RESTful Web Services

C# Web 2

DE HOGESCHOOL
MET HET NETWERK

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

PeopleAppSolution

- Github classroom

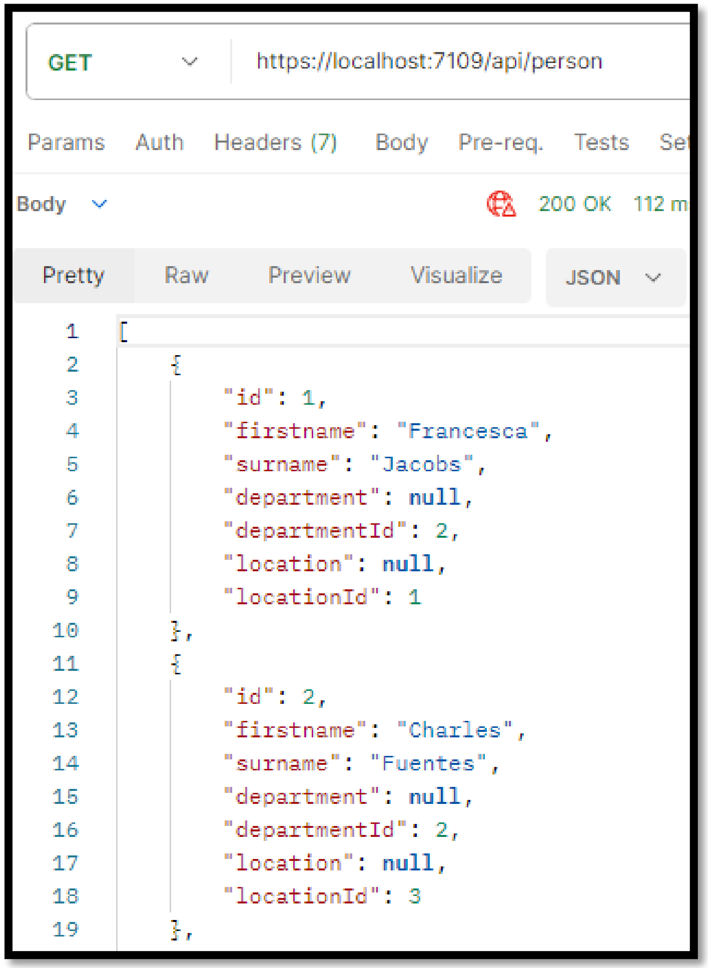
https://classroom.github.com/a/C-gifMjE

Uitbreiding Startproject

- We hebben in onze **PeopleApp webservice** een controller PersonController die onze Http requests behandelt.

- We gaan deze requests uit onze controller kopiëren naar een nieuwe Controller : PersonKeyController.

- Zo blijft onze oude Postman service nog werken met de PersonController en kunnen we een nieuwe service ontwikkelen gebaseerd op ons Key attribuut.



```
[
    {
        "id": 1,
        "firstname": "Francesca",
        "surname": "Jacobs",
        "department": null,
        "departmentId": 2,
        "location": null,
        "locationId": 1
    },
    {
        "id": 2,
        "firstname": "Charles",
        "surname": "Fuentes",
        "department": null,
        "departmentId": 2,
        "location": null,
        "locationId": 3
    },
```

PeopleAppSolution

```
Controllers
•  Add new Api Controller
     •  PersonKeyController
          •  Copy constructor & http actions from PersonController

namespace PeopleApp.Api.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class PersonKeyController : ControllerBase
    {
        AppDbContext _context;
        public PersonKeyController(AppDbContext context)
        {
            _context = context;
        }
        #region Get
          . . .
```

