" autofocus required />
        <br />

        <input id="OKButton" type="submit" value="OK" class="btn btn-info" />

    </form>
    <br />

    @if (@Model.companyData.companyName != null)
    {
        <p>Company Information:</p>

        <p>
            Company Name: <strong>@Model.companyData.companyName</strong>
            <br />
            Website: @Model.companyData.website
        </p>
    }

</div>
```



Use SQL Parameters

Improve Code Robustness and avoid SQL Injection

Hans-Petter Halvorsen

[Table of Contents](#)

Company.cs

```
using System.Data;
using Microsoft.Data.SqlClient;

namespace GetDatafromDatabase.Models
{
    public class Company
    {
        public string companyName { get; set; }
        public string website { get; set; }

        public Company GetCompanyData(string companyName)
        {
            string connectionString = "Data Source=XPS15HPH\\SQLEXPRESS;Initial Catalog=COMPANYDB;Integrated Security=True";
            SqlConnection con = new SqlConnection(connectionString);

            string sqlQuery = "select CompanyName, WebSite from COMPANY where CompanyName=@companyname";

            con.Open();
            SqlCommand cmd = new SqlCommand(sqlQuery, con);

            var companyNameParameter = new SqlParameter("companyname", SqlDbType.VarChar);
            companyNameParameter.Value = companyName;
            cmd.Parameters.Add(companyNameParameter);

            SqlDataReader dr = cmd.ExecuteReader();

            Company company = new Company();
            if (dr.Read())
            {
                company.companyName = dr["CompanyName"].ToString();
                company.website = dr["WebSite"].ToString();
            }
            con.Close();
            return company;
        }
    }
}
```



Code Example 3b

Show specific Database Information based on User Input

ASP.NET Core Example

Here we will add a “dropdown” List where the user can select the Company

GetDatafromDatabase Home Employees

Company

Company Name:

Select Company

OK



GetDatafromDatabase Home Employees

Company

Company Name:

Select Company ▼

OK

Company Information:

Company Name: **USN**

Website: www.usn.no

```
using System;  
using System.Data;  
using System.Collections.Generic;  
using Microsoft.Data.SqlClient;
```

```
namespace GetDatafromDatabase.Models  
{
```

```
    public class Company
```

```
    {
```

```
        public int CompanyId { get; set; }
```

```
        public string CompanyName { get; set; }
```

```
        public string Website { get; set; }
```

```
        string connectionString = "Data Source=xxx;Initial Catalog=COMPANYDB;Integrated Security=True";
```

```
        public Company GetCompanyData(int companyId)
```

```
        {
```

```
            ..
```

```
        }
```

```
        public List<Company> GetCompanyList()
```

```
        {
```

```
            ..
```

```
        }
```

```
    }
```

```
}
```

GetCompanyList ()

```
public List<Company> GetCompanyList()
```

```
{
    SqlConnection con = new SqlConnection(connectionString);
    con.Open();

    string sqlQuery = "select CompanyId, CompanyName, Website from COMPANY order by
    CompanyName";

    SqlCommand cmd = new SqlCommand(sqlQuery, con);

    SqlDataReader dr = cmd.ExecuteReader();

    List<Company> companyList = new List<Company>();

    while (dr.Read())
    {
        Company company = new Company();

        company.CompanyId = Convert.ToInt32(dr["CompanyId"]);
        company.CompanyName = dr["CompanyName"].ToString();
        company.Website = dr["Website"].ToString();

        companyList.Add(company);
    }
    con.Close();
    return companyList;
}
```

GetCompanyData()

```
public Company GetCompanyData(int companyId)
```

```
{  
    SqlConnection con = new SqlConnection(connectionString);  
  
    string sqlQuery = "select CompanyId, CompanyName, WebSite from COMPANY where  
    CompanyId=@CompanyId";  
  
    con.Open();  
  
    SqlCommand cmd = new SqlCommand(sqlQuery, con);  
  
    var companyIdParameter = new SqlParameter("CompanyId", SqlDbType.Ints);  
    companyIdParameter.Value = companyId;  
    cmd.Parameters.Add(companyIdParameter);  
  
    SqlDataReader dr = cmd.ExecuteReader();  
  
    Company company = new Company();  
  
    if (dr.Read())  
    {  
        company.CompanyId = Convert.ToInt32(dr["CompanyId"]);  
        company.CompanyName = dr["CompanyName"].ToString();  
        company.Website = dr["WebSite"].ToString();  
    }  
    con.Close();  
    return company;  
}
```


Index.cshtml.cs

```
using System;
using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc.RazorPages;
using GetDatafromDatabase.Models;

namespace GetDatafromDatabase.Pages
{
    public class IndexModel : PageModel
    {
        public List<Company> companyList = new List<Company>();
        public Company companyData = new Company();

        public void OnGet()
        {
            CompanyList();
        }

        public void OnPost()
        {
            CompanyList();
            GetCompany();
        }

        void CompanyList()
        {
            Company company = new Company();
            companyList = company.GetCompanyList();
        }

        void GetCompany()
        {
            int companyId = Convert.ToInt32(Request.Form["SelectedCompany"]);

            Company company = new Company();
            companyData = company.GetCompanyData(companyId);
        }
    }
}
```

Index.cshtml

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div>

    <h1>Company</h1>

    <form method="post">

        Company Name:
        <br />

        <select name="SelectedCompany" class="form-control input-lg">

            <option value="0">Select Company</option>

            @foreach (var company in Model.companyList)
            {
                <option value="@company.CompanyId">@company.CompanyName</option>
            }

        </select>

        <input id="OKButton" type="submit" value="OK" class="btn btn-info" />

    </form>
    <br />

    @if (@Model.companyData.CompanyName != null)
    {
        <p>Company Information:</p>

        <p>
            Company Name: <strong>@Model.companyData.CompanyName</strong>
            <br />
            Website: @Model.companyData.Website
        </p>
    }
</div>
```



Improved Webpage

ASP.NET Core Example

Here we will add a “dropdown” List where the user can select the Company

GetDatafromDatabase Home Employees

Company

Company Name:

Select Company

OK



GetDatafromDatabase Home Employees

Company

Company Name:

USN

OK

Company Information:

Company Name: **USN**

Website: www.usn.no

The Company Name stays selected after the Postback

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div>

    <h1>Company</h1>
    <form method="post">

        Company Name:
        <br />

        <select name="SelectedCompany" class="form-control input-lg">

            <option value="0">Select Company</option>

            @foreach (var company in Model.companyList)
            {
                @if (Model.companyData.CompanyId == @company.CompanyId)
                {
                    <option value="@company.CompanyId" selected>@company.CompanyName</option>
                }
                else
                {
                    <option value="@company.CompanyId">@company.CompanyName</option>
                }
            }

        </select>

        <input id="OKButton" type="submit" value="OK" class="btn btn-info" />

    </form>
    <br />

    @if (@Model.companyData.CompanyName != null)
    {
        <p>Company Information:</p>

        <p>
            Company Name: <strong>@Model.companyData.CompanyName</strong>
            <br />
            Website: @Model.companyData.Website
        </p>
    }
</div>
```



Improved Webpage

ASP.NET Core Example

GetDatafromDatabase Home Employees

Company

Company Name:

Select Company

OK Button has
been removed



© 2021 - GetDatafromDatabase - [Privacy](#)

GetDatafromDatabase Home Employees

Company

Company Name:

USN

Company Information:

Company Name: **USN**

Website: www.usn.no

© 2021 - GetDatafromDatabase - [Privacy](#)

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div>

    <h1>Company</h1>

    <form method="post">

        Company Name:
        <br />

        <select name="SelectedCompany" class="form-control input-lg"
            onchange="this.form.submit()" >

            <option value="0">Select Company</option>

            @foreach (var company in Model.companyList)
            {

                @if (Model.companyData.CompanyId == @company.CompanyId)
                {
                    <option value="@company.CompanyId" selected>@company.CompanyName</option>
                }
                else
                {
                    <option value="@company.CompanyId">@company.CompanyName</option>
                }

            }

        </select>

    </form>
    <br />

    @if (@Model.companyData.CompanyName != null)
    {
        <p>Company Information:</p>

        <p>
            Company Name: <strong>@Model.companyData.CompanyName</strong>
            <br />
            Website: @Model.companyData.Website
        </p>
    }
</div>
```




Part 4

ASP.NET Core

Get Data from Database

Hans-Petter Halvorsen

[Table of Contents](#)

ASP.NET Core Example

There you see the Final Solution. We will take it step by step and create this ASP.NET Web Application in Iterations

GetDatafromDatabase Home

Company

Webpage1

Company Name:

USN



Company Information:

Company Name: **USN**

Website: www.usn.no

Show Employees



© 2021 - GetDatafromDatabase - [Privacy](#)

GetDatafromDatabase Home

Employees

Webpage2

Employee Information for Customer: USN

EmployeeId	First Name	Last Name
1	Elvis	Presley
2	Knut	Hamsun

© 2021 - GetDatafromDatabase - [Privacy](#)

Contents

This Tutorial are for rookies when it comes to Web Programming and ASP.NET Core

- ASP.NET Core
- SQL Server
- Practical Examples: Retrieving Data from Database
 - Example 1 (Part 1) – Show Company Information for a given Company
 - Example 2 (Part 2) – Show All Employees within the Company inside a Table
 - Example 3 (Part 3) – Show specific Database Information based on User Input
 - Example 4 (Part 4) – Show Employee Information based on selected Customer



Code Example 4

Show Employee Information based on selected Customer

ASP.NET Core Example

GetDatafromDatabase [Home](#)

Company

Company Name:

USN

Company Information:

