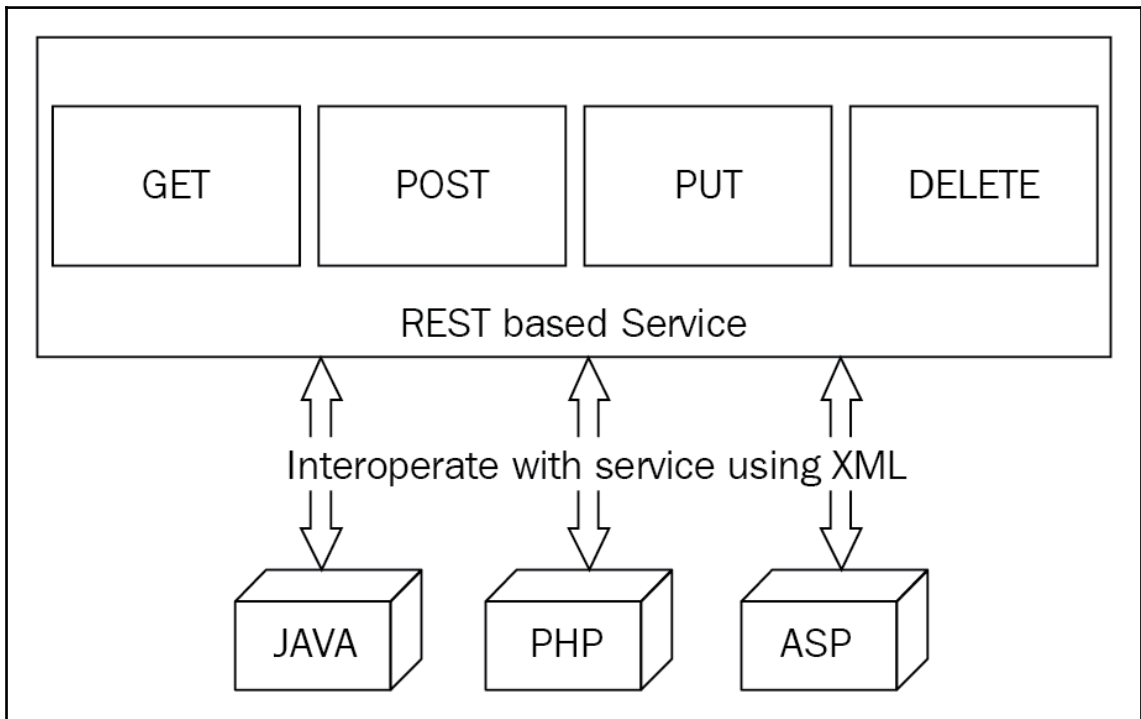
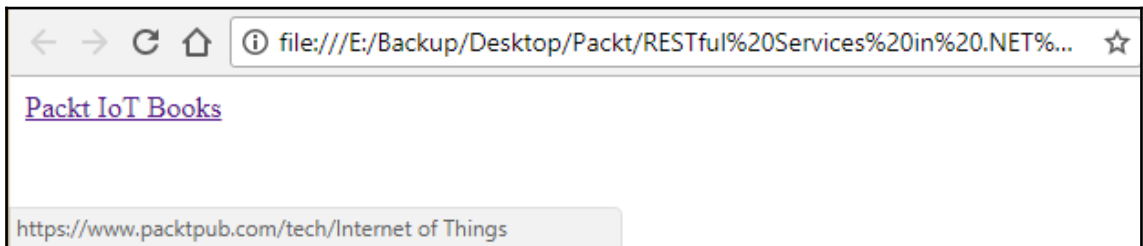
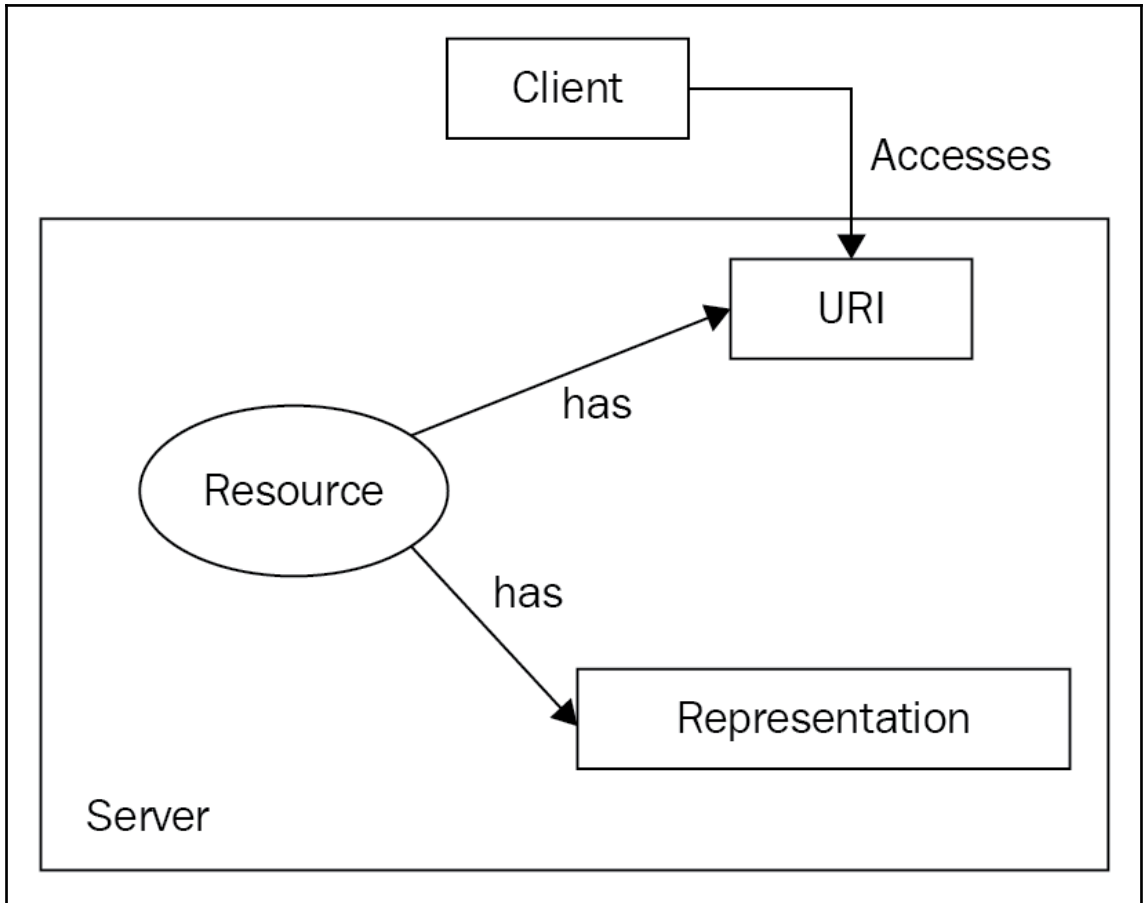
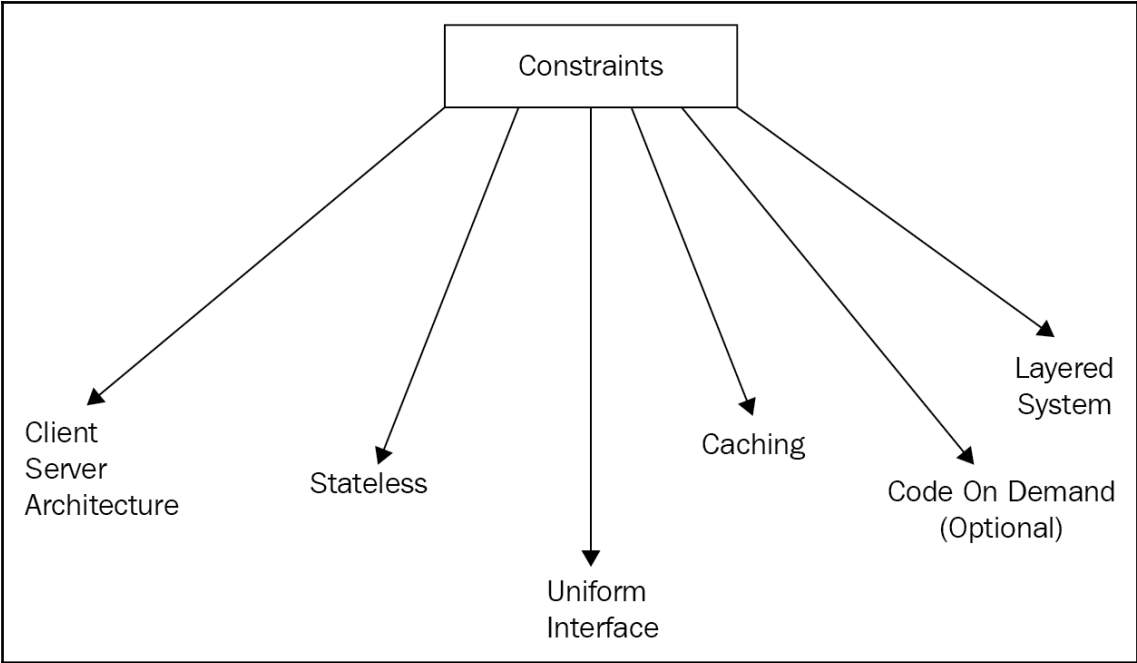
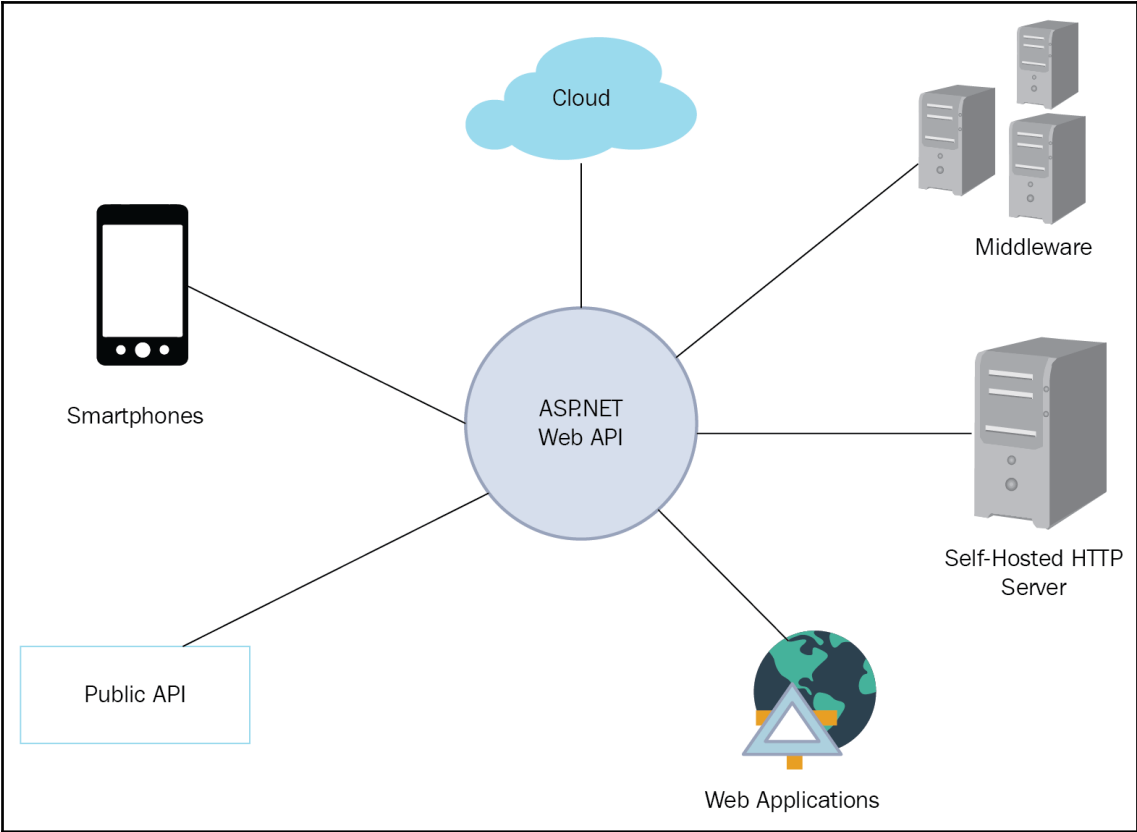


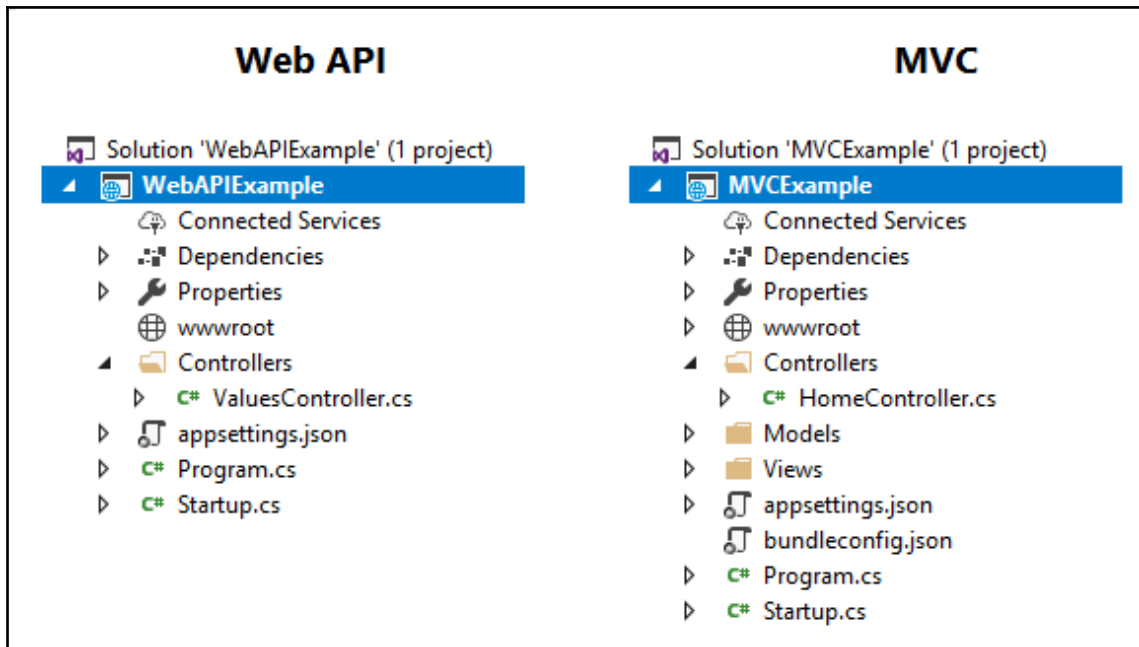
Chapter 1: Getting Started



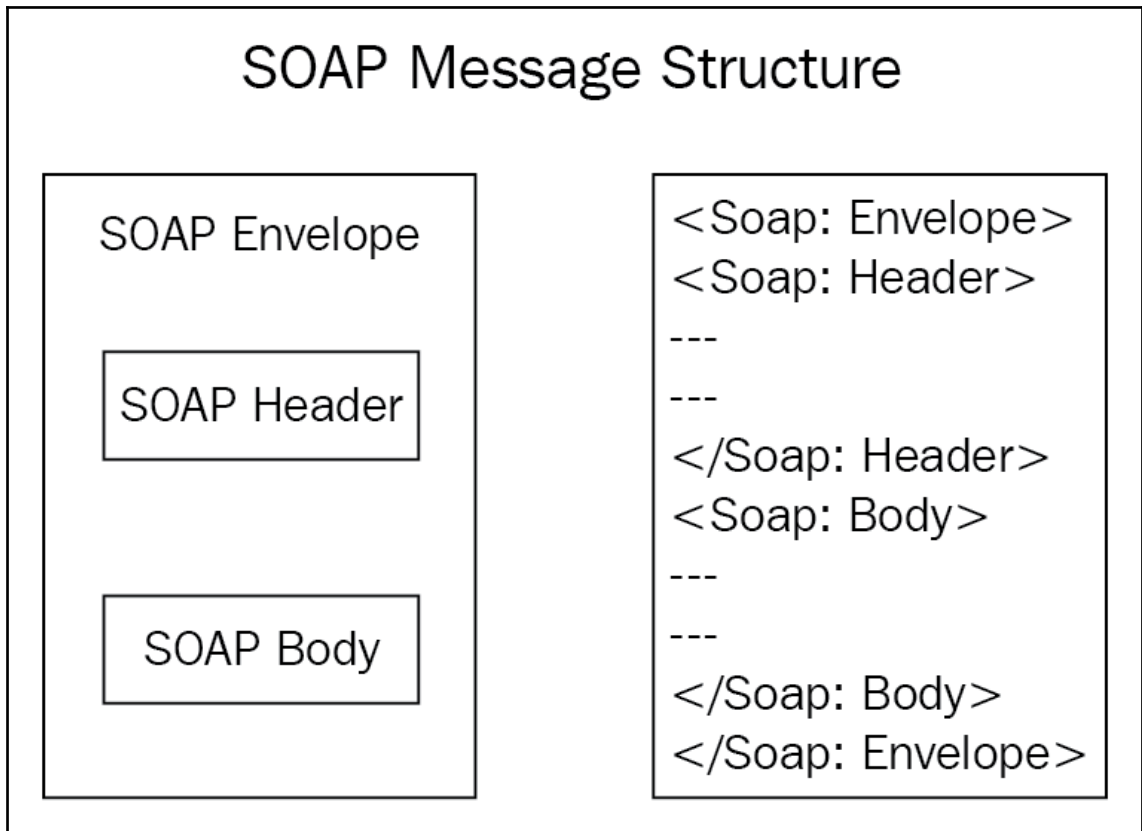


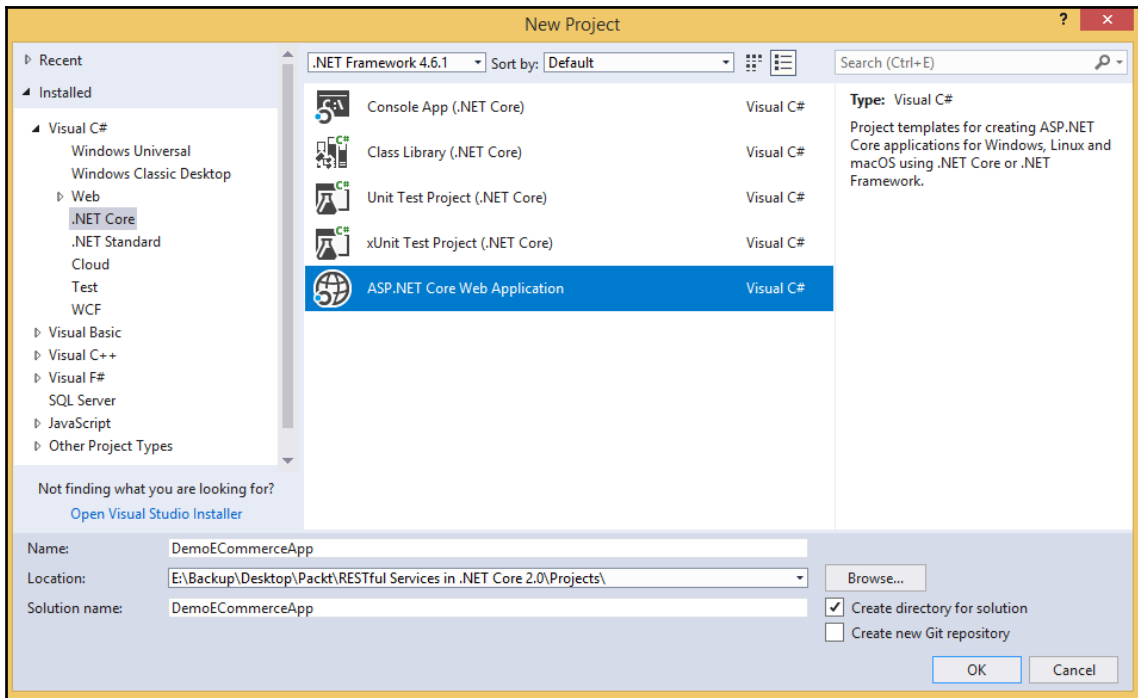


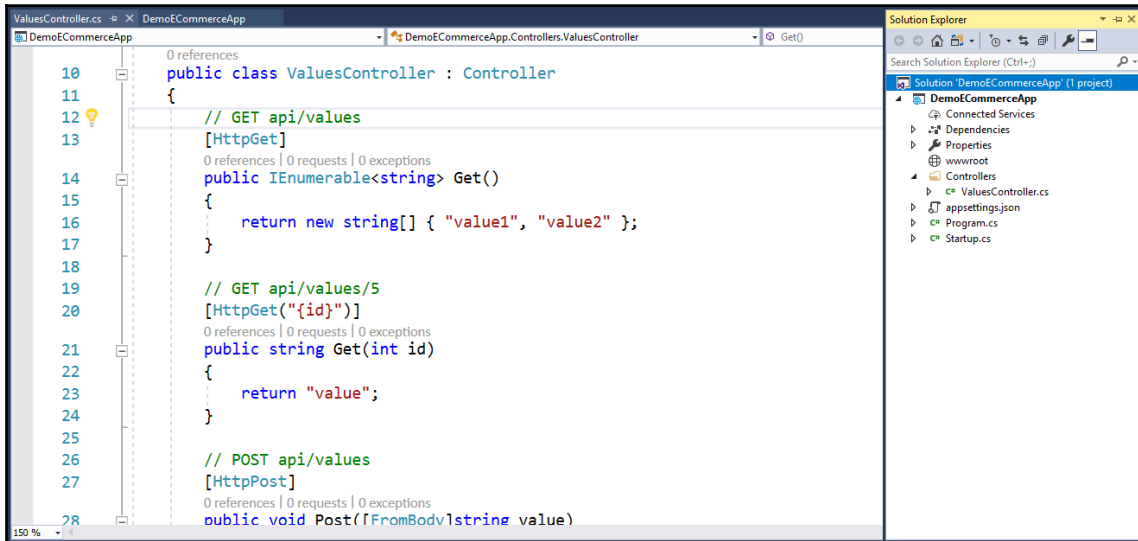
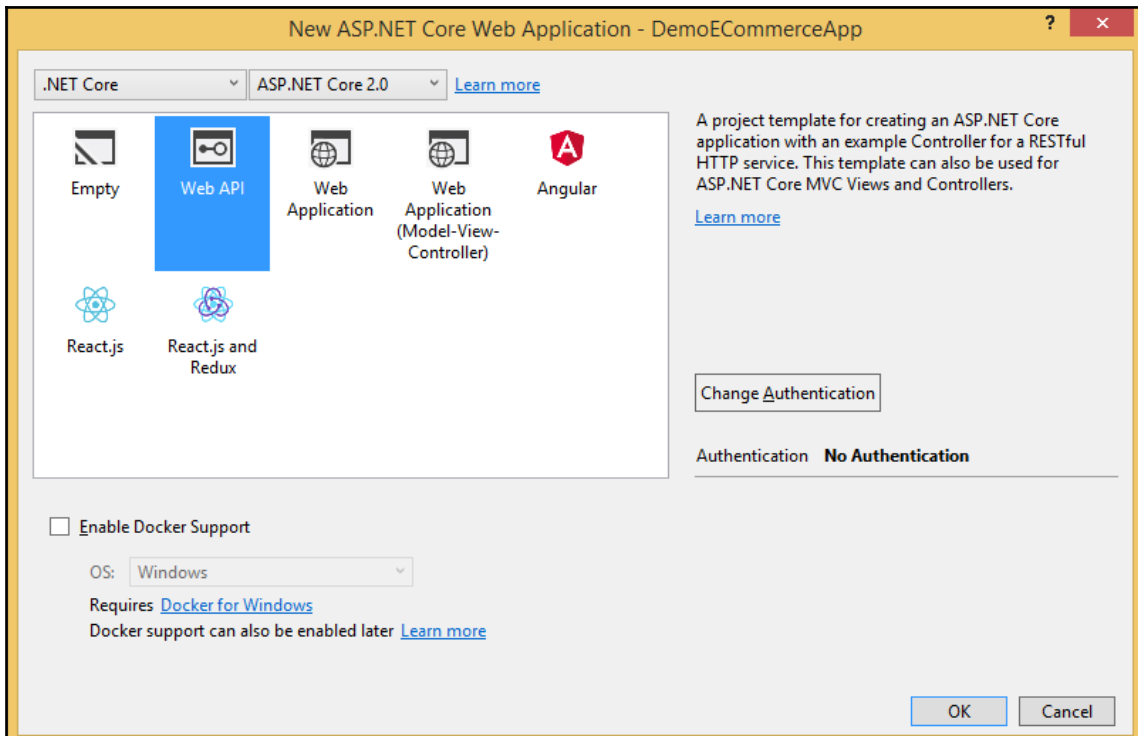


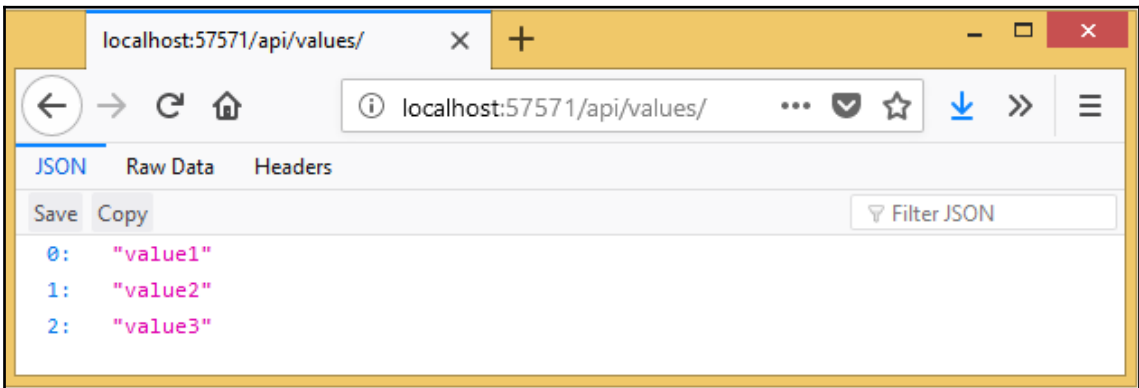
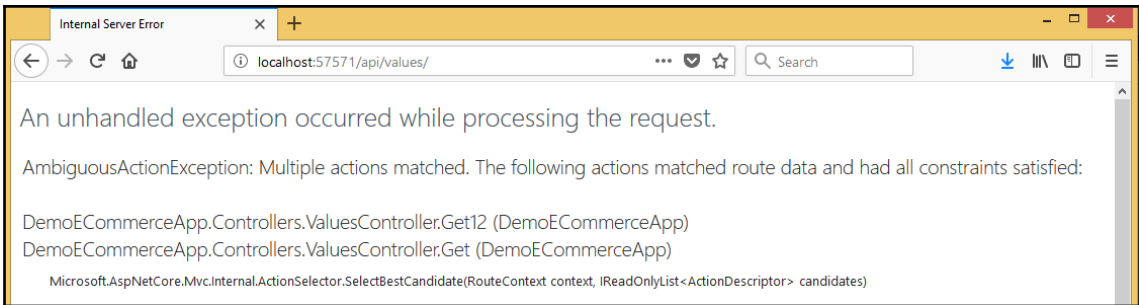
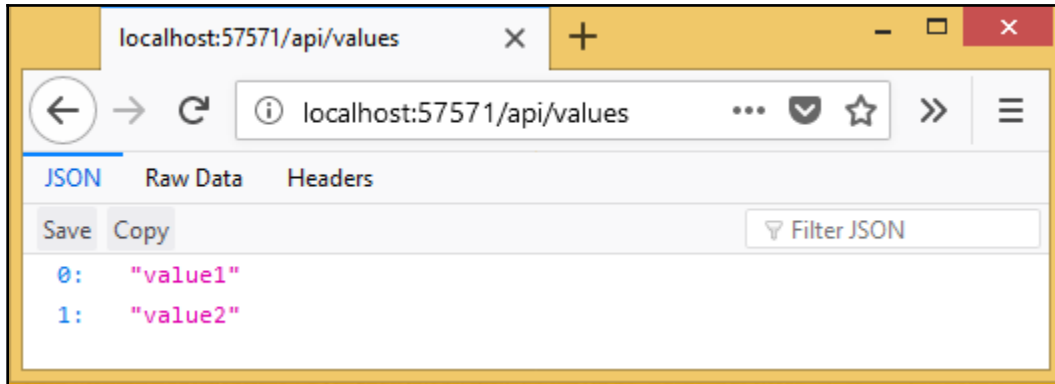


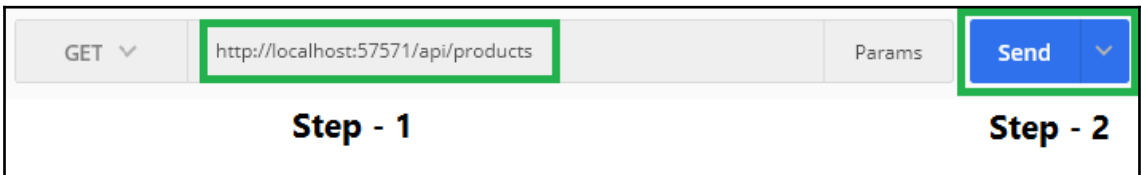
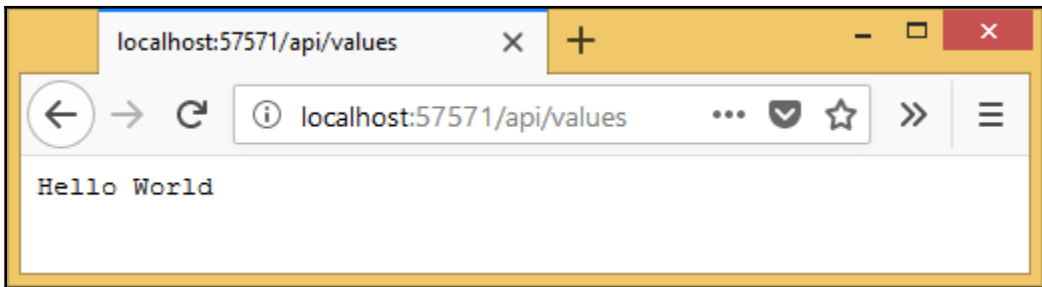
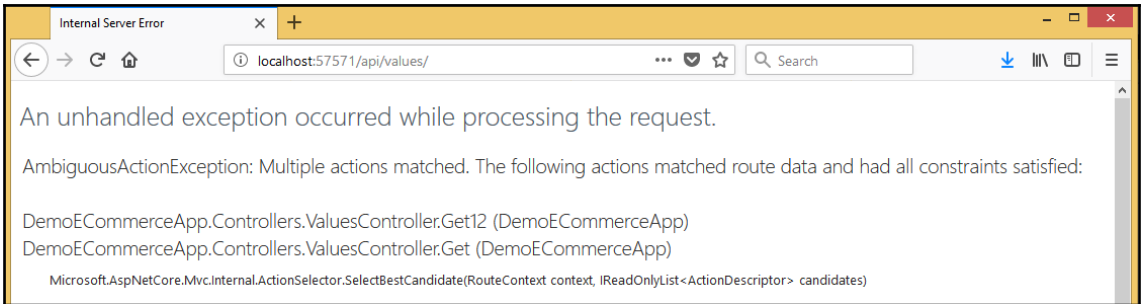
Chapter 2: Building the Initial Framework – Laying the Foundation of the Application

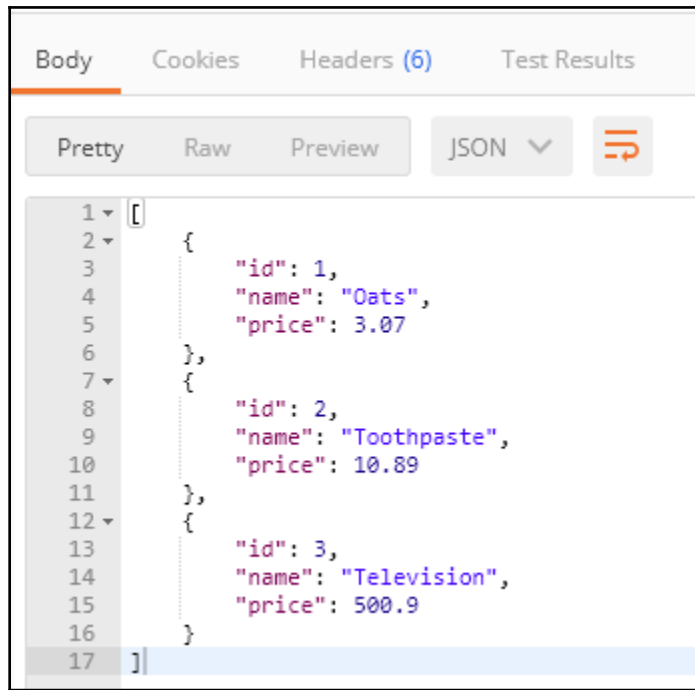




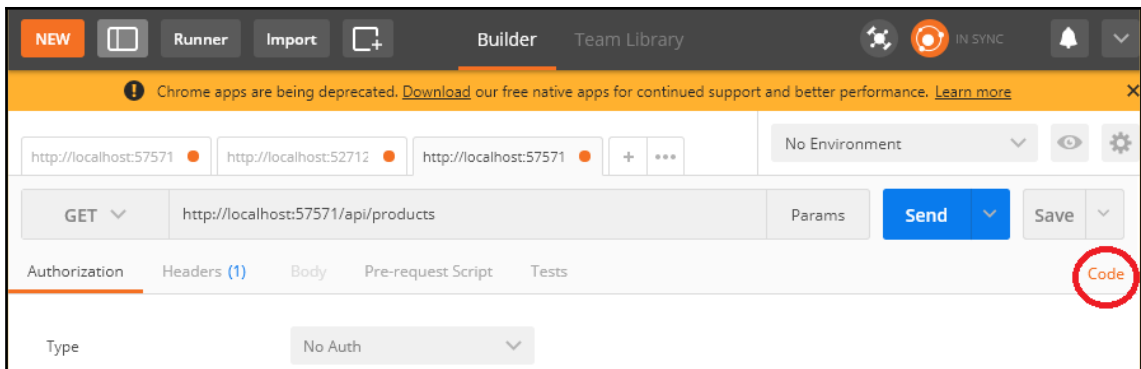








```
1 [
2   {
3     "id": 1,
4     "name": "Oats",
5     "price": 3.07
6   },
7   {
8     "id": 2,
9     "name": "Toothpaste",
10    "price": 10.89
11  },
12  {
13    "id": 3,
14    "name": "Television",
15    "price": 500.9
16  }
17 ]
```



Chrome apps are being deprecated. Download our free native apps for continued support and better performance. Learn more

GENERATE CODE SNIPPETS

HTTP ▾ Copy to Clipboard

```
1 GET /api/products HTTP/1.1
2 Host: localhost:57571
3 Cache-Control: no-cache
4 Postman-Token: 8a2b38b6-61ed-1cb3-66c2-ebb438f3f634
5
```

GENERATE CODE SNIPPETS

JavaScript, jQuery, AJAX ▾ Copy to Clipboard

```
1 var settings = {
2   "async": true,
3   "crossDomain": true,
4   "url": "http://localhost:57571/api/products/",
5   "method": "GET",
6   "headers": {
7     "cache-control": "no-cache",
8     "postman-token": "bcf0f3d4-7ed7-012f-06fc-5b373ddb299e"
9   }
10 }
11
12 $.ajax(settings).done(function (response) {
13   console.log(response);
14 });
```

GET http://localhost:57571/api/products

Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (6) Test Results Status: 200 OK Time: 55 ms

content-type → application/json; charset=utf-8

date → Wed, 14 Mar 2018 11:14:55 GMT

server → Kestrel

transfer-encoding → chunked

x-powered-by → ASP.NET

x-sourcefiles → =?UTF-8?B?RTpcQmFja3VwXERlc2t0b3BcUGFja3RcUkVTVGZ1bCBTZXJ2aWNlcy8pbiAuTkVUIENvcuUgMi4wXFByb2piY3RzXERlbW9FQ29tbWVvY2V8cHBcRGVtb0VDb21tZXJzU=

POST http://localhost:57571/api/products

Params Send Save

```
1 {
2   "Name": "Noddles",
3   "Price": 12.9
4 }
```

Data Passed to API Action

Body Cookies Headers (7) Test Results Status: 201 Created Time: 70 ms

content-type → application/json; charset=utf-8

date → Sun, 18 Mar 2018 19:08:35 GMT

location → api/products/1

server → Kestrel

transfer-encoding → chunked

x-powered-by → ASP.NET

x-sourcefiles → =?UTF-8?B?RTpcQmFja3VwXERlc2t0b3BcUGFja3RcUkVTVGZ1bCBTZXJ2aWNlcy8pbiAuTkVUIENvcuUgMi4wXFByb2piY3RzXERlbW9FQ29tbWVvY2V8cHBcRGVtb0VDb21tZXJzU=

POST ▼ http://localhost:57571/api/products

Authorization Headers (1) Body ● Pre-request Script Tests

Key	Value
<input checked="" type="checkbox"/> Content-Type	application/json
New key	Value

http://localhost:57571 ● http://localhost:57571 ● http://localhost:57571 ● + ... No Environment ▼ 👁 ⚙

PUT ▼ http://localhost:57571/api/products/1 Params Send ▼ Save ▼

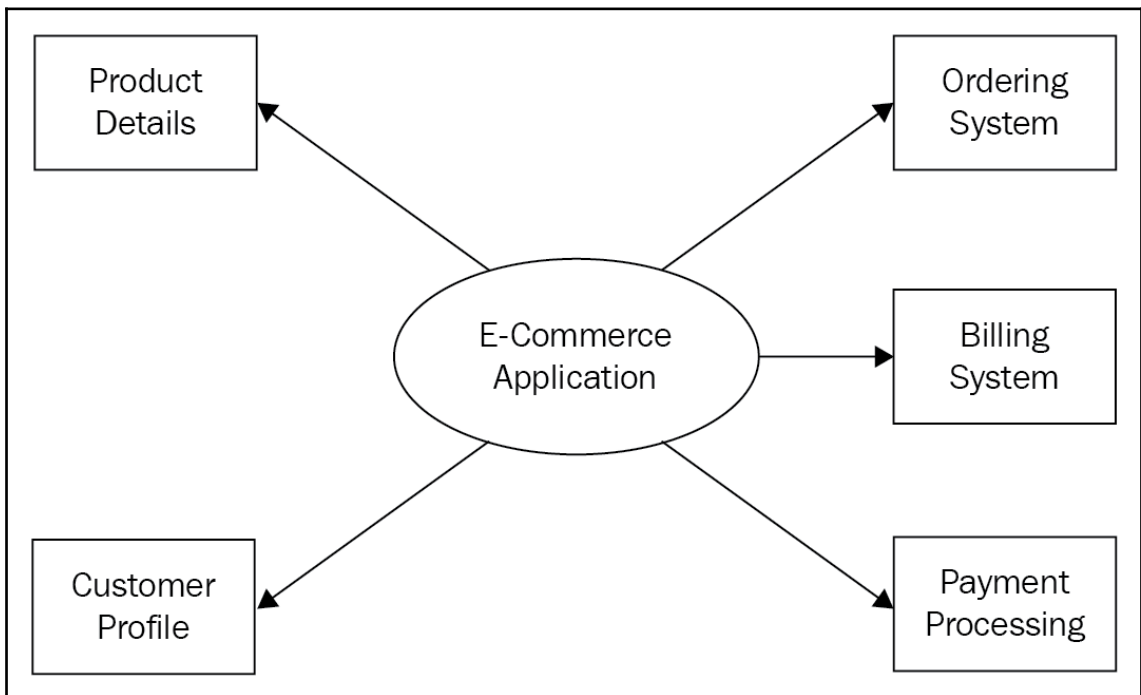
Authorization Headers (1) Body ● Pre-request Script Tests Code

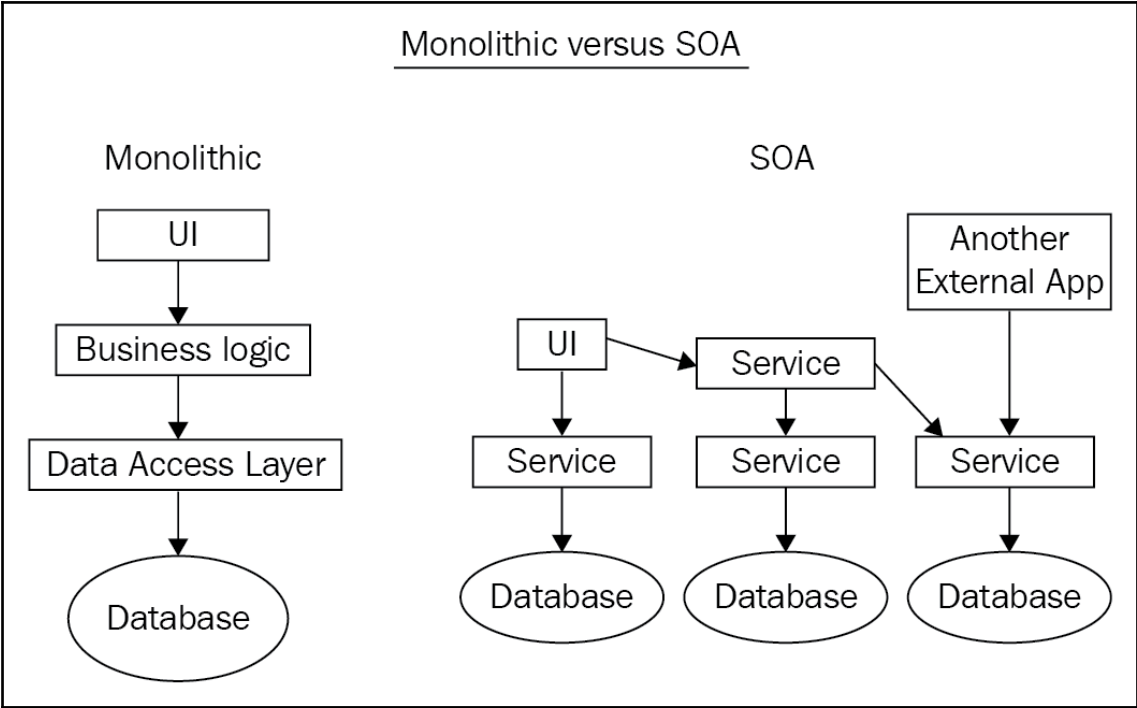
form-data x-www-form-urlencoded raw binary JSON (application/json) ▼

```
1 {  
2   "Name": "Noddles",  
3   "Price": 12.9  
4 }
```

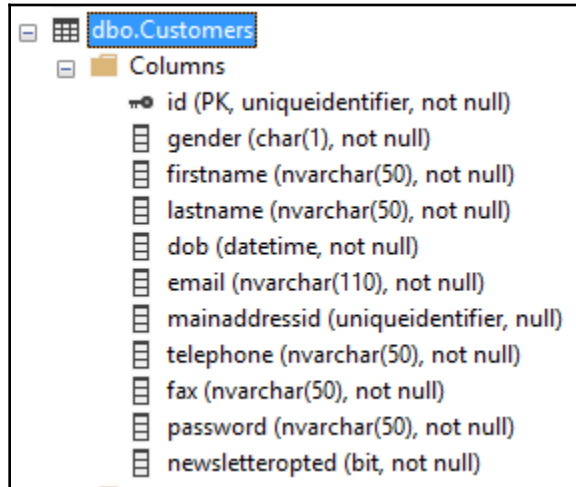
Body Cookies Headers (5) Test Results Status: 200 OK Time: 67 ms



The screenshot shows a web browser's developer tools interface. At the top, there are three tabs for the URL `http://localhost:57571`. The main area displays a `DELETE` request to `http://localhost:57571/api/products/1`. The `Headers` tab is selected, showing five headers: `content-length` (0), `date` (Tue, 20 Mar 2018 15:09:27 GMT), `server` (Kestrel), `x-powered-by` (ASP.NET), and `x-sourcefiles` (=?UTF-8?B?RTpcQmFja3VwXERlc2t0b3BcUGFja3RcUkVTVGZ1bCBTZXJ2aWNLcyBpbIAuTKVUIENvcml4wXFByb2piY3RzXERlbW9FGQ29tbWVvY2V2bCkHbCtRGVtb0VDb21tZXJjZU=). The status bar at the bottom right indicates a `200 OK` response with a time of `22 ms`.





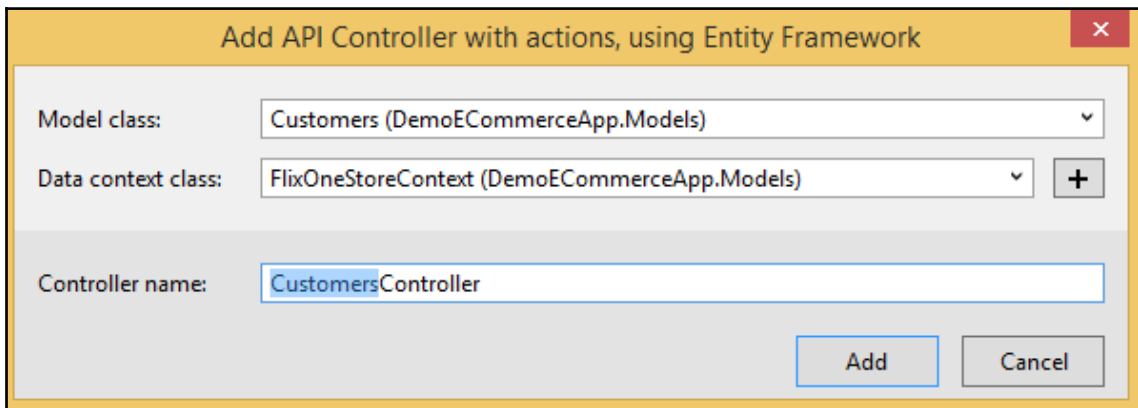
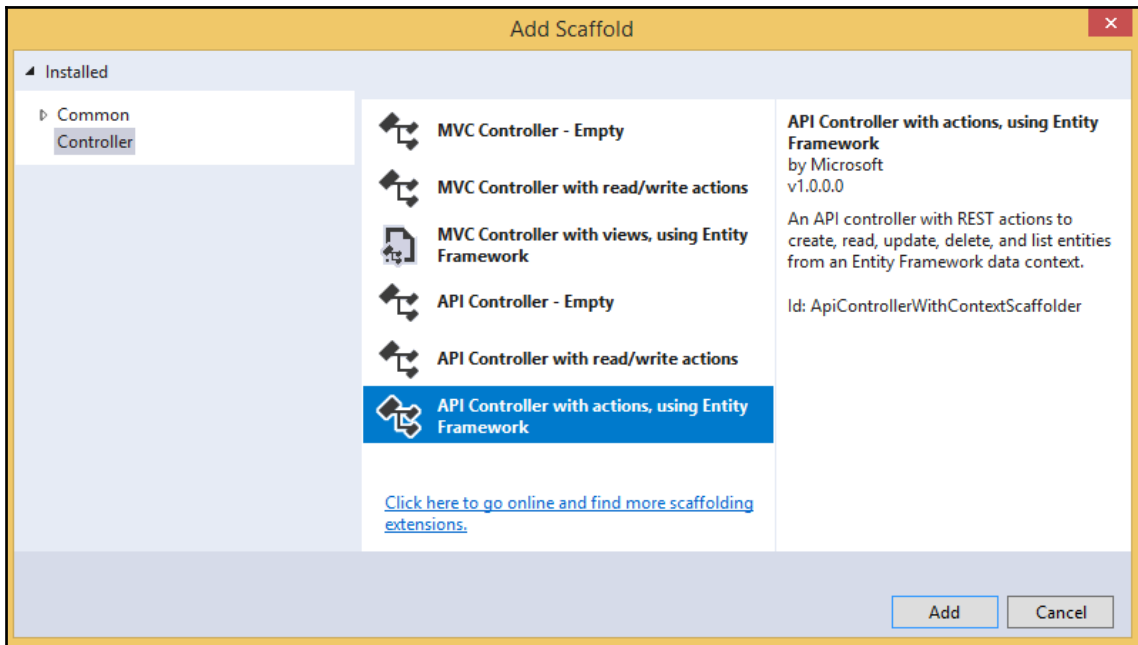


Chapter 3: User Registration and Administration



 **Microsoft.EntityFrameworkCore.SqlServer**  by Microsoft, **5M** downloads v2.0.2
Microsoft SQL Server database provider for Entity Framework Core.

 **Microsoft.EntityFrameworkCore.Tools**  by Microsoft, **3.88M** downloads v2.0.2
Entity Framework Core Package Manager Console Tools. Includes Scaffold-DbContext, Add-Migration, and Update-Database.



Register for FlixOneStore

From this page
A customer record created for: Tadit Dash

First Name:

Last Name:

Email:

Gender: Male Female

Date of Birth:

Mobile Number:

Fax:

Password:

Confirm Password:

```
$.ajax({
  url: 'http://localhost:57571/api/Customers',
  type: "POST",
  contentType: "application/json",
  data: JSON.stringify(customer),
  dataType: "json",
  success: function (result) { result = {id: "910d4c2f-b394-4578-
    alert("A customer record created for: "
      + result.);
  },
  error: function (err) {
    alert(err.responseText);
  }
});
```

"container">
ister for FlixOneStore
5

Object

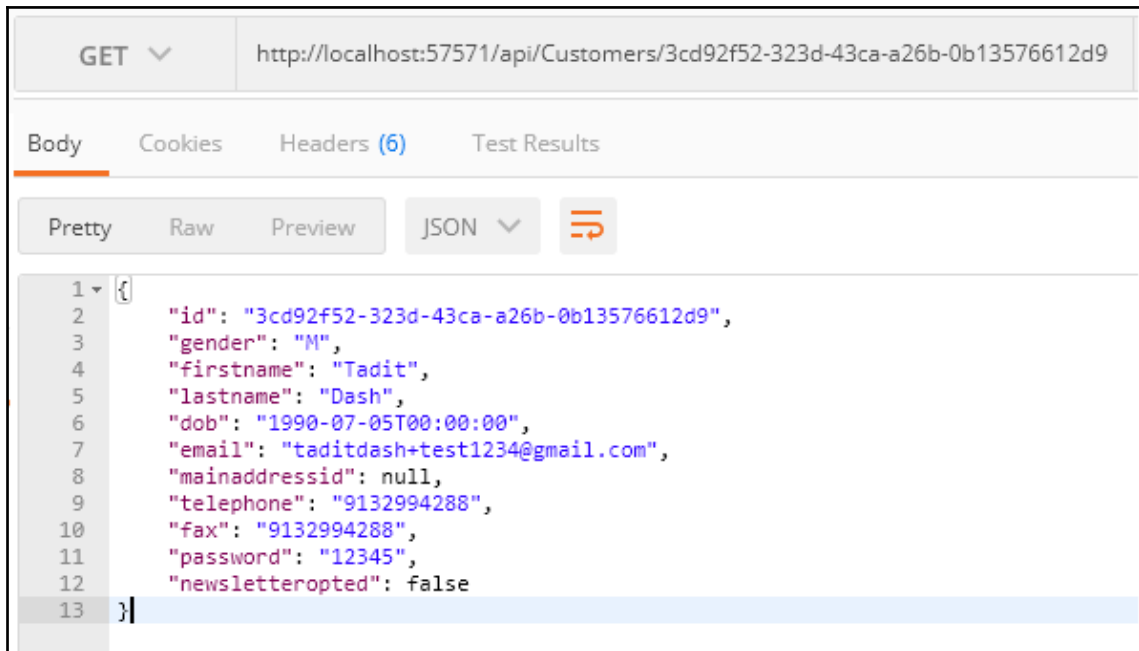
- dob: "1990-07-05T00:00:00"
- email: "taditdash@gmail.com"
- fax: "9132994288"
- firstname: "Tadit"
- gender: "M"
- id: "910d4c2f-b394-4578-8d9c-7ca3fd3266e2"
- lastname: "Dash"
- mainaddressid: null
- newsletteropted: false
- password: "12345"
- telephone: "9132994288"
- __proto__: Object

▼ General

Request URL: http://localhost:57571/api/Customers
Request Method: POST
Status Code: ● 201 Created
Remote Address: [::1]:57571
Referrer Policy: no-referrer-when-downgrade

▼ Response Headers [view source](#)

Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Date: Tue, 01 May 2018 02:54:02 GMT
Location: http://localhost:57571/api/Customers/3cd92f52-323d-43ca-a26b-0b13576612d9
Server: Kestrel
Transfer-Encoding: chunked
X-Powered-By: ASP.NET



The screenshot shows a web browser's developer tools interface. At the top, a GET request is shown for the URL `http://localhost:57571/api/Customers/3cd92f52-323d-43ca-a26b-0b13576612d9`. Below the URL bar, there are tabs for 'Body', 'Cookies', 'Headers (6)', and 'Test Results'. The 'Body' tab is selected, and the response is displayed in a 'Pretty' JSON view. The JSON object contains the following fields:

```
1 {  
2   "id": "3cd92f52-323d-43ca-a26b-0b13576612d9",  
3   "gender": "M",  
4   "firstname": "Tadit",  
5   "lastname": "Dash",  
6   "dob": "1990-07-05T00:00:00",  
7   "email": "taditdash+test1234@gmail.com",  
8   "mainaddressid": null,  
9   "telephone": "9132994288",  
10  "fax": "9132994288",  
11  "password": "12345",  
12  "newsletteropted": false  
13 }
```

Register for FlixOneStore

From this page
{"email":["User with mail id already exists!"]}

First Name:

Last Name:

Email:

Gender: Male Female



Date of Birth:

Mobile Number:

Fax:

Password:

Confirm Password:

 **Microsoft.AspNetCore.Cors**  by Microsoft, **7.64M** downloads
CORS middleware and policy for ASP.NET Core to enable cross-origin resource sharing.
Commonly used types:

My Profile

id	<input type="text" value="910d4c2f-b394-4578-8d9c-7ca3fd3266e2"/>
gender	<input type="text" value="M"/>
firstname	<input type="text" value="Tadit"/>
lastname	<input type="text" value="Dash"/>
dob	<input type="text" value="1990-07-05T00:00:00"/>
email	<input type="text" value="taditdash@gmail.com"/>
mainaddressid	<input type="text" value="null"/>
telephone	<input type="text" value="9132994288"/>
fax	<input type="text" value="9132994288"/>
password	<input type="text" value="12345"/>
newsletteropted	<input type="text" value="false"/>

▼ **General**


Request URL: http://localhost:57571/api/Customers/910D4C2F-B394-4578-8D9C-7CA3FD3266E2
Request Method: GET
Status Code: ● 401 Unauthorized
Remote Address: [::1]:57571
Referrer Policy: no-referrer-when-downgrade


▼ **Response Headers** [view source](#)

Access-Control-Allow-Origin: *
Content-Length: 0
Date: Sat, 05 May 2018 14:07:23 GMT
Server: Kestrel
WWW-Authenticate: Basic realm="http://localhost:57571", charset="UTF-8"
X-Powered-By: ASP.NET
X-SourceFiles: =?UTF-8?B?RTpcQmFja3VwXERlc2t0b3BcUGFja3RcUkVTVGZ1bCBTZXJ2aWwN1cyBpb1AuTkVU...
cHBcRGVtb0VDb21tZXJjZUFwcFxcG1cQ3VzdG9tZXJzXDkxMEQ0QzJGLUIzOTQtNDU3OC04RD1DLTddQTNGRDMy

From this page

Either wrong email and password or you are not authorized to access the data!

 **IdentityServer4** by Brock Allen, Dominick Baier v2.2.0
OpenID Connect and OAuth 2.0 Framework for ASP.NET Core

 **IdentityServer4.AccessTokenValidation** by Brock Allen, Dominick Baier v2.6.0
ASP.NET Core authentication handler to validate JWT and reference tokens from IdentityServer4

GET http://localhost:57571/api/Customers/910D4C2F-B394-4578-8D9C-7CA3FD3266E2

Authorization Headers Body Pre-request Script Tests

Key	Value	Description
New key	Value	Description

Body Cookies Headers (6) Test Results

Status: 401 Unauthorized

content-length → 0

date → Mon, 07 May 2018 17:55:59 GMT

server → Kestrel

www-authenticate → Bearer

x-powered-by → ASP.NET

x-sourcefiles → =?UTF-8?B?RTpcQmFja3VwXERlc2t0b3BcUGFja3RcUkVTVGZ1bC...=

POST http://localhost:57571/connect/token

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary Text

1 grant_type=password&scope=FlixOneStore.ReadAccess&client_id=htmlClient&client_secret=secret&username=taditdash@gmail.com&password=12345

Body Cookies Headers (8) Test Results

Status: 400 Bad Request

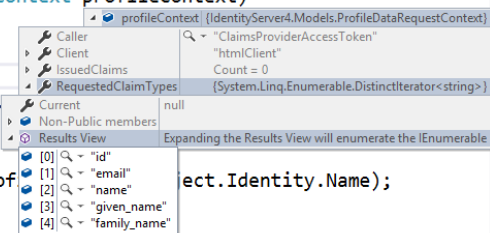
Pretty Raw Preview JSON

```
1 {
2   "error": "invalid_client"
3 }
```



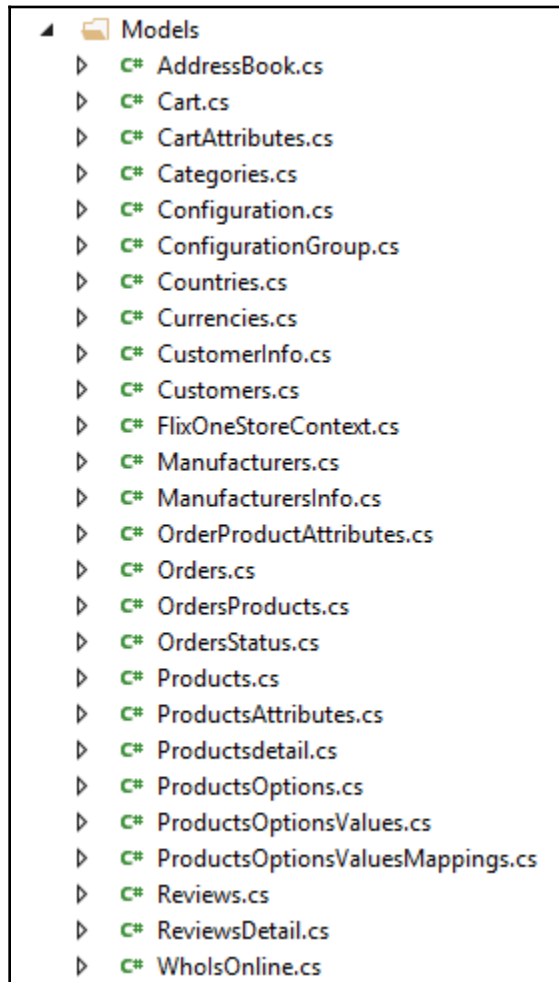
```
0 references | 0 exceptions
public async Task GetProfileDataAsync(ProfileDataRequestContext profileContext)
{
    try
    {
        ≤ 2,410ms elapsed
        if (!string.IsNullOrEmpty(profileContext.Subject.Identity.Name))
        {
            var customer = await _context.Customers
                .SingleOrDefaultAsync(m => m.Email == profileContext.Identity.Name);

            if (customer != null)
                return customer;
        }
    }
}
```



```
{
  "nbf": 1525716685,
  "exp": 1525720285,
  "iss": "http://localhost:57571",
  "aud": [
    "http://localhost:57571/resources",
    "FlixOneStore.ReadAccess"
  ],
  "client_id": "htmlClient",
  "sub": "910d4c2f-b394-4578-8d9c-7ca3fd3266e2",
  "auth_time": 1525716684,
  "idp": "local",
  "id": "910d4c2f-b394-4578-8d9c-7ca3fd3266e2",
  "name": "Tadit Dash",
  "given_name": "Tadit",
  "family_name": "Dash",
  "email": "taditdash@gmail.com",
  "scope": [
    "FlixOneStore.ReadAccess"
  ],
  "amr": [
    "database"
  ]
}
```

Chapter 4: Item Catalogue, Cart, and Checkout



```
GET http://localhost:57571/api/Productsdetails Params Send Save  
Body Cookies Headers (6) Test Results Status: 200 OK Time: 16431 m  
Pretty Raw Preview JSON  
1 [ {  
2   {  
3     "id": "9a93a82a-3952-435e-9091-2ae92dd3e1d6",  
4     "productId": "9f35bc15-c6c3-4afa-8683-97513f5ae077",  
5     "name": "Building Microservices with .NET Core 2.0 - Second Edition",  
6     "description": "Transitioning monolithic architectures using microservices with .NET Core 2.0 using C# 7.0",  
7     "url": "https://www.amazon.com/Building-Microservices-.NET-Core-Transitioning/dp/1788393333/ref=tmm_pap_swatch_0?_encoding=UTF8&qid=&sr=",  
8     "views": 10000,  
9     "product": null  
10  },  
11  },  
12  {  
13    "id": "c96ac991-6581-4675-b00c-439df3961f03",  
14    "productId": "98a95bb6-c573-450d-a470-0a637e126dd7",  
15    "name": "Dependency Injection in .NET Core 2.0",  
16    "description": "Make use of constructors, parameters, setters, and interface injection to write reusable and loosely-coupled code",  
17    "url": "https://www.amazon.com/Dependency-Injection-.NET-Core-loosely-coupled/dp/1787121305/ref=tmm_pap_swatch_0?_encoding=UTF8&qid=  
18    =1510939068&sr=8-3",  
19    "views": 5000,  
20    "product": null  
21  },  
22  ],  
23  ],
```

FlixOneStore Home

Search

Building Microservices with .NET Core 2.0 - Second Edition
Transitioning monolithic architectures using microservices with .NET Core 2.0 using C# 7.0
[Amazon Link](#) [Add To Cart](#)

Dependency Injection in .NET Core 2.0
Make use of constructors, parameters, setters, and interface injection to write reusable and loosely-coupled code
[Amazon Link](#) [Add To Cart](#)

Learn C# in 7 days
Get up and running with C# 7 with async main, tuples, pattern matching, LINQ, regex, indexers, and more
[Amazon Link](#) [Add To Cart](#)

Building RESTful Web services with .NET Core
Leveraging ASP.NET Core MVC and Visual Studio 17
[Amazon Link](#) [Add To Cart](#)

FlixOneStore Home

Search

Dependency Injection in .NET Core 2.0
Make use of constructors, parameters, setters, and interface injection to write reusable and loosely-coupled code
[Amazon Link](#) **Price: \$49.99** [Add To Cart](#)

Building RESTful Web services with .NET Core
Leveraging ASP.NET Core MVC and Visual Studio 17
[Amazon Link](#) **Price: \$39.99** [Add To Cart](#)

Learn C# in 7 days
Get up and running with C# 7 with async main, tuples, pattern matching, LINQ, regex, indexers and more
[Amazon Link](#) **Price: \$39.99** [Add To Cart](#)

Building Microservices with .NET Core 2.0 - Second Edition
Transitioning monolithic architectures using microservices with .NET Core 2.0 using C# 7.0
[Amazon Link](#) **Price: \$44.99** [Add To Cart](#)

FlixOneStore Home

building

Building RESTful Web services with .NET Core
Leveraging ASP.NET Core MVC and Visual Studio 17 **Price: \$39.99**
[Amazon Link](#)

Building Microservices with .NET Core 2.0 - Second Edition
Transitioning monolithic architectures using microservices with .NET Core 2.0 using C# 7.0 **Price: \$44.99**
[Amazon Link](#)

FlixOneStore Home

Enter text to search

Dependency Injection in .NET Core 2.0
Make use of constructors, parameters, setters, and interface injection to write reusable and loosely-coupled code **Price: \$49.99**
[Amazon Link](#)

Building RESTful Web services with .NET Core
Leveraging ASP.NET Core MVC and Visual Studio 17 **Price: \$39.99**
[Amazon Link](#)

My Cart

Product	Quantity	Price	
Dependency Injection in .NET Core 2.0	2	\$99.98	<input type="button" value="Remove"/>
Building RESTful Web services with .NET Core	1	\$39.99	<input type="button" value="Remove"/>

```
function PostCart(customerId, productId, qty, finalPrice) {
    var cart = {
        Customerid: customerId,
        Productid: productId,
        Qty: qty,
        Finalprice: finalPrice
    };

    $.ajax({
        url: 'http://localhost:57571/api/Carts',
        type: "POST",
        contentType: "application/json",
        dataType: "json",
        data: JSON.stringify(cart),
        success: function (result) {
            console.log(result);
        },
        error: function (message) {
            console.log(message.statusText);
        }
    });
}
```

"Unauthorized"

```
<table id="tblCart" class="table">
  <thead>...</thead>
  <tbody>
    <tr data-product-id="98a95bb6-c573-450d-a470-0a637e126dd7" data-cart-id="a8b64589-7e2e-4eab-8940-6865d9d32bda"> == $0
      <td>Dependency Injection in .NET Core 2.0</td>
      <td class="qty">1</td>
      <td class="price">$49.99</td>
      <td>...</td>
    </tr>
  </tbody>
</table>
```

id	customerid	productid	qty	finalprice	addledon	
1	11FD1B87-7B9A-4C4C-B070-3E34F72D8827	910D4C2F-B394-4578-8D9C-7CA3FD3266E2	98A95BB6-C573-450D-A470-0A637E126DD7	4	199.96	2018-05-

Place Order ×

Shipping Details

Street Address

City

State

Postal Code

Country

Place Order ×

Shipping Details

Street Address

This page says
Order Placed Successfully.

Column Name	Data Type	Nullability
id	uniqueidentifier	not null
customerid	uniqueidentifier	null
customername	nvarchar(65)	not null
customerstreetaddress	nvarchar(65)	not null
customercity	nvarchar(50)	not null
customerstate	nvarchar(50)	not null
customerpostalcode	nvarchar(50)	not null
customercountry	nvarchar(50)	not null
customertelephone	nvarchar(50)	not null
customeremail	nvarchar(110)	not null
deliveryname	nvarchar(65)	not null
deliverystreetaddress	nvarchar(65)	not null
deliverycity	nvarchar(50)	not null
deliverystate	nvarchar(50)	not null
deliverypostalcode	nvarchar(50)	not null
deliverycountry	nvarchar(50)	not null
paymentmethodid	uniqueidentifier	null
lastmodified	datetime	not null
datepurchased	datetime	not null
shippingcost	decimal(10,2)	null
shippingmethodid	uniqueidentifier	null
orderstatus	nvarchar(10)	not null
orderdatefinished	datetime	not null
comments	nvarchar(max)	null
currency	nvarchar(5)	not null
currency_value	decimal(16,6)	not null

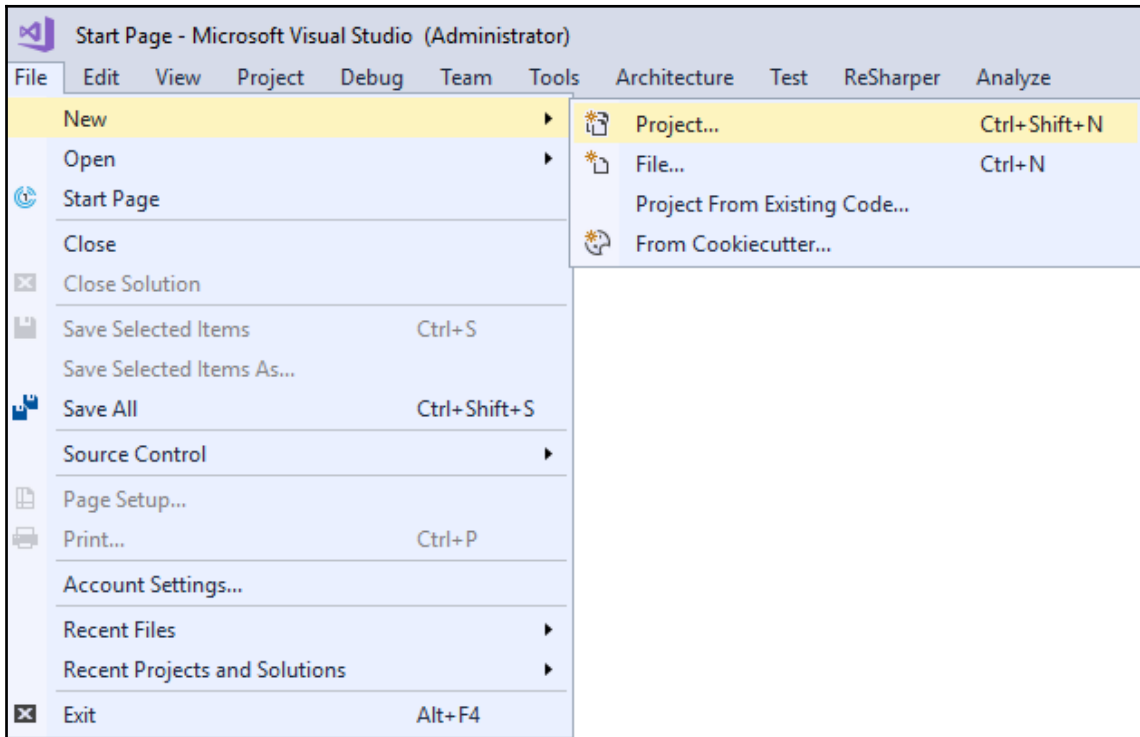
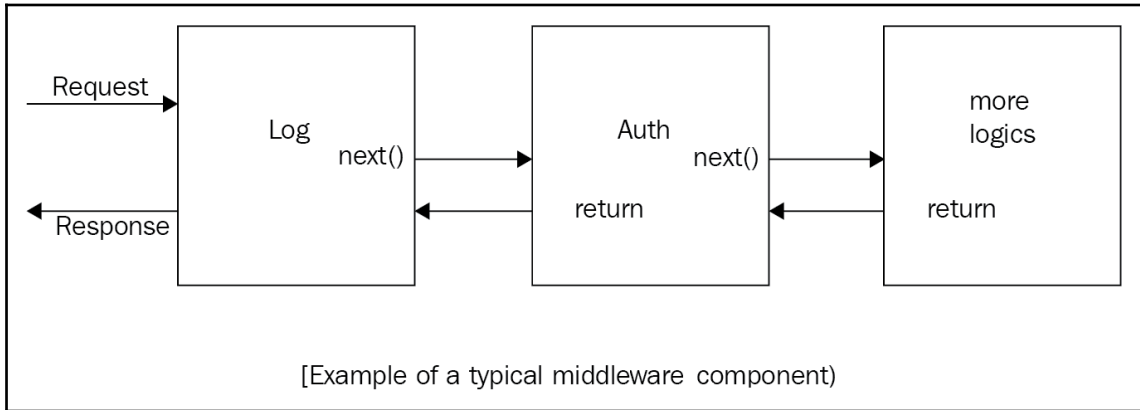
GET ▼ http://localhost:57571/api/Orders/0B4058FD-A6BB-42BC-BC0E-305EDF8704A6

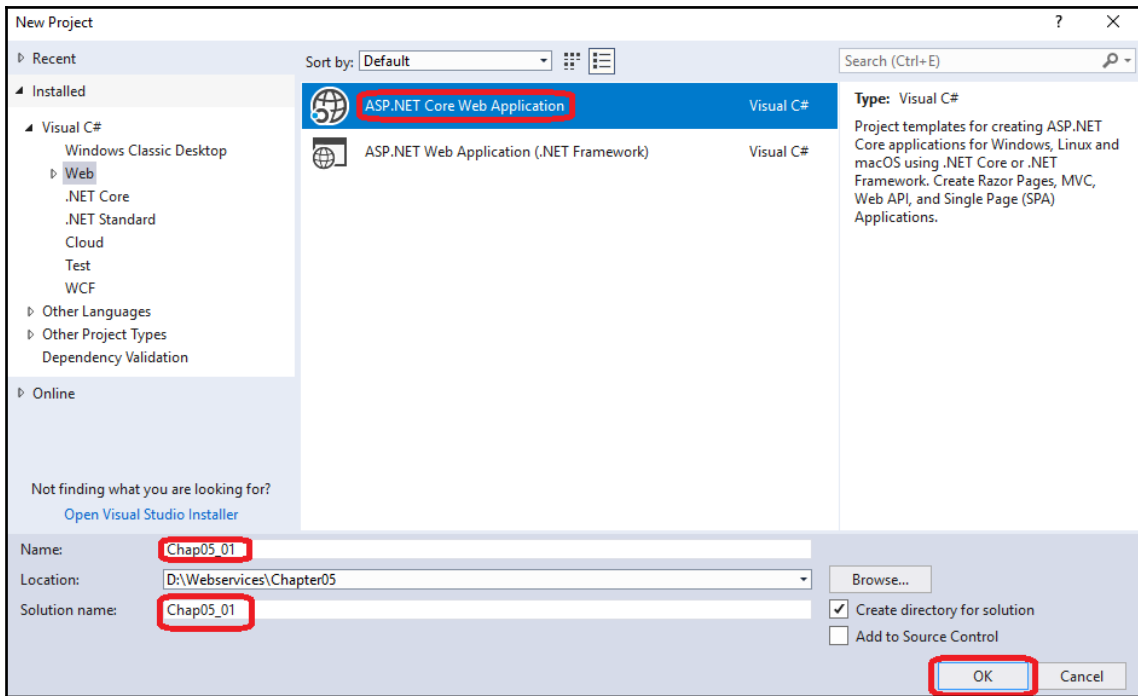
Body Cookies Headers (6) Test Results

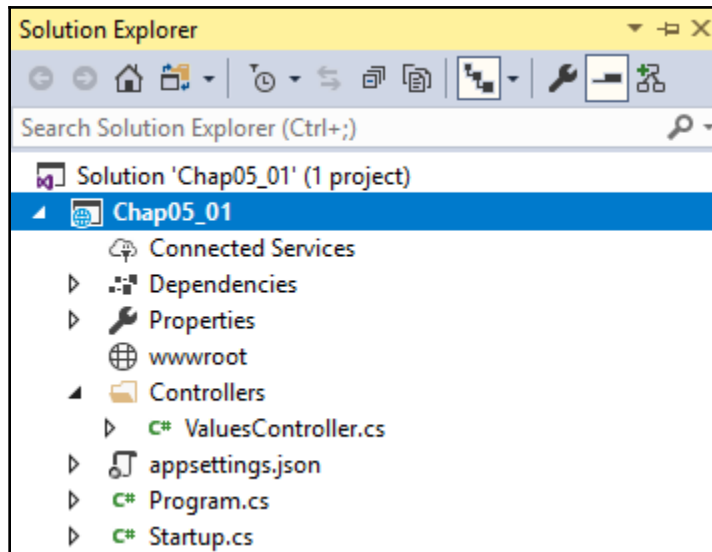
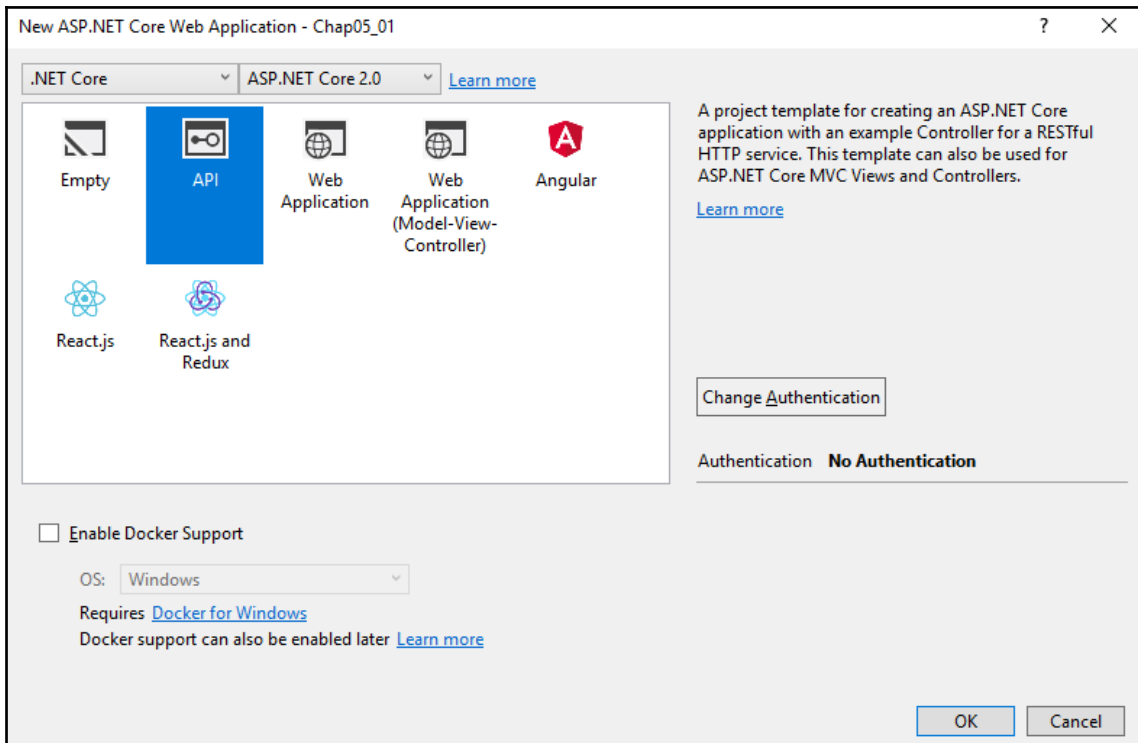
Pretty Raw Preview JSON ▼

```
17     "deliverycountry": "India",
18     "paymentmethodid": "aldf61de-d9a0-4ca9-8cc2-26a16e8bfe14",
19     "lastmodified": "2018-05-17T17:53:11.177",
20     "datepurchasded": "2018-05-17T17:53:11.177",
21     "shippingcost": 0,
22     "shipingmethodid": "65bf649d-f6d2-41d8-a30e-3d6d8d00882f",
23     "orderstatus": "Placed",
24     "orderdatefinished": "2018-05-17T17:53:11.177",
25     "comments": null,
26     "currency": "$",
27     "currencyValue": 0,
28     "customer": null,
29     "orderProductAttributes": [],
30     "ordersProducts": [
31     {
32         "id": "64c24ba5-791c-46d0-9d05-aec88c4f6540",
33         "orderid": "0b4058fd-a6bb-42bc-bc0e-305edf8704a6",
34         "productid": "ee41beee-9800-4bee-ac84-20f1d32c4a53",
35         "productname": "Building RESTful Web services with .NET Core",
36         "productprice": 39.99,
37         "finalprice": 39.99,
38         "productstax": 0,
39         "productqty": 1,
40         "product": null,
41         "orderProductAttributes": []
42     },
43     {
44         "id": "eb76d628-7df6-4663-b1ca-c5947366acef",
45         "orderid": "0b4058fd-a6bb-42bc-bc0e-305edf8704a6",
46         "productid": "98a95bb6-c573-450d-a470-0a637e126dd7",
47         "productname": "Dependency Injection in .NET Core 2.0",
```

Chapter 5: Integrating External Components and Handling





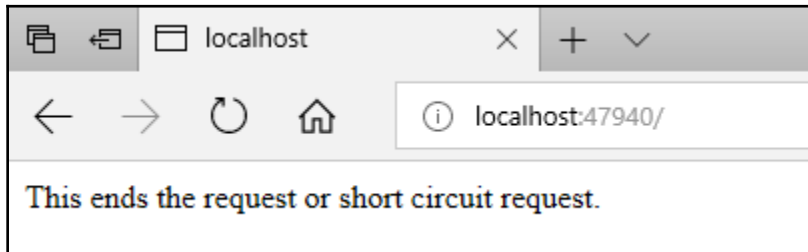



```
(Func<RequestDelegate,RequestDelegate> middleware):IApplicationBuilder  
(this IApplicationBuilder app, Func<HttpContext,Func<Task>,Task>  
middleware):IApplicationBuilder
```

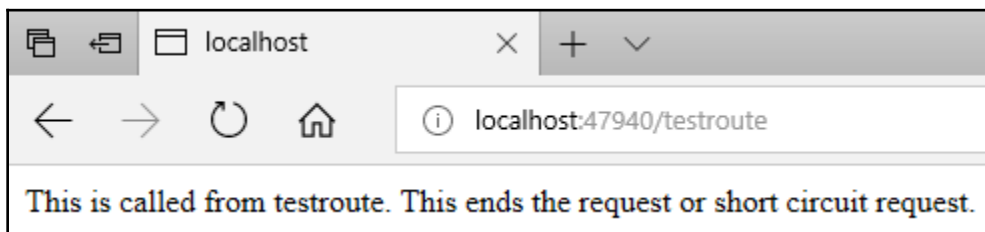
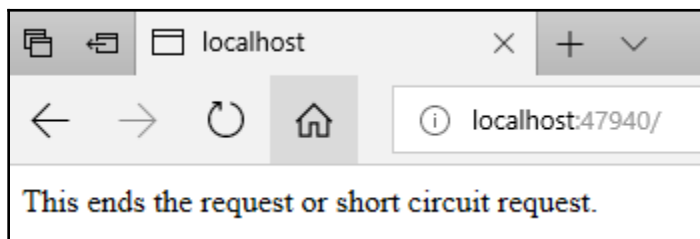
Adds a middleware delegate defined in-line to the application's request pipeline.
middleware: A function that handles the request or calls the given next function.

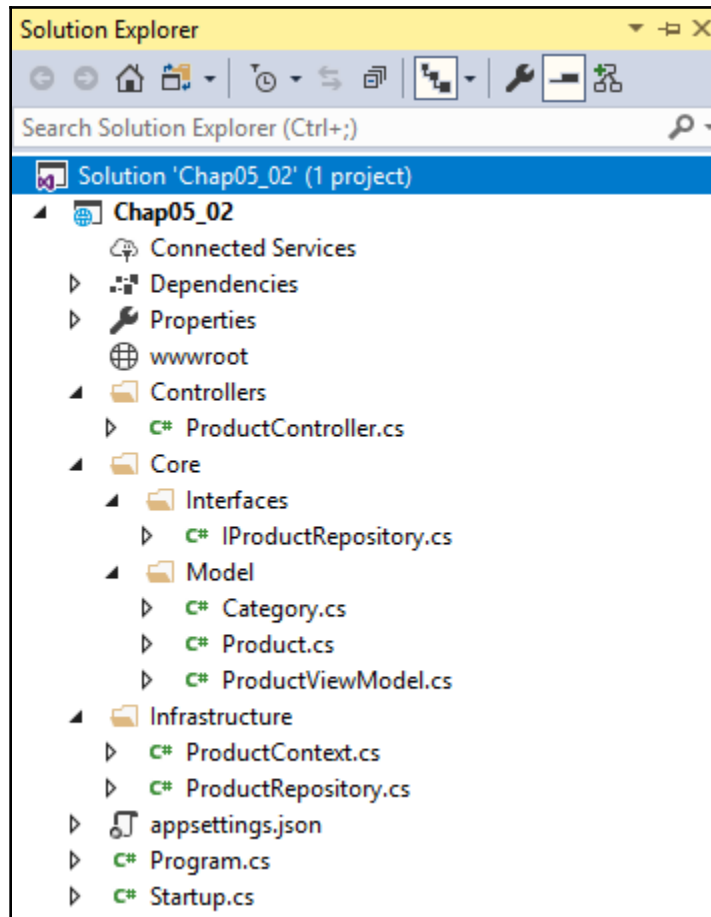
```
(this IApplicationBuilder app, RequestDelegate handler):void
```

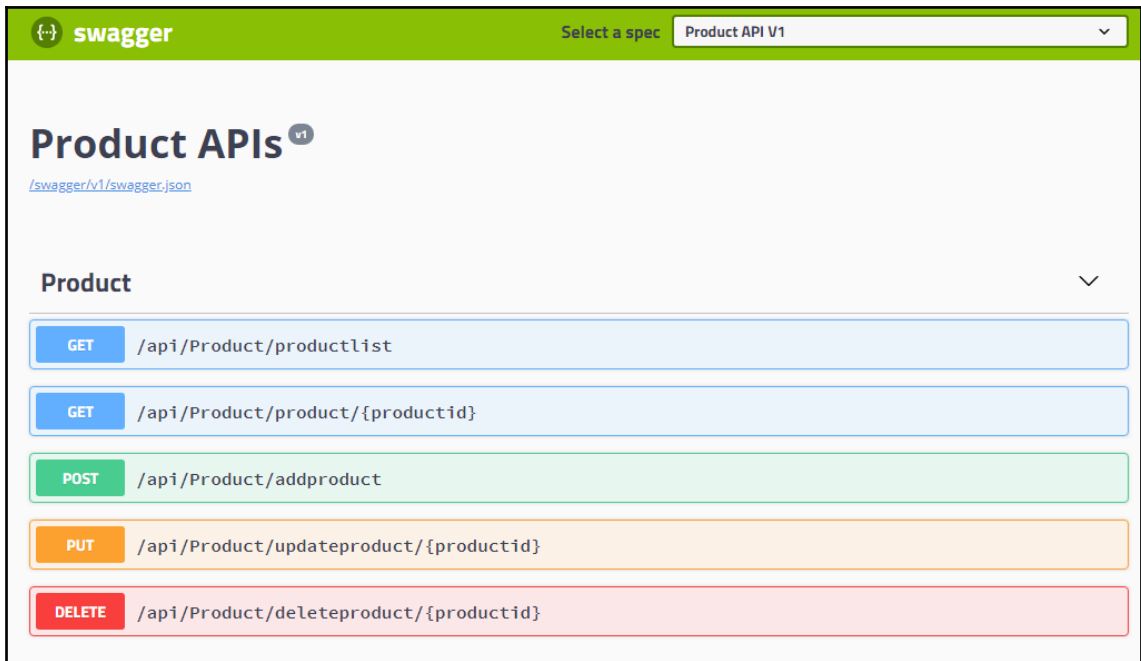
Adds a terminal middleware delegate to the application's request pipeline.
handler: A delegate that handles the request.



 (extension) `IApplicationBuilder.IApplicationBuilder.Map(PathString pathMatch, Action<IApplicationBuilder> configuration)`
Branches the request pipeline based on matches of the given request path. If the request path starts with the given path, the branch is executed.







The image shows a Swagger UI interface for an API. At the top, there is a green header with the Swagger logo and the text "swagger". To the right of the header, there is a dropdown menu labeled "Select a spec" with "Product API V1" selected. Below the header, the main content area displays "Product APIs v1" in a large font, with a small "v1" badge. Underneath, there is a link to the Swagger JSON file: "/swagger/v1/swagger.json". A section titled "Product" is expanded, showing a list of API endpoints. Each endpoint is represented by a colored bar indicating the HTTP method: GET (blue), POST (green), PUT (orange), and DELETE (red). The endpoints are: GET /api/Product/productlist, GET /api/Product/product/{productid}, POST /api/Product/addproduct, PUT /api/Product/updateproduct/{productid}, and DELETE /api/Product/deleteproduct/{productid}.

Method	Endpoint
GET	/api/Product/productlist
GET	/api/Product/product/{productid}
POST	/api/Product/addproduct
PUT	/api/Product/updateproduct/{productid}
DELETE	/api/Product/deleteproduct/{productid}

```
Response body

[
  {
    "productId": "02341321-c20b-48b1-a2be-47e67f548f0f",
    "productName": "Microservices for .NET",
    "productDescription": "Microservices for .NET Core",
    "productImage": "microservices.jpeg",
    "productPrice": 651,
    "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f",
    "categoryName": "Books",
    "categoryDescription": "Technical Books"
  },
  {
    "productId": "4d261e4a-a657-4add-a0f6-dde6e1464d55",
    "productName": "Learn C#",
    "productDescription": "Learn C# in 7 days",
    "productImage": "csharp.jpeg",
    "productPrice": 520,
    "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f",
    "categoryName": "Books",
    "categoryDescription": "Technical Books"
  }
]
```

```
D:\Webservices\Chapter05\02 Logging\Chap05_02>dotnet run
Using launch settings from D:\Webservices\Chapter05\02 Logging\Chap05_02\Properties\launchSettings.json...
Hosting environment: Development
Content root path: D:\Webservices\Chapter05\02 Logging\Chap05_02
Now listening on: http://localhost:60431
Application started. Press Ctrl+C to shut down.
```



An unhandled exception occurred while processing the request.

InvalidOperationException: Unable to resolve service for type 'Microsoft.Extensions.Logging.ILogger' while attempting to activate 'Chap05_02.Controllers.ProductController'.

Microsoft.Extensions.Internal.ActivatorUtilities.GetService(IServiceProvider sp, Type type, Type requiredBy, bool isDefaultParameterRequired)

```
D:\Webservices\Chapter05\02 Logging\Chap05_02>dotnet build
Microsoft (R) Build Engine version 15.6.84.34536 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

Restore completed in 87.32 ms for D:\Webservices\Chapter05\02 Logging\Chap05_02\Chap05_02.csproj.
Restore completed in 28.26 ms for D:\Webservices\Chapter05\02 Logging\Chap05_02\Chap05_02.csproj.
Chap05_02 -> D:\Webservices\Chapter05\02 Logging\Chap05_02\bin\Debug\netcoreapp2.0\Chap05_02.dll

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:02.96
```

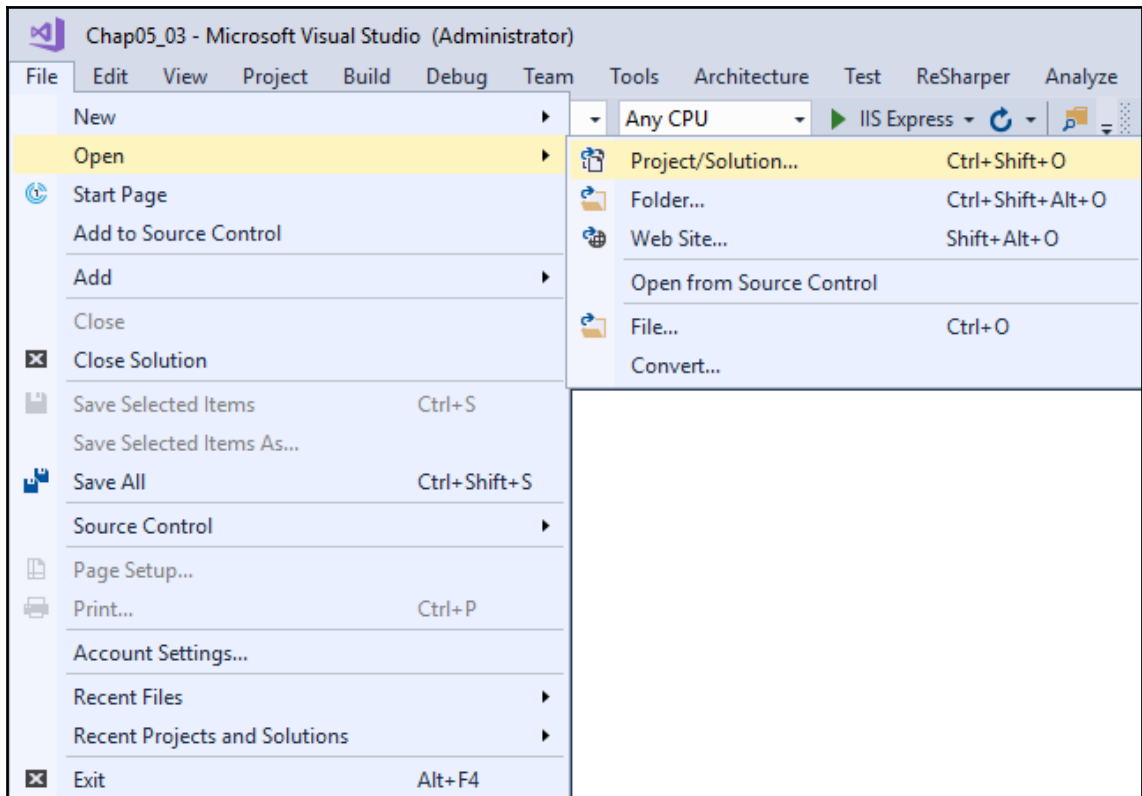
The screenshot shows a REST client interface with the following details:

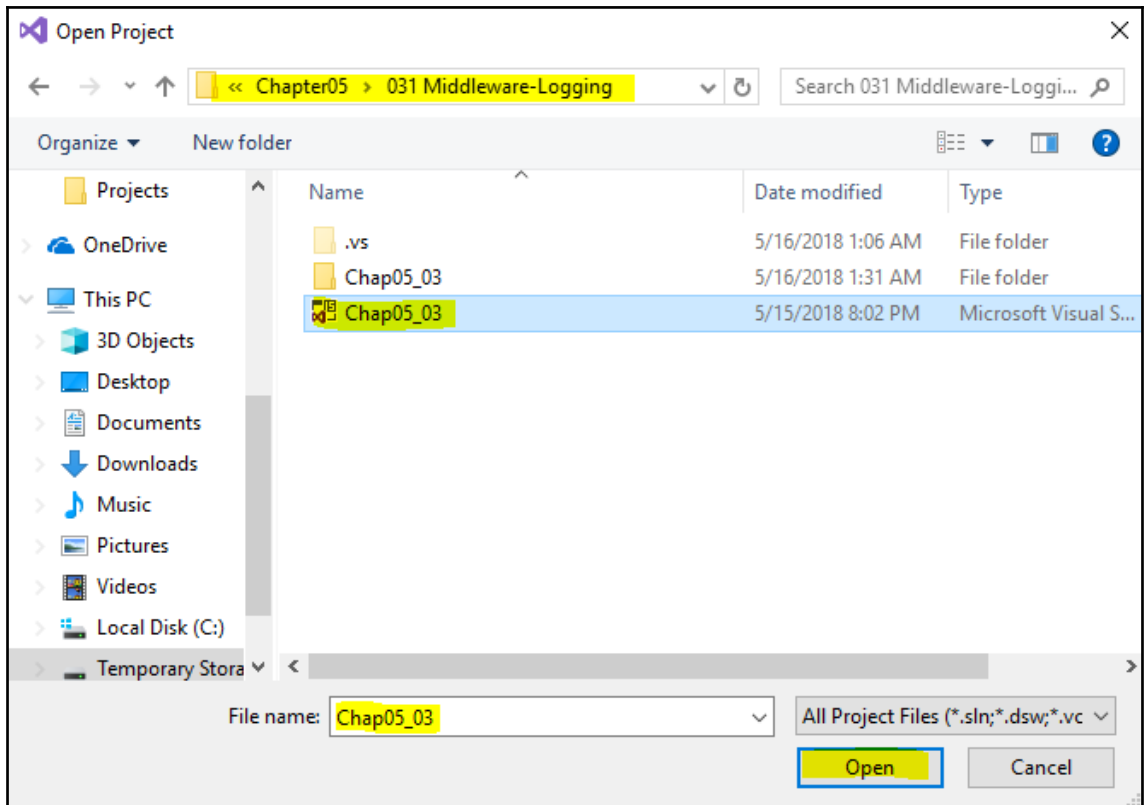
- Method: GET
- URL: http://localhost:60431/api/Product/productlist
- Status: 200 OK
- Time: 1287 ms
- Size: 751 B

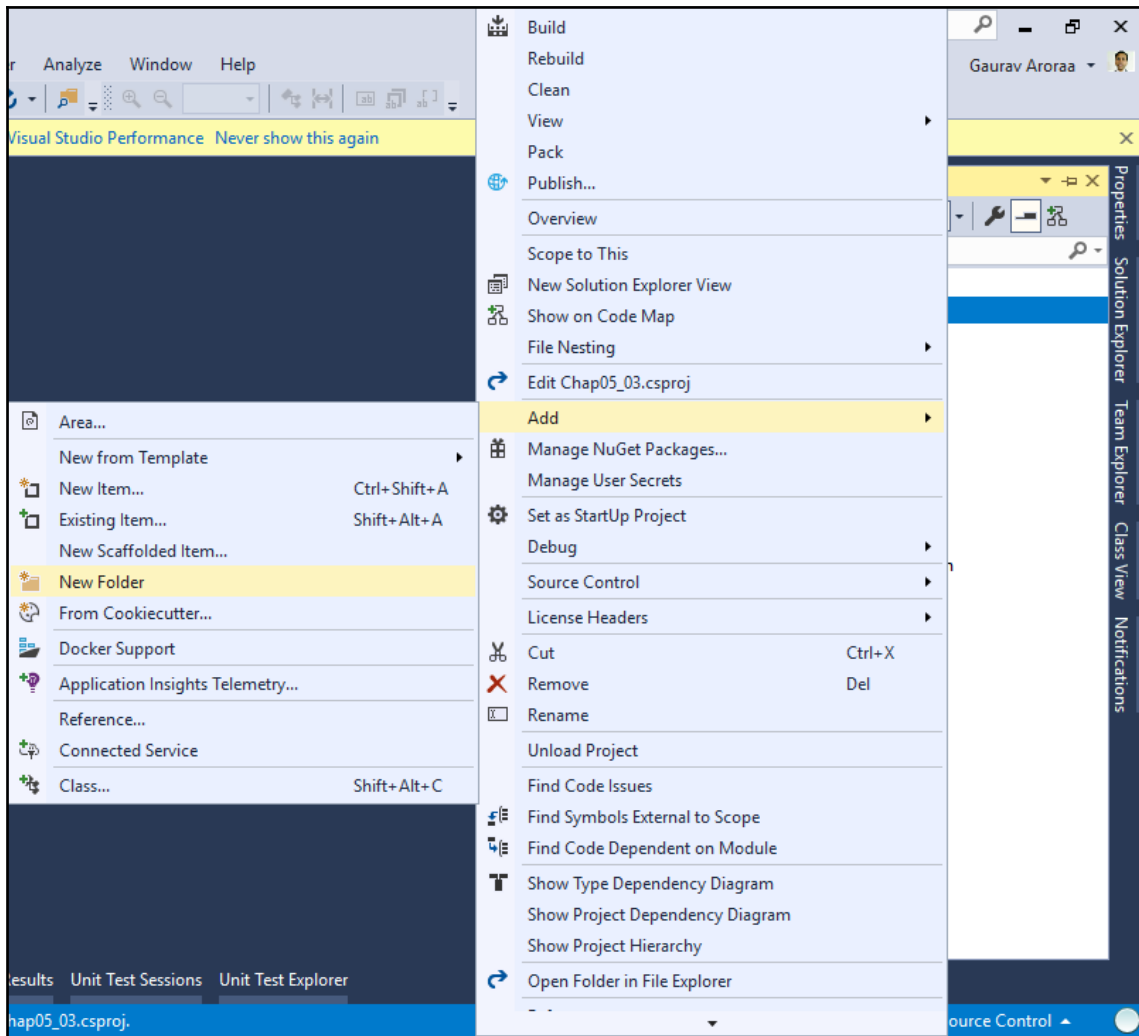
The response body is displayed in JSON format:

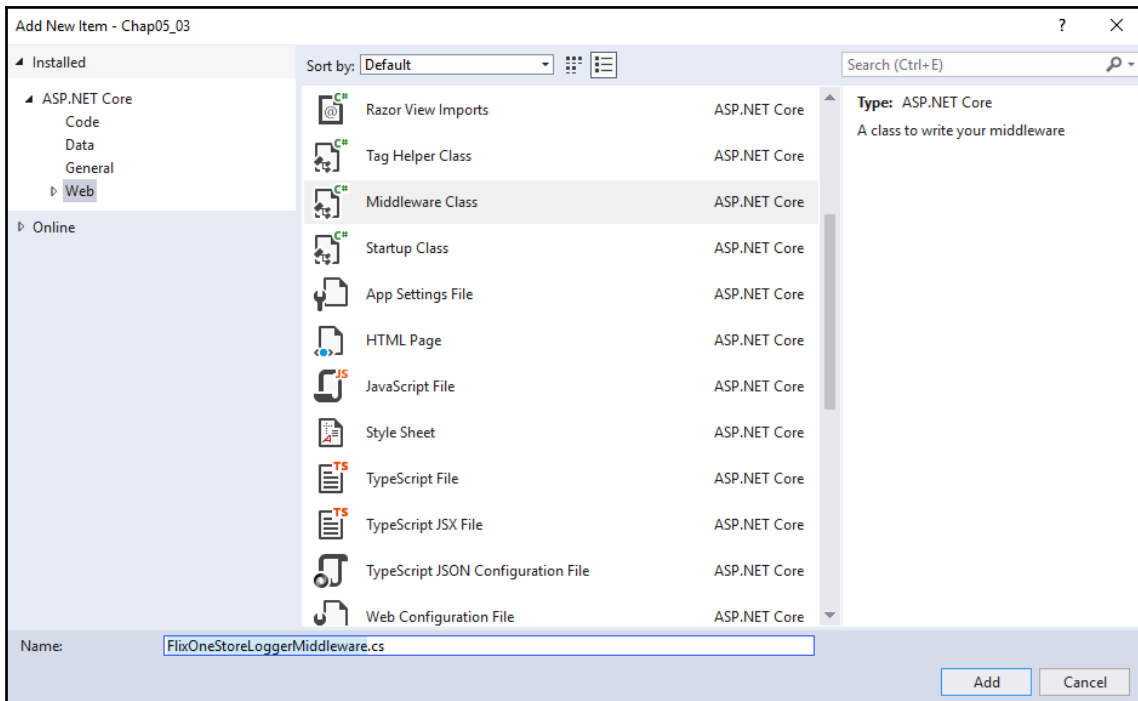
```
[
  {
    "productId": "02341321-c20b-48b1-a2be-47e67f548f0f",
    "productName": "Microservices for .NET",
    "productDescription": "Microservices for .NET Core",
    "productImage": "microservices.jpeg",
    "productPrice": 651,
    "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f",
    "categoryName": "Books",
    "categoryDescription": "Technical Books"
  },
  {
    "productId": "4d261e4a-a657-4add-a0f6-dde6e1464d55",
    "productName": "Learn C#",
    "productDescription": "Learn C# in 7 days",
    "productImage": "csharp.jpeg",
    "productPrice": 520,
    "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f",
    "categoryName": "Books",
    "categoryDescription": "Technical Books"
  }
]
```

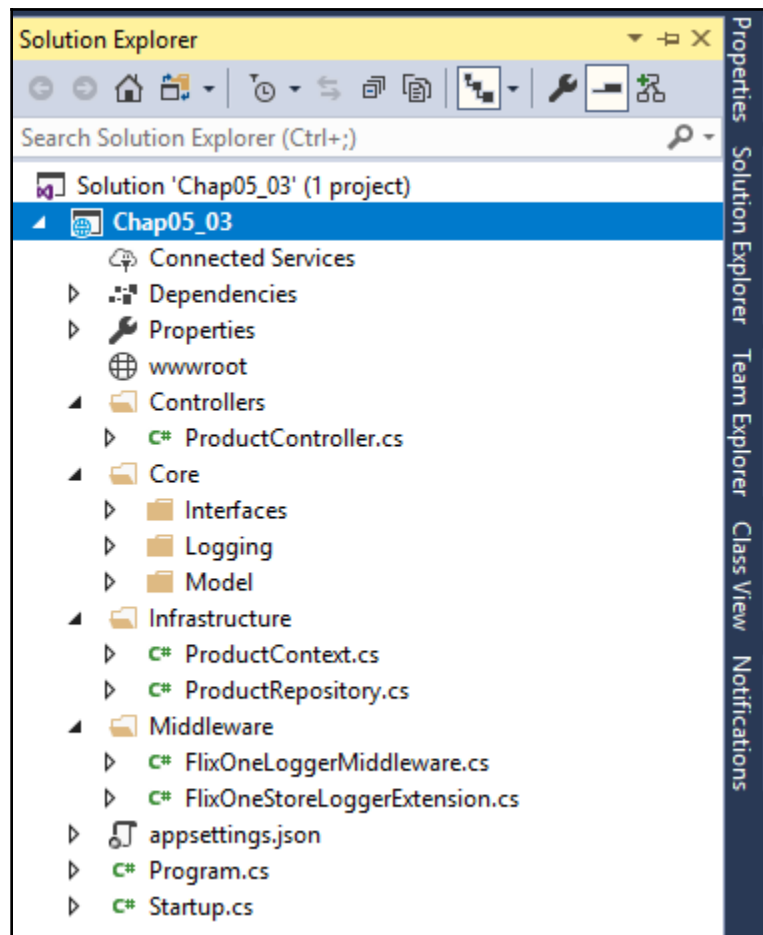
```
D:\Webservices\Chapter05\02 Logging\Chap05_02>dotnet run
Using launch settings from D:\Webservices\Chapter05\02 Logging\Chap05_02\Properties\launchSettings.json...
info: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[0]
      User profile is available. Using 'C:\Users\gaurav\AppData\Local\ASP.NET\DataProtection-Keys' as key repository and Windows DPAPI to encrypt keys at rest.
Hosting environment: Development
Content root path: D:\Webservices\Chapter05\02 Logging\Chap05_02
Now listening on: http://localhost:60431
Application started. Press Ctrl+C to shut down.
info: Microsoft.AspNetCore.Hosting.Internal.WebHost[1]
      Request starting HTTP/1.1 GET http://localhost:60431/api/Product/productlist
info: Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker[1]
      Executing action method Chap05_02.Controllers.ProductController.GetList (Chap05_02) with arguments ((null))
- ModelState is Valid
info: Microsoft.EntityFrameworkCore.Infrastructure[10403]
      Entity Framework Core 2.0.2-rtm-10011 initialized 'ProductContext' using provider 'Microsoft.EntityFrameworkCore.SqlServer' with options: None
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (5ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
      SELECT [c].[Id], [c].[CategoryId], [c].[Description], [c].[Image], [c].[Name], [c].[Price], [c.Category].[Id], [c.Category].[Description], [c.Category].[Name]
      FROM [Products] AS [c]
      INNER JOIN [Categories] AS [c.Category] ON [c].[CategoryId] = [c.Category].[Id]
info: Microsoft.AspNetCore.Mvc.Internal.ObjectResultExecutor[1]
      Executing ObjectResult, writing value Microsoft.AspNetCore.Mvc.ControllerContext.
info: Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker[2]
      Executed action Chap05_02.Controllers.ProductController.GetList (Chap05_02) in 411.8216ms
info: Microsoft.AspNetCore.Hosting.Internal.WebHost[2]
```





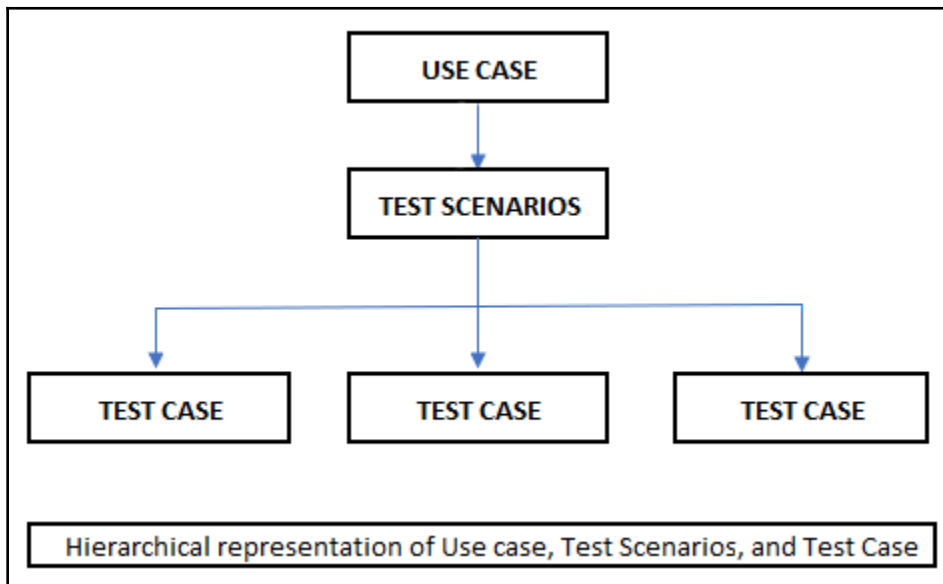
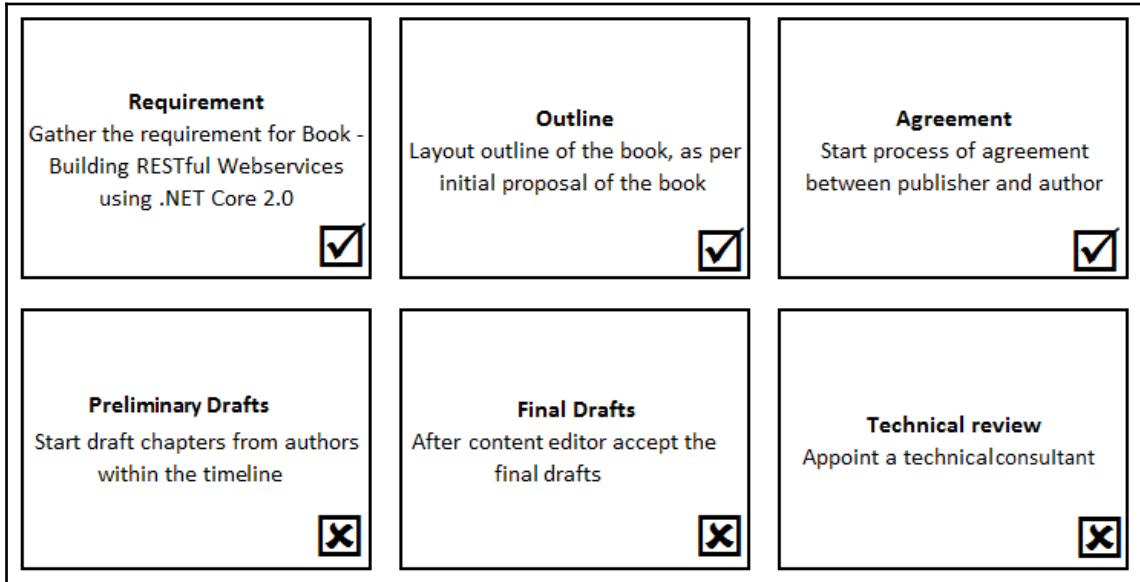


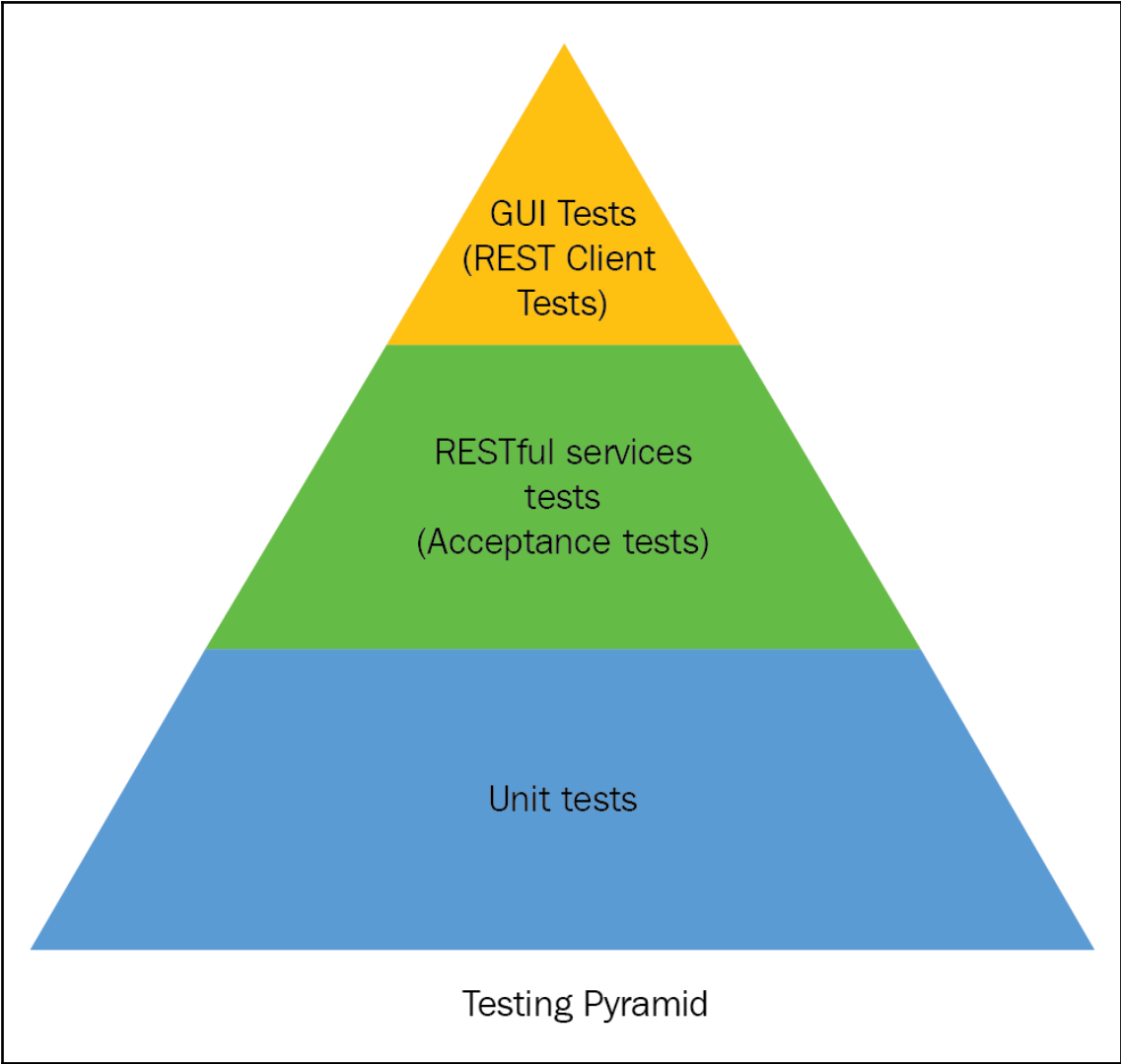


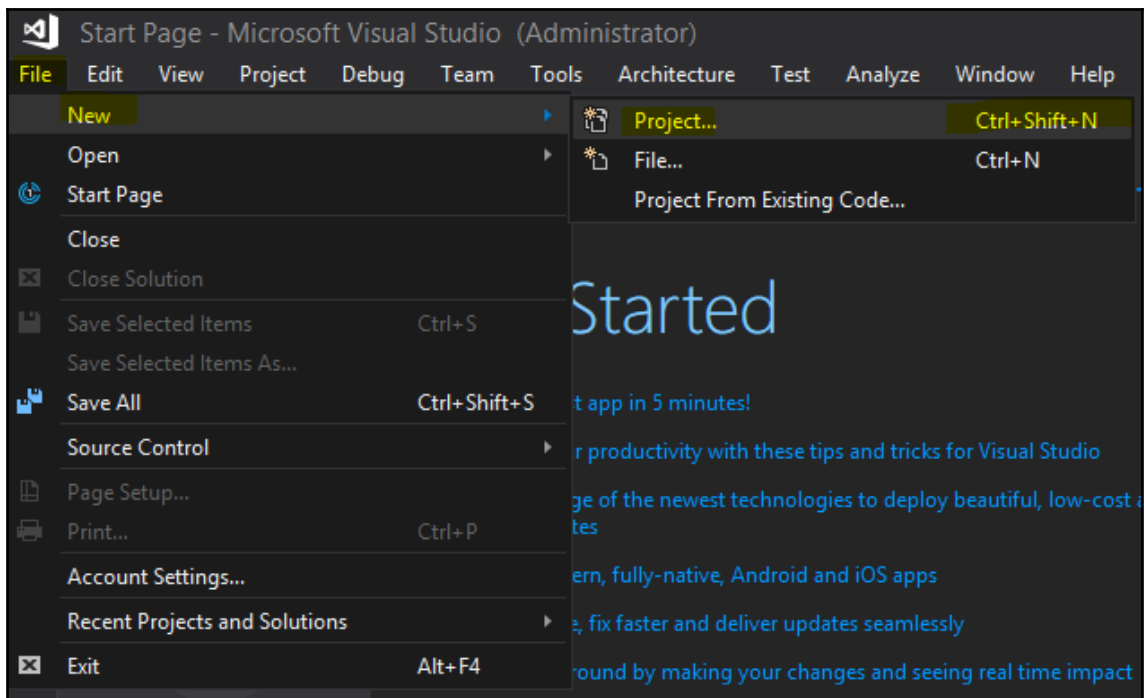


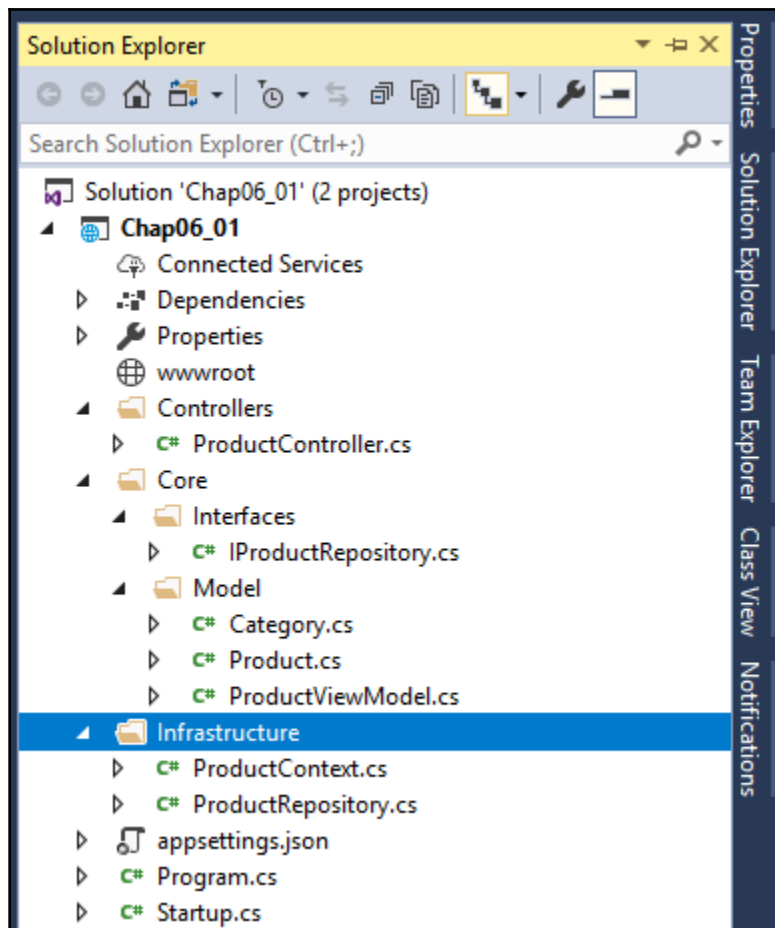
```
Request starting HTTP/1.1 GET http://localhost:60431/api/Product/productlist
  info:info: Chap05_03.Middleware.FixOneLoggerMiddleware[0]
        Log Request from middleware: http localhost:60431/api/Product/productlist
: Chap05_03.Middleware.FixOneLoggerMiddleware[0]
  Log Request from middleware: http localhost:60431/api/Product/productlist
  info:info: Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker[1]
        Executing action method Chap05_03.Controllers.ProductController.GetList (Chap05_03) with arguments ((null)) - ModelState is Valid
: Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker[1]
  Executing action method Chap05_03.Controllers.ProductController.GetList (Chap05_03) with arguments ((null)) - ModelState is Valid
  info:info: Microsoft.EntityFrameworkCore.Infrastructure[10403]
        Entity Framework Core 2.0.2-rtm-10011 initialized 'ProductContext' using provider 'Microsoft.EntityFrameworkCore.SqlServer' with options: None
        : Microsoft.EntityFrameworkCore.Infrastructure[10403]
          Entity Framework Core 2.0.2-rtm-10011 initialized 'ProductContext' using provider 'Microsoft.EntityFrameworkCore.SqlServer' with options
: None
  info:info: Microsoft.EntityFrameworkCore.Database.Command[20101]
        Executed DbCommand (5ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
        SELECT [c].[Id], [c].[CategoryId], [c].[Description], [c].[Image], [c].[Name], [c].[Price], [c].[Category].[Id], [c].[Category].[Description], [c].[Category].
        Name
        FROM [Products] AS [c]
        INNER JOIN [Categories] AS [c].[Category] ON [c].[CategoryId] = [c].[Category].[Id]
: Microsoft.EntityFrameworkCore.Database.Command[20101]
  Executed DbCommand (5ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
        SELECT [c].[Id], [c].[CategoryId], [c].[Description], [c].[Image], [c].[Name], [c].[Price], [c].[Category].[Id], [c].[Category].[Description]
], [c].[Category].[Name]
        FROM [Products] AS [c]
        INNER JOIN [Categories] AS [c].[Category] ON [c].[CategoryId] = [c].[Category].[Id]
  info:info: Microsoft.AspNetCore.Mvc.Internal.ObjectResultExecutor[1]
        Executing ObjectResult, writing value Microsoft.AspNetCore.Mvc.ControllerContext.
: Microsoft.AspNetCore.Mvc.Internal.ObjectResultExecutor[1]
```

Chapter 6: Testing RESTful Web Services









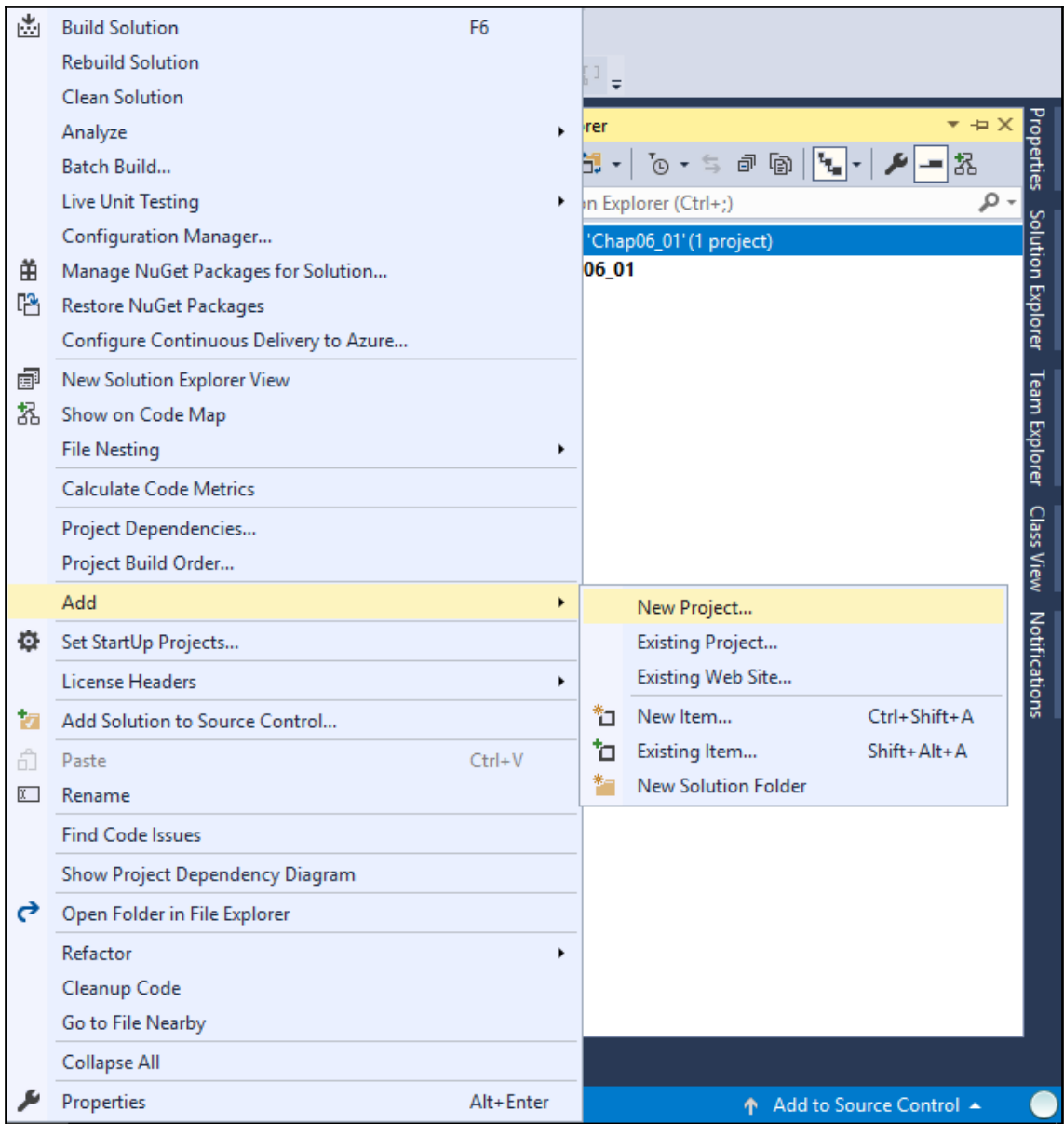
localhost:60431/swagger/

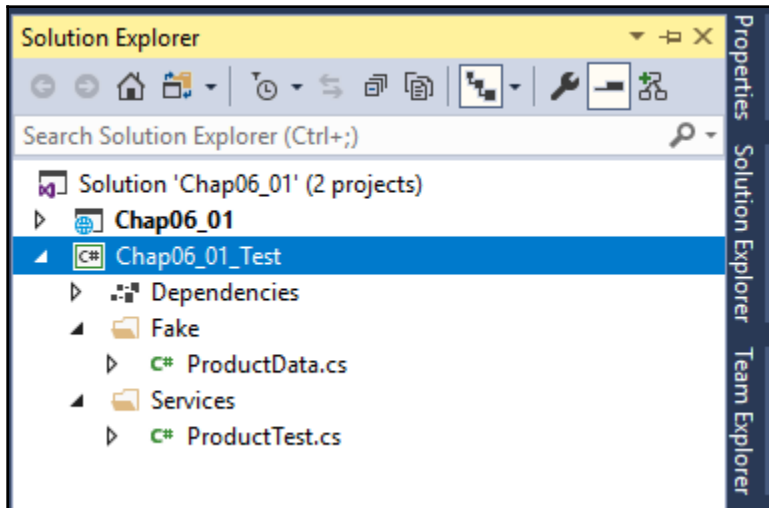