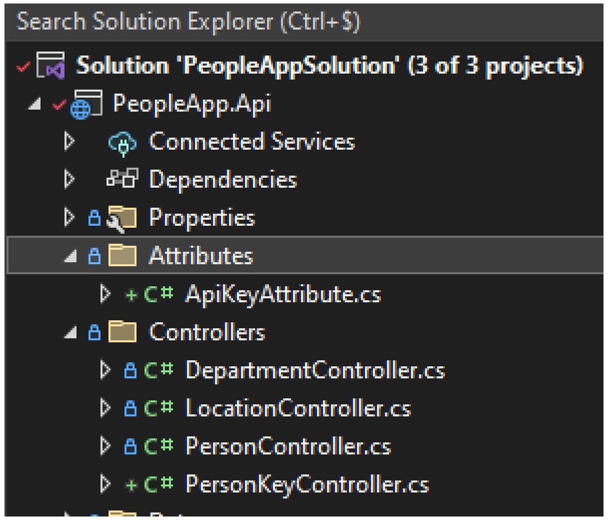
Appsettings – Attribute - ApiKey

appsettings.json:

```
{
  "Logging": {
      "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information",
      "Microsoft.EntityFrameworkCore": "Information"
      }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
      "PeopleConnection":
"Server=(localdb)\\MSSQLLocalDB;Database=People;MultipleActiveResultSets=True"
  },
   "ApiKey": "TestApiKey"
}
```

ApiKeyAttribute

- Add Folder in project folder
  - Attributes
    - Add new class
      - ApiKeyAttribute.cs



```
Search Solution Explorer (Ctrl+$)
Solution 'PeopleAppSolution' (3 of 3 projects)
  PeopleApp.Api
    Connected Services
    Dependencies
    Properties
    Attributes
      ApiKeyAttribute.cs
    Controllers
      DepartmentController.cs
      LocationController.cs
      PersonController.cs
      PersonKeyController.cs
```

PeopleApp - ApiKeyAttribute

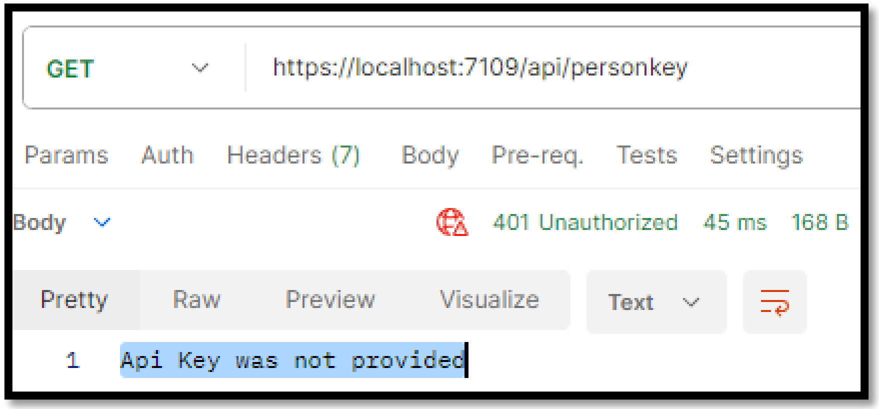
```
namespace PeopleApp.Api.Attributes
{
    [AttributeUsage(validOn: AttributeTargets.Class)]
    public class ApiKeyAttribute : Attribute, IAsyncActionFilter
    {
        private const string APIKEYNAME = "ApiKey";
        private ContentResult GetContentResult(int statusCode, string content)
        {
            var result=new ContentResult();
            result.StatusCode = statusCode;
            result.Content = content;
            return result;
        }
        public async Task OnActionExecutionAsync(
              ActionExecutingContext context, ActionExecutionDelegate next)
        {

         }
    }
}
```

PeopleApp - ApiKeyAttribute

```
public async Task OnActionExecutionAsync(
    ActionExecutingContext context, ActionExecutionDelegate next)
{
      if (!context.HttpContext.Request.Headers.TryGetValue(
                              APIKEYNAME, out var extractedApiKey))
      {
            context.Result = GetContentResult(
                        401, "Api Key was not provided");
            return;
      }
}
```