Company Name: **USN**

Website: www.usn.no

Show Employees



© 2021 - GetDatafromDatabase - [Privacy](#)

GetDatafromDatabase [Home](#)

Employees

Employee Information for Customer: USN

EmployeeId	First Name	Last Name
1	Elvis	Presley
2	Knut	Hamsun

© 2021 - GetDatafromDatabase - [Privacy](#)

Index.cshtml

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<script>

    function ShowEmployees() {
        var companyId = document.getElementById("SelectedCompany").value;
        window.location.href = "Employee?CompanyId=" + companyId;
    }

</script>
<div>
    <h1>Company</h1>
    <form method="post">

        Company Name:
        <br />

        <select name="SelectedCompany" id="SelectedCompany" class="form-control input-lg" onchange="this.form.submit()">

            <option value="0">Select Company</option>

            @foreach (var company in Model.companyList)
            {
                @if (Model.companyData.CompanyId == @company.CompanyId)
                {
                    <option value="@company.CompanyId" selected>@company.CompanyName</option>
                }
                else
                {
                    <option value="@company.CompanyId">@company.CompanyName</option>
                }
            }
        </select>
    </form>
    <br />

    @if (@Model.companyData.CompanyName != null)
    {
        <p>Company Information:</p>
        <p>
            Company Name: <strong>@Model.companyData.CompanyName</strong>
            <br />
            Website: @Model.companyData.Website
            <br />
            <br />
            <input type="button" value="Show Employees" onclick="javascript:ShowEmployees()" class="btn btn-info" />
        </p>
    }
</div>
```

Employee.cs

```
using System;
using System.Data;
using System.Collections.Generic;
using Microsoft.Data.SqlClient;

namespace GetDatafromDatabase.Models
{
    public class Employee
    {
        public int EmployeeId { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }

        public List<Employee> GetEmployeeData(int companyId)
        {
            string connectionString = "Data Source=xxx\\SQLEXPRESS;Initial Catalog=COMPANYDB;Integrated Security=True";
            SqlConnection con = new SqlConnection(connectionString);
            con.Open();

            string sqlQuery = "select EmployeeId, FirstName, LastName from EMPLOYEE where CompanyId=@CompanyId";

            SqlCommand cmd = new SqlCommand(sqlQuery, con);

            var companyIdParameter = new SqlParameter("CompanyId", SqlDbType.VarChar);
            companyIdParameter.Value = companyId;
            cmd.Parameters.Add(companyIdParameter);

            SqlDataReader dr = cmd.ExecuteReader();

            List<Employee> employeeList = new List<Employee>();

            while (dr.Read())
            {
                Employee employee = new Employee();

                employee.EmployeeId = Convert.ToInt32(dr["EmployeeId"]);
                employee.FirstName = dr["FirstName"].ToString();
                employee.LastName = dr["LastName"].ToString();

                employeeList.Add(employee);
            }
            con.Close();
            return employeeList;
        }
    }
}
```

Employee.cshtml.cs

```
using System;
using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc.RazorPages;
using GetDatafromDatabase.Models;

namespace GetDatafromDatabase.Pages
{
    public class EmployeeModel : PageModel
    {
        public Company companyData = new Company();
        public List<Employee> employeeList = new List<Employee>();

        public void OnGet()
        {
            int companyId = Convert.ToInt32(Request.Query["CompanyId"]);

            Company company = new Company();
            companyData = company.GetCompanyData(companyId);

            Employee employee = new Employee();
            employeeList = employee.GetEmployeeData(companyId);
        }
    }
}
```


Employee.cshtml

```
@page
@model GetDatafromDatabase.Pages.EmployeeModel
@{
}
<div>
    <h1>Employees</h1>
    <p>Employee Information for Customer: @Model.companyData.CompanyName</p>

    <table class="table">
        <thead>
            <tr>
                <th>EmployeeId</th>
                <th>First Name</th>
                <th>Last Name</th>
            </tr>
        </thead>
        <tbody>
            @foreach (var employee in Model.employeeList)
            {
                <tr>
                    <td> @employee.EmployeeId</td>
                    <td> @employee.FirstName</td>
                    <td> @employee.LastName</td>
                </tr>
            }
        </tbody>
    </table>
</div>
```

Summary

GetDatafromDatabase Home

Company

Webpage1

Company Name:

USN



Company Information:

Company Name: **USN**

Website: www.usn.no

Show Employees



Webpage2

GetDatafromDatabase Home

Employees

Employee Information for Customer: USN

EmployeeId	First Name	Last Name
1	Elvis	Presley
	Knut	Hamsun

base - Privacy

The following ASP.NET Core Web Application with focus on Retrieving Data from Database has been developed. It has been developed in iterations:

- Example 1 – Show Company Information for a given Company
- Example 2 – Show All Employees within the Company inside a Table
- Example 3 – Show specific Database Information based on User Input
- Example 4 – Show Employee Information based on selected Customer

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

