**Practical No: 6**

**Date: 13/04/2023**

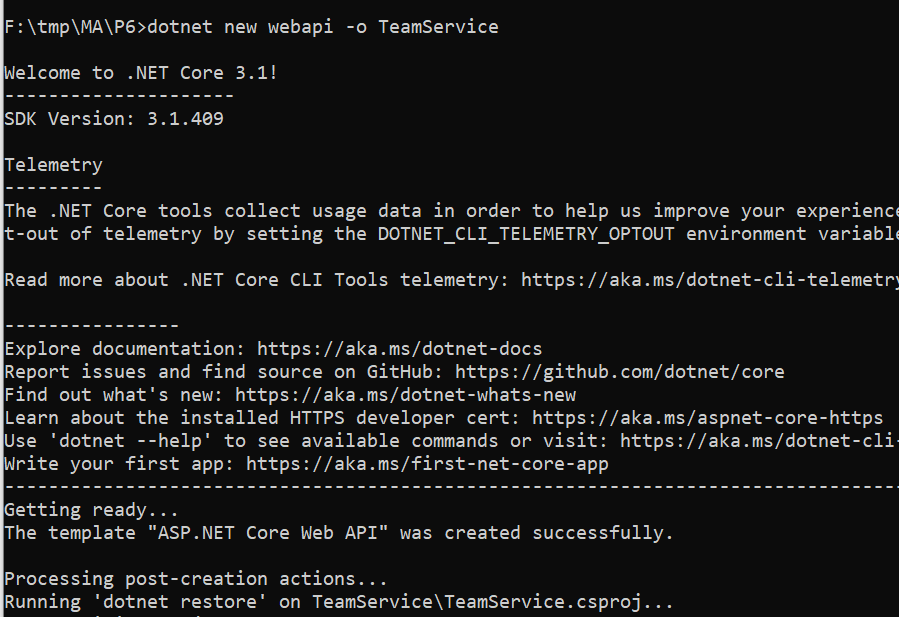
**Aim: Working with Members API of TeamService**

**Description:**

Give Overview of Members API of TeamService

**Code & Output:**

**Steps:**

1. **Create new project:  
   Command :  
   dotnet new webapi -o TeamService  
   output:  
   **
2. **Remove existing weatherforecast files both model and controller files.**
3. **Add Member.cs to "D:\TeamService\Models" folder and add following code.**using System;

namespace TeamService.Models{

public class Member {

public Guid ID { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public Member() { }

public Member(Guid id) : this(){

this.ID = id;

}

public Member(string firstName, string lastName, Guid id) : this(id){

this.FirstName = firstName;

this.LastName = lastName;

}

public override string ToString(){

return this.LastName;

}

}

}

1. **Add Team.cs to "D:\TeamService\Models" folder and add following code.**

using System;

using System.Collections.Generic;

namespace TeamService.Models

{

public class Team

{

public string Name { get; set; }

public Guid ID { get; set; }

public ICollection<Member> Members { get; set; }

public Team()

{

this.Members = new List<Member>();

}

public Team(string name) : this()

{

this.Name = name;

}

public Team(string name, Guid id) : this(name)

{

this.ID = id;

}

public override string ToString()

{

return this.Name;

}

}

}

1. **Add TeamsController.cs to "D:\TeamService\Controllers" folder and add following code.**

using System;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Mvc;

using System.Collections.Generic;

using System.Linq;

using TeamService.Models;

using System.Threading.Tasks;

using TeamService.Persistence;

namespace TeamService

{

[Route("[controller]")]

public class TeamsController : Controller

{

ITeamRepository repository;

public TeamsController(ITeamRepository repo)

{

repository = repo;

}

[HttpGet]

public virtual IActionResult GetAllTeams()

{

return this.Ok(repository.List());

}

[HttpGet("{id}")]

public IActionResult GetTeam(Guid id)

{

Team team = repository.Get(id);

if (team != null)

{

return this.Ok(team);

}

else {

return this.NotFound();

}

}

[HttpPost]

public virtual IActionResult CreateTeam([FromBody]Team newTeam)

{

repository.Add(newTeam);

return this.Created($"/teams/{newTeam.ID}", newTeam);

}

[HttpPut("{id}")]

public virtual IActionResult UpdateTeam([FromBody]Team team, Guid id)

{

team.ID = id;

if(repository.Update(team) == null)

{

return this.NotFound();

}

else

{

return this.Ok(team);

}

}

[HttpDelete("{id}")]

public virtual IActionResult DeleteTeam(Guid id)

{

Team team = repository.Delete(id);

if (team == null)

{

return this.NotFound();

}

else {

return this.Ok(team.ID);

}

}

}

}

1. Add MembersController.cs to "D:\TeamService\Controllers" folder and add following code.

using System;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Mvc;

using System.Collections.Generic;

using System.Linq;

using TeamService.Models;

using System.Threading.Tasks;

using TeamService.Persistence;

namespace TeamService

{

[Route("/teams/{teamId}/[controller]")]

public class MembersController : Controller

{

ITeamRepository repository;

public MembersController(ITeamRepository repo)

{

repository = repo;

}

[HttpGet]

public virtual IActionResult GetMembers(Guid teamID)

{

Team team = repository.Get(teamID);

if (team == null)

{

return this.NotFound();

}

else

{

return this.Ok(team.Members);

}

}

[HttpGet]

[Route("/teams/{teamId}/[controller]/{memberId}")]

public virtual IActionResult GetMember(Guid teamID, Guid memberId)

{

Team team = repository.Get(teamID);

if (team == null)

{

return this.NotFound();

}

else

{

var q = team.Members.Where(m => m.ID == memberId);

if (q.Count() < 1)

{

return this.NotFound();

}

else

{

return this.Ok(q.First());

}

}

}

[HttpPut]

[Route("/teams/{teamId}/[controller]/{memberId}")]

public virtual IActionResult UpdateMember([FromBody]Member updatedMember, Guid teamID, Guid memberId)

{

Team team = repository.Get(teamID);

if (team == null)

{

return this.NotFound();

}

else

{

var q = team.Members.Where(m => m.ID == memberId);

if (q.Count() < 1)

{

return this.NotFound();

}

else

{

team.Members.Remove(q.First());

team.Members.Add(updatedMember);

return this.Ok();

}

}

}

[HttpPost]

public virtual IActionResult CreateMember([FromBody]Member newMember, Guid teamID)

{

Team team = repository.Get(teamID);

if (team == null)

{

return this.NotFound();

}

else

{

team.Members.Add(newMember);

var teamMember = new { TeamID = team.ID, MemberID = newMember.ID };

return this.Created($"/teams/{teamMember.TeamID}/[controller]/{teamMember.MemberID}", teamMember);

}

}

[HttpGet]

[Route("/members/{memberId}/team")]

public IActionResult GetTeamForMember(Guid memberId)

{

var teamId = GetTeamIdForMember(memberId);

if (teamId != Guid.Empty)

{

return this.Ok(new { TeamID = teamId });

}

else

{

return this.NotFound();

}

}

private Guid GetTeamIdForMember(Guid memberId)

{

foreach (var team in repository.List())

{

var member = team.Members.FirstOrDefault(m => m.ID == memberId);

if (member != null)

{

return team.ID;

}

}

return Guid.Empty;

}

}

}

1. **Create folder "D:\TeamService\Persistence":**
2. **Add ITeamReposiroty.cs to "D:\TeamService\Persistence" folder and add following code.**

using System;

using System.Collections.Generic;

using TeamService.Models;

namespace TeamService.Persistence

{

public interface ITeamRepository

{

IEnumerable<Team> List();

Team Get(Guid id);

Team Add(Team team);

Team Update(Team team);

Team Delete(Guid id);

}

}

1. Add MemoryTeamRepository.cs to "D:\TeamService\Persistence" folder and add following code.

using System;

using System.Collections.Generic;

using System.Linq;

using TeamService;

using TeamService.Models;

namespace TeamService.Persistence

{

public class MemoryTeamRepository : ITeamRepository

{

protected static ICollection<Team> teams;

public MemoryTeamRepository()

{

if (teams == null)

{

teams = new List<Team>();

}

}

public MemoryTeamRepository(ICollection<Team> teams)

{

MemoryTeamRepository.teams = teams;

}

public IEnumerable<Team> List()

{

return teams;

}

public Team Get(Guid id)

{

return teams.FirstOrDefault(t => t.ID == id);

}

public Team Update(Team t)

{

Team team = this.Delete(t.ID);

if (team != null)

{

team = this.Add(t);

}

return team;

}

public Team Add(Team team)

{

teams.Add(team);

return team;

}

public Team Delete(Guid id)

{

var q = teams.Where(t => t.ID == id);

Team team = null;

if (q.Count() > 0)

{

team = q.First();

teams.Remove(team);

}

return team;

}

}

}

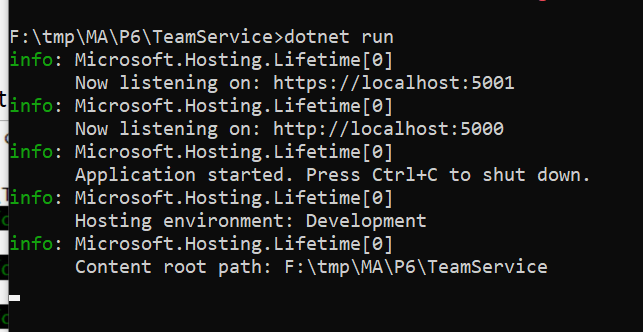
1. **Add following line to Startup.cs in public void ConfigureServices(IServiceCollection services) method**

services.AddScoped<ITeamRepository, MemoryTeamRepository>();

1. Now open two command prompts to run this project
2. On Command prompt 1: (go inside folder teamservice first)

Commands:

dotnet run

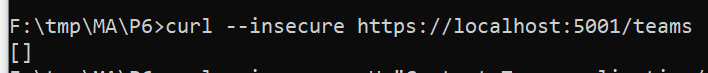


1. On command prompt 2
   1. To get all teams

Commands: To get all teams

curl --insecure <https://localhost:5001/teams>

Output

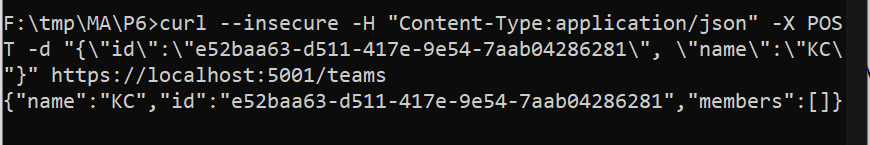


* 1. To create new team

Commands

curl --insecure -H "Content-Type:application/json" –X POST –d "{\"id\":\"e52baa63-d511-417e-9e54-7aab04286281\", \"name\":\"KC\"}" <https://localhost:5001/teams>

Output

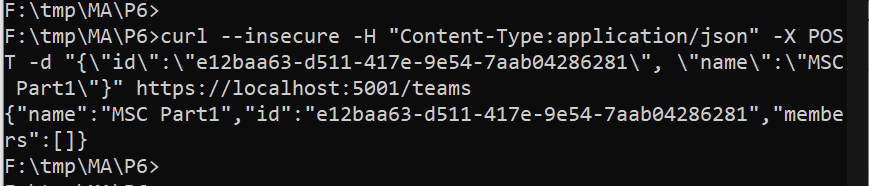


* 1. **To create one more new team**

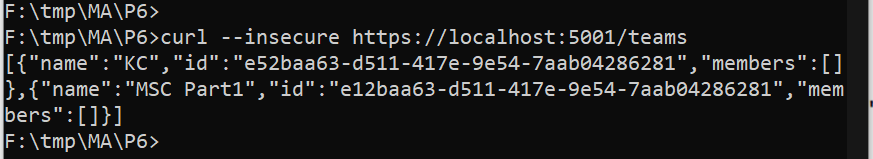
**Commands**

curl --insecure -H "Content-Type:application/json" –X POST –d "{\"id\":\"e12baa63-d511-417e-9e54-7aab04286281\", \"name\":\"MSC Part1\"}" <https://localhost:5001/teams>

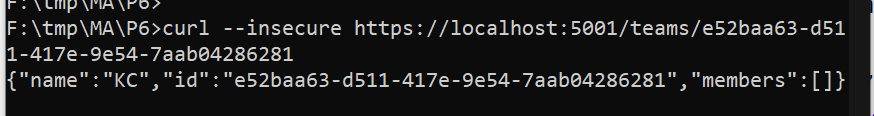
Output



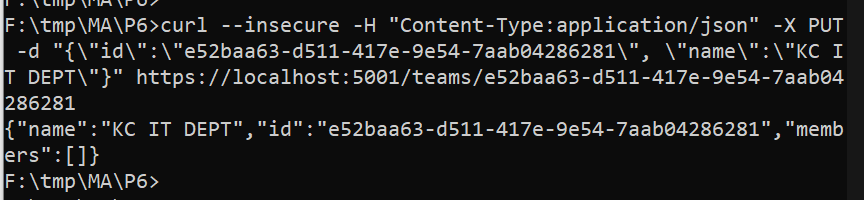
* 1. **To get all teams  
     Commands: To get all teams  
       
     curl --insecure** [**https://localhost:5001/teams**](https://localhost:5001/teams) **Output**

****

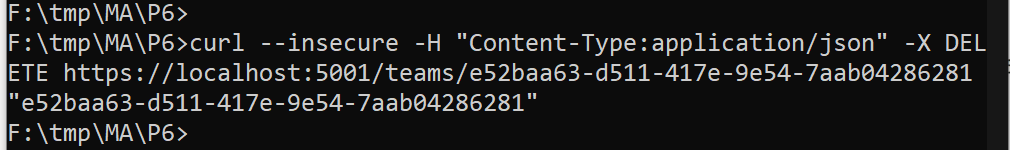
* 1. **to get single team with team-id as parameter  
     Commands  
       
     curl --insecure** [**https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281**](https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281) **Output**

****

* 1. **to update team details (change name of first team from "KC" to "KC IT DEPT")  
     Commands**curl --insecure -H "Content-Type:application/json" –X PUT –d "{\"id\":\"e52baa63-d511-417e-9e54-7aab04286281\", \"name\":\"KC IT DEPT\"}" <https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281>  
       
     Output



* 1. To delete team  
     Commands  
       
     curl --insecure -H "Content-Type:application/json" –X DELETE <https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281>  
       
     Output



* 1. With get all teams now it shows only one team (first one is deleted)  
     Commands  
       
     curl --insecure <https://localhost:5001/teams>  
       
     Output