PersonKeyController - ApiKeyAttribute

```
namespace PeopleApp.Api.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    [ApiKey]
    public class PersonKeyController : ControllerBase
```



https://localhost:7109/api/personkey

401 Unauthorized 45 ms 168 B

```
Api Key was not provided
```

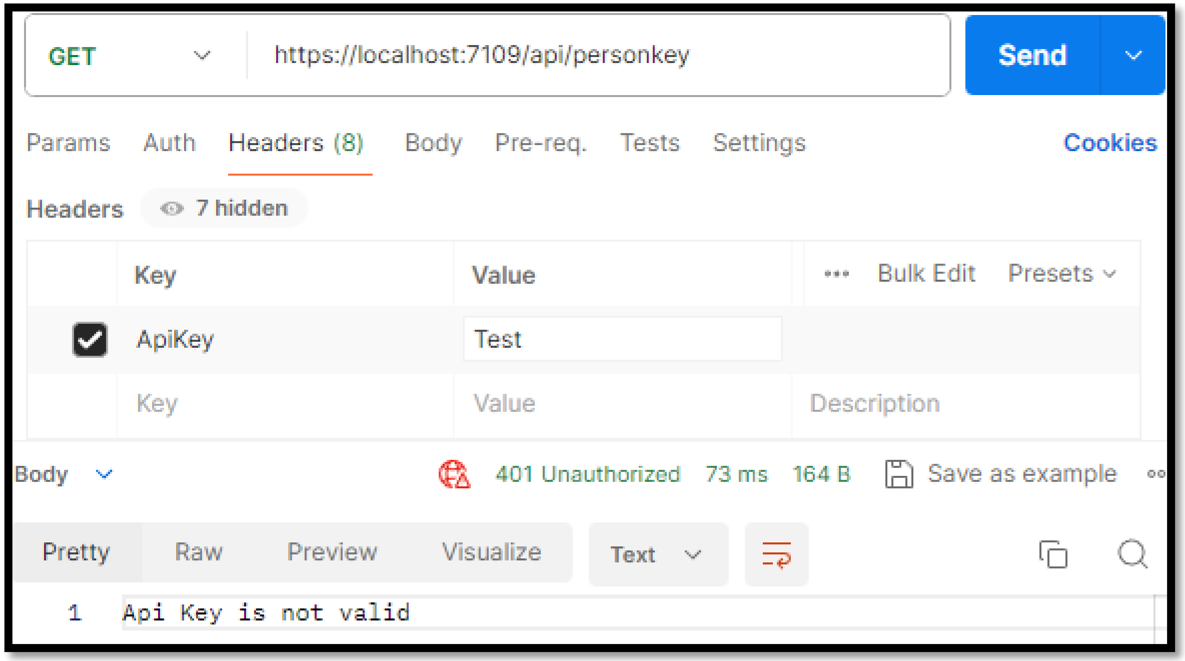
PeopleApp - ApiKeyAttribute

```
public async Task OnActionExecutionAsync(ActionExecutingContext context, ActionExecutionDelegate next)
{
    try
    {
        if (!context.HttpContext.Request.Headers.TryGetValue(APIKEYNAME, out var extractedApiKey))
        {
            context.Result = GetContentResult(401, "Api Key was not provided");
            return;
        }
        var appSettings = context.HttpContext.RequestServices.GetRequiredService<IConfiguration>();
        if (appSettings == null)
        {
            context.Result = GetContentResult(401, "Appsettings not found");
            return;
        }
        var apiKey = appSettings.GetValue<string>(APIKEYNAME);
        if (apiKey == null)
        {
            context.Result = GetContentResult(401, "Appsettings - ApiKey - not found");
            return;
        }
```

PeopleApp - ApiKeyAttribute

```
public async Task OnActionExecutionAsync(ActionExecutingContext context, ActionExecutionDelegate next)
{
    try
    {
        …
        if (apiKey == null)
        {
            context.Result = GetContentResult(401, "Appsettings - ApiKey - not found");
            return;
        }
        if (!apiKey.Equals(extractedApiKey))
        {
            context.Result = GetContentResult(401, "Api Key is not valid");
            return;
        }
        await next();
    }
    catch (Exception ex)
    {
        context.Result = GetContentResult(401, ex.Message);
        return;
    }
```

