





//Classes:

//Authentication:

namespace Microservices\_Casestudy.Models

{

using System;

using System.Collections.Generic;

public partial class Authentication

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public Authentication()

{

this.Orders = new HashSet<Order>();

this.Shop\_cart = new HashSet<Shop\_cart>();

}

public string user\_name { get; set; }

public string password { get; set; }

public int user\_id { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<Order> Orders { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<Shop\_cart> Shop\_cart { get; set; }

}

}

//Order.cs

namespace Microservices\_Casestudy.Models

{

using System;

using System.Collections.Generic;

public partial class Order

{

public int order\_id { get; set; }

public Nullable<int> user\_is { get; set; }

public string delivery\_addr { get; set; }

public Nullable<System.DateTime> Order\_date { get; set; }

public string payment\_mode { get; set; }

public Nullable<int> order\_amt { get; set; }

public virtual Authentication Authentication { get; set; }

}

}

//Product.cs:

namespace Microservices\_Casestudy.Models

{

using System;

using System.Collections.Generic;

public partial class Product

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public Product()

{

this.Shop\_cart = new HashSet<Shop\_cart>();

}

public int Product\_id { get; set; }

public string Product\_name { get; set; }

public Nullable<int> Product\_price { get; set; }

public string Product\_des { get; set; }

public string Prod\_img\_url { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<Shop\_cart> Shop\_cart { get; set; }

}

}

//Shop\_cart.cs:

namespace Microservices\_Casestudy.Models

{

using System;

using System.Collections.Generic;

public partial class Shop\_cart

{

public int id { get; set; }

public Nullable<int> user\_id { get; set; }

public Nullable<int> Prod\_id { get; set; }

public string Prod\_name { get; set; }

public Nullable<int> Qty { get; set; }

public Nullable<int> sub\_tot { get; set; }

public virtual Authentication Authentication { get; set; }

public virtual Product Product { get; set; }

}

}

//CONTROLLERS:

//Auth\_controller:

using Microservices\_Casestudy.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace Microservices\_Casestudy.Controllers

{

public class AuthController : ApiController

{

public HttpResponseMessage Post([FromBody] Authentication obj)

{

Microservice\_DBEntities microservice\_DBEntities = new Microservice\_DBEntities();

var res=microservice\_DBEntities.Authentications.Add(obj);

if (res!=null)

{

return Request.CreateResponse(HttpStatusCode.OK, res);

}

else

{

return Request.CreateErrorResponse(HttpStatusCode.NotFound, "Not found");

}

}

}

}

//Products

using Microservices\_Casestudy.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace Microservices\_Casestudy.Controllers

{

public class ProductsController : ApiController

{

public HttpResponseMessage Get()

{

Microservice\_DBEntities microservice\_DBEntities = new Microservice\_DBEntities();

return Request.CreateResponse(HttpStatusCode.OK, microservice\_DBEntities.Products.ToList());

}

}

}

//order:

using Microservices\_Casestudy.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace Microservices\_Casestudy.Controllers

{

public class OrderController : ApiController

{

public HttpResponseMessage Post([FromBody] Order obj)

{

Microservice\_DBEntities microservice\_DBEntities = new Microservice\_DBEntities();

microservice\_DBEntities.Orders.Add(obj);

return Request.CreateResponse(HttpStatusCode.OK, "added");

}

}

}

//cart:

using Microservices\_Casestudy.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace Microservices\_Casestudy.Controllers

{

public class CartController : ApiController

{

public HttpResponseMessage Get([FromUri] int id)

{

Microservice\_DBEntities microservice\_DBEntities = new Microservice\_DBEntities();

var res = microservice\_DBEntities.Shop\_cart.FirstOrDefault(x => x.id == id);

if (res != null)

{

return Request.CreateResponse(HttpStatusCode.OK, microservice\_DBEntities.Shop\_cart.ToList());

}

else

{

return Request.CreateErrorResponse(HttpStatusCode.NotFound, "record not found");

}

}

public HttpResponseMessage Post([FromBody] Shop\_cart obj)

{

Microservice\_DBEntities microservice\_DBEntities = new Microservice\_DBEntities();

var res = microservice\_DBEntities.Shop\_cart.Add(obj);

microservice\_DBEntities.SaveChanges();

return Request.CreateResponse(HttpStatusCode.OK, res);

}

public HttpResponseMessage Delete([FromUri] int id,int p\_id)

{

Microservice\_DBEntities microservice\_DBEntities = new Microservice\_DBEntities();

var res = microservice\_DBEntities.Shop\_cart.FirstOrDefault(x => x.id == id && x.Prod\_id == p\_id);

if (res != null)

{

microservice\_DBEntities.Shop\_cart.Remove(res);

microservice\_DBEntities.SaveChanges();

return Request.CreateResponse(HttpStatusCode.OK, res);

}

else

{

return Request.CreateErrorResponse(HttpStatusCode.NotFound, "not found");

}

}

}

}