## **自定义存储库**

## **定义Character实体**

public class Character

{

    public Character(string name)

    {

        Name = name;

    }

    [Key]

    public Guid Id { get; private set; } = Guid.NewGuid();

    public string Name { get; private set; } = String.Empty;

}

## **定义存储库接口**

public interface ICharacterRepository {

    IEnumerable<Character> ListAll ();

    void Add (Character character);

}

## **存储库的具体实现**

public class CharacterRepository : ICharacterRepository {

    private readonly ApplicationDbContext \_dbContext;

    public CharacterRepository (ApplicationDbContext dbContext) {

        \_dbContext = dbContext;

}

    public IEnumerable<Character> ListAll () {

        return \_dbContext.Characters.AsEnumerable ();

}

    public void Add (Character character) {

        \_dbContext.Characters.Add (character);

        \_dbContext.SaveChanges ();

    }

}

## **注册存储库服务**

public void ConfigureServices (IServiceCollection services) {

    // 添加数据库上下文

    services.AddDbContext<ApplicationDbContext> (options => options.UseInMemoryDatabase ("InMemoryDb"));

    // 注册 CharacterRepository

services.AddScoped<ICharacterRepository, CharacterRepository>();

    ...

}

**应用存储库**

## **注入存储库的实例**

public class IndexModel : PageModel {

    private readonly ICharacterRepository \_characterRepository;

    // 依赖注入

    public IndexModel (ICharacterRepository characterRepository) {

        \_characterRepository = characterRepository;

}

public IEnumerable<Character> Characters { get; set; }

    public void OnGet () {

        PopulateCharactersIfNoneExist ()

        Characters = \_characterRepository.ListAll ();

}

    private void PopulateCharactersIfNoneExist () {

        if (!\_characterRepository.ListAll ().Any ()) {

            \_characterRepository.Add (new Character ("Darth Maul"));

            \_characterRepository.Add (new Character ("Darth Vader"));

            \_characterRepository.Add (new Character ("Yoda"));

            \_characterRepository.Add (new Character ("Mace Windu"));

        }

    }

}