Postman - ApiKeyAttribute


GET https://localhost:7109/api/personkey

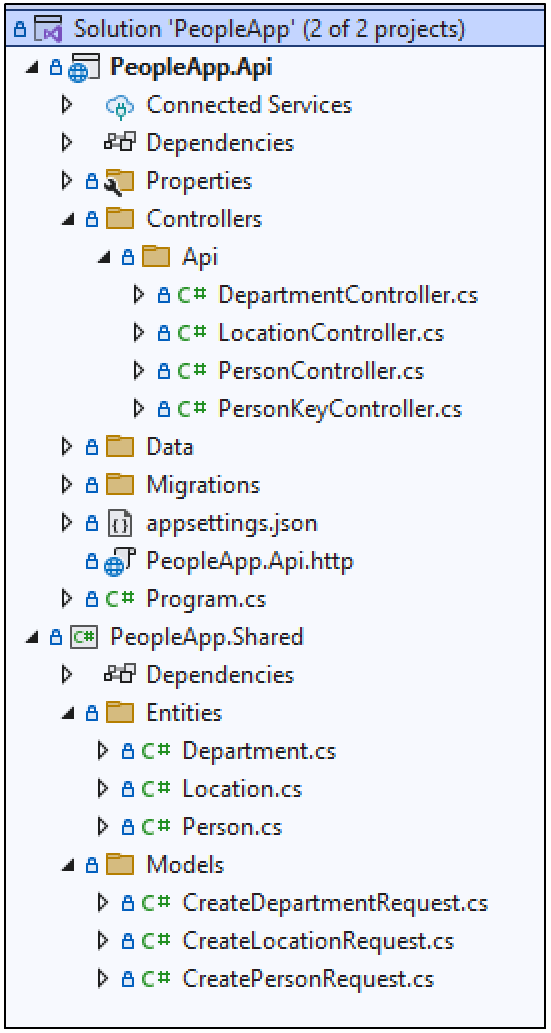
Headers (8)

	Key	Value	Description
✓	ApiKey	Test	

401 Unauthorized 73 ms 164 B

```
Api Key is not valid
```

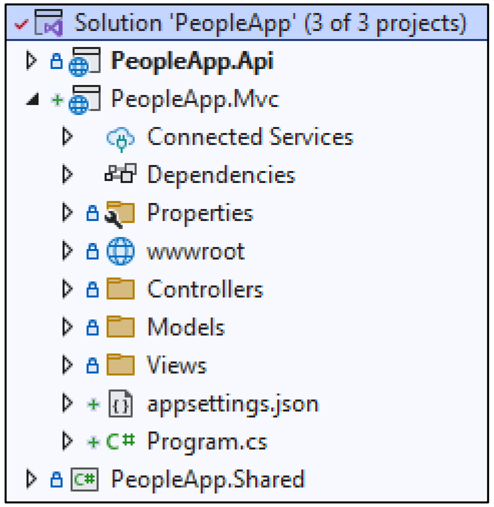
PeopleAppSolution - MVC



```
Solution 'PeopleApp' (2 of 2 projects)
  PeopleApp.Api
    Connected Services
    Dependencies
    Properties
    Controllers
      Api
        DepartmentController.cs
        LocationController.cs
        PersonController.cs
        PersonKeyController.cs
    Data
    Migrations
    appsettings.json
    PeopleApp.Api.http
    Program.cs
  PeopleApp.Shared
    Dependencies
    Entities
      Department.cs
      Location.cs
      Person.cs
    Models
      CreateDepartmentRequest.cs
      CreateLocationRequest.cs
      CreatePersonRequest.cs
```

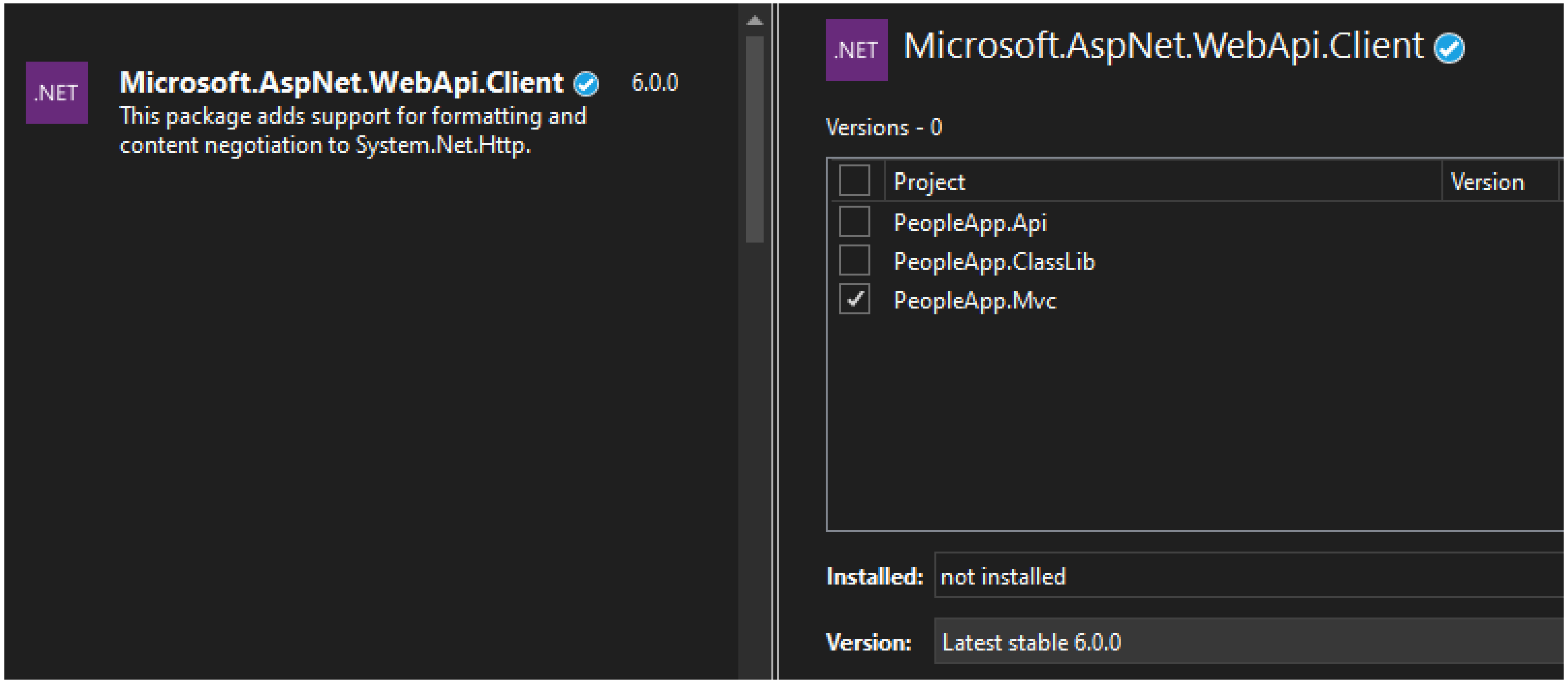
<u>PeopleAppSolution</u>

- Add new project
  - MVC template
    - PeopleApp.Mvc



```
Solution 'PeopleApp' (3 of 3 projects)
  PeopleApp.Api
  PeopleApp.Mvc
    Connected Services
    Dependencies
    Properties
    wwwroot
    Controllers
    Models
    Views
    appsettings.json
    Program.cs
  PeopleApp.Shared
```

Nuget package



.NET **Microsoft.AspNet.WebApi.Client** 6.0.0
This package adds support for formatting and content negotiation to System.Net.Http.

.NET Microsoft.AspNet.WebApi.Client

Versions - 0

	Project	Version
☐	PeopleApp.Api	
☐	PeopleApp.ClassLib	
✓	PeopleApp.Mvc	

Installed: not installed

Version: Latest stable 6.0.0

Helpers - ApiConstants

PeopleApp.Mvc

- Add folder
  - Helpers (Add class)
    - ApiHelper.cs

```
namespace PeopleApp.Mvc.Helpers
{
    public class ApiHelper
    {
        public static string ClientName = "PeopleAppApi";
        public static Uri BaseAddress =
           new Uri("https://localhost:7109/api/");
    }
}
```

Program.cs

```
using PeopleApp.Mvc.Helpers;
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddControllersWithViews();
builder.Services.AddHttpClient(
    ApiHelper.ClientName,
    client =>{ client.BaseAddress = ApiHelper.BaseAddress; });
var app = builder.Build();
```

Helpers - BaseResult

PeopleApp.Mvc.Helpers

- Add class
  - BaseResult.cs

```
namespace PeopleApp.Mvc.Helpers
{
    public class BaseResult
    {
        public bool Succeeded { get; set; }
        public string Error { get; set; }
        public void Failed(string message)
        {
            Error = message;
            Succeeded = false;
        }
    }
}
```

Helpers - ApiResult

PeopleApp.Mvc.Helpers

- Add class
  - ApiResult.cs

```
namespace PeopleApp.Mvc.Helpers
{
    public class ApiResult<T> : BaseResult
    {
        public IEnumerable<T>? Entities { get; set; }
        public T? Entity { get; set; }
    }
}
```

Services - DepartmentRepository

PeopleApp.Mvc.Helpers

- Add folder
  - Services
  - Add class
    - DepartmentRepository.cs

```
namespace PeopleApp.Mvc.Services
{
    public class DepartmentRepository
    {
    }
}
```

Interface - IDepartmentRepository

- PeopleApp.Mvc.Helpers
- Add folder
  - Services/Interfaces
  - Add interface
    - IDepartmentRepository.cs

```
namespace PeopleApp.Mvc.Services.Interfaces
{
    public interface IDepartmentRepository
    {
        ApiResult<Department> Get();
        ApiResult<Department> GetById(long id);
        ApiResult<Department> Add(Department department);
    }
}
```

Services – DepartmentRepository Dependency Injection - HttpClient

```
public class DepartmentRepository : IDepartmentRepository
{
    private readonly IHttpClientFactory _httpFactory;
    public DepartmentRepository(IHttpClientFactory httpFactory)
    {
            _httpFactory = httpFactory;
    }
```

Services – DepartmentRepository

```
public async Task<ApiResult<Department>> GetAsync()
{
    var result = new ApiResult<Department>();
    var client = _httpFactory.CreateClient(ApiHelper.ClientName);
    HttpResponseMessage response = client.GetAsync("department").Result;
    if (response.IsSuccessStatusCode)
    {
        result.Entities =
    await response.Content.ReadAsAsync<IEnumerable<Department>>();
        result.Succeeded = true;
    }
    return result;
}
```

```
public interface IDepartmentRepository
    {
        Task<ApiResult<Department>> GetAsync();
```

Views - Index

PeopleApp.Mvc.Views

- Add folder
  - Department
  - Add Razor View
    - **Name**: Index
    - **Template:** List
    - **Model**: Department

```
@model IEnumerable<PeopleApp.ClassLib.Models.Department>

<h1>Departments</h1>
<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">


…
```

DepartmentController - Index

```
public class DepartmentController : Controller
{
    IDepartmentRepository _repo;
    public DepartmentController(IDepartmentRepository repo)
    {
        _repo = repo;
    }
    public async Task<IActionResult> IndexAsync()
    {
        var result = await _repo.GetAsync();
        if (result.Succeeded)
        {
            return View(result.Entities);
        }
        return View(Enumerable.Empty<Department>());
    }
```

PeopleApp.Mvc - Program.cs

```csharp
using PeopleApp.Mvc.Helpers;
using PeopleApp.Mvc.Services;
using PeopleApp.Mvc.Services.Interfaces;

var builder = WebApplication.CreateBuilder(args);
builder.Services.AddControllersWithViews();
builder.Services.AddHttpClient(
    ApiHelper.ClientName,
    client =>{ client.BaseAddress = ApiHelper.BaseAddress; });
builder.Services.AddScoped<IDepartmentRepository,
        DepartmentRepository>();

var app = builder.Build();
```

Views/Shared – _Layout

```html
<ul class="navbar-nav flex-grow-1">
    <li class="nav-item">
        <a class="nav-link text-dark" asp-area="" asp-controller="People"
                 asp-action="Index">People</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-area="" asp-controller="Department"
                 asp-action="Index">Departments</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-area="" asp-controller="Location"
                 asp-action="Index">Locations</a>
    </li>
</ul>
```

DepartmentRepository - Add

```
public async Task<ApiResult<Department>> AddAsync(Department department)
{
    var result = new ApiResult<Department>();
    var client = _httpFactory.CreateClient(ApiHelper.ClientName);
    HttpResponseMessage response =
        client.PostAsJsonAsync("department", department).Result;
    if (response.IsSuccessStatusCode)
    {
        result.Succeeded = true;
    }
    else
    {
        result.Failed("Error saving department!");
    }
    return result;
}
```

Views - Create

- Views/Department
  - Department
  - Add Razor View
    - **Name**: Create
    - **Template**: Create
    - **Model**: Department

```
@model PeopleApp.ClassLib.Models.Department
@{
    ViewData["Title"] = "Create";
}
<h1>Create</h1>
<h4>Department</h4>
<hr />
<div class="row">
…
```

DepartmentController - Create

```
public class DepartmentController : Controller
{
    ...
    [HttpGet]
    public IActionResult Create()
    {
        return View();
    }
    [HttpPost]
    public IActionResult Create(Department department)
    {
        return View();
    }
```

DepartmentController - Create

```
[HttpPost]
public async Task<IActionResult> CreateAsync(Department department)
{
    if (ModelState.IsValid)
    {
        var result = await _repo.AddAsync(department);
        if (result.Succeeded)
        {
            return RedirectToAction("Index");
        }
        ModelState.AddModelError("", result.Error);
    }
    return View();
}
```

DepartmentRepository - GetById

```
public async Task<ApiResult<Department>> GetByIdAsync(long id)
{
    var result = new ApiResult<Department>();
    var client = _httpFactory.CreateClient(ApiHelper.ClientName);
    HttpResponseMessage response =
         client.GetAsync($"department/{id}").Result;
    if (response.IsSuccessStatusCode)
    {
        result.Entity =
               await response.Content.ReadAsAsync<Department>();
        result.Succeeded = true;
    }
    return result;
}
```

Views - Details

- Views/Department
  - Department
  - Add Razor View
    - **Name**: Details
    - **Template:** Details
    - **Model**: Department

```
@model PeopleApp.ClassLib.Models.Department
@{
    ViewData["Title"] = "Details";
}
<h1>Details</h1>
<div>
    <h4>Department</h4>
    <hr />
    <dl class="row"> …     </dl>
</div>
<div>
    <a asp-action="Index">Back to List</a>
</div>
```

Views - Index

```
@model IEnumerable<PeopleApp.ClassLib.Models.Department>
<h1>Departments</h1>
<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">
  <tbody>
   @foreach (var item in Model)
   {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Id)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Name)
            </td>
            <td>
                @Html.ActionLink("Details", "Details", new { id=item.Id  })
            </td>
        </tr>
   }
  </tbody>
</table>
```

DepartmentController - Details

```
public class DepartmentController : Controller
{
    ...
    public async Task<IActionResult> DetailsAsync(long id)
    {
        var result = await _repo.GetByIdAsync(id);
        if (result.Succeeded)
        {
            return View(result.Entity);
        }
        ModelState.AddModelError("", result.Error);
        return View();
    }
```

PeopleSolution - IDepartmentRepository

```
namespace PeopleApp.Mvc.Services.Interfaces
{
    public interface IDepartmentRepository
    {
        Task<ApiResult<Department>> GetAsync();
        Task<ApiResult<Department>> GetByIdAsync(long id);
        Task<ApiResult<Department>> AddAsync(
                                  Department department);
    }
}
```

PeopleSolution - Oefening

PeopleSolution – PeopleApp.MVC

1) LocationRepository
   - Get
   - Add
   - GetById
2) LocationController
   - Index
   - Create
   - Details
3) PersonRepository
   - Get
   - Add
   - GetById
4) PersonController
   - Index
   - Create
   - Details