Qn: Web API with Entity Framework

public class Customer : IComparable<Customer>

{

public int Id { get; set; }

public string Name { get; set; }

public int Age { get; set; }

public int CompareTo(Customer? other)

{

return this.Id.CompareTo(other.Id);

}

}

public class OfficeContext : DbContext

{

public OfficeContext(DbContextOptions options) : base(options)

{

}

public DbSet<Customer> Customers { get; set; }

protected override void OnModelCreating(ModelBuilder modelBuilder) //override onmodel tab

{

modelBuilder.Entity<Customer>().HasData(

new Customer() { Id = 101, Name = "Peppers", Age = 23 }

);

}

}

public interface IRepo<K, T>

{

ICollection<T> GetAll();

T Get(K k);

T Add(T item);

bool Remove(K id);

bool Update(T item);

}

public class CustomerRepo : IRepo<int, Customer>

{

private readonly OfficeContext \_context;

public CustomerRepo(OfficeContext context)

{

\_context = context;

}

public CustomerRepo()

{

}

public Customer Add(Customer item)

{

\_context.Customers.Add(item);

\_context.SaveChanges();

return item;

}

public Customer Get(int id)

{

Customer customer = \_context.Customers.FirstOrDefault(x => x.Id == id);

return customer;

}

public ICollection<Customer> GetAll()

{

return \_context.Customers.ToList();

}

public bool Remove(int id)

{

Customer customer = Get(id);

\_context.Customers.Remove(customer);

\_context.SaveChanges();

return true;

}

public bool Update(Customer item)

{

Customer customer = \_context.Customers.FirstOrDefault(x => x.Id == item.Id);

if (customer != null)

{

customer.Name = item.Name;

customer.Age = item.Age;

\_context.Customers.Update(customer);

\_context.SaveChanges();

return true;

}

return false;

}

}

[Route("api/[controller]")]

[ApiController]

public class CustomerController : ControllerBase

{

private readonly IRepo<int, Customer> \_repo;

public CustomerController(IRepo<int, Customer> repo)

{

\_repo = repo;

}

[HttpGet]

public IEnumerable<Customer> Get()

{

return \_repo.GetAll();

}

[HttpGet]

[Route("SingleCustomer")]

public Customer Get(int id)

{

return \_repo.Get(id);

}

[HttpPost]

public Customer Post(Customer customer)

{

\_repo.Add(customer);

return customer;

}

[HttpPut]

public Customer Put(int id, Customer cust) //update //in postman to put in parameter id(3) and body(one customer obj id 3 in json)

{

var customer = \_repo.Get(cust.Id);

if (customer != null)

{

customer.Name = cust.Name;

customer.Age = cust.Age;

\_repo.Update(customer);

}

return customer;

}

[HttpDelete]

public Customer Delete(int id)

{

\_repo.Remove(id);

var customer = \_repo.Get(id);

return customer;

}

}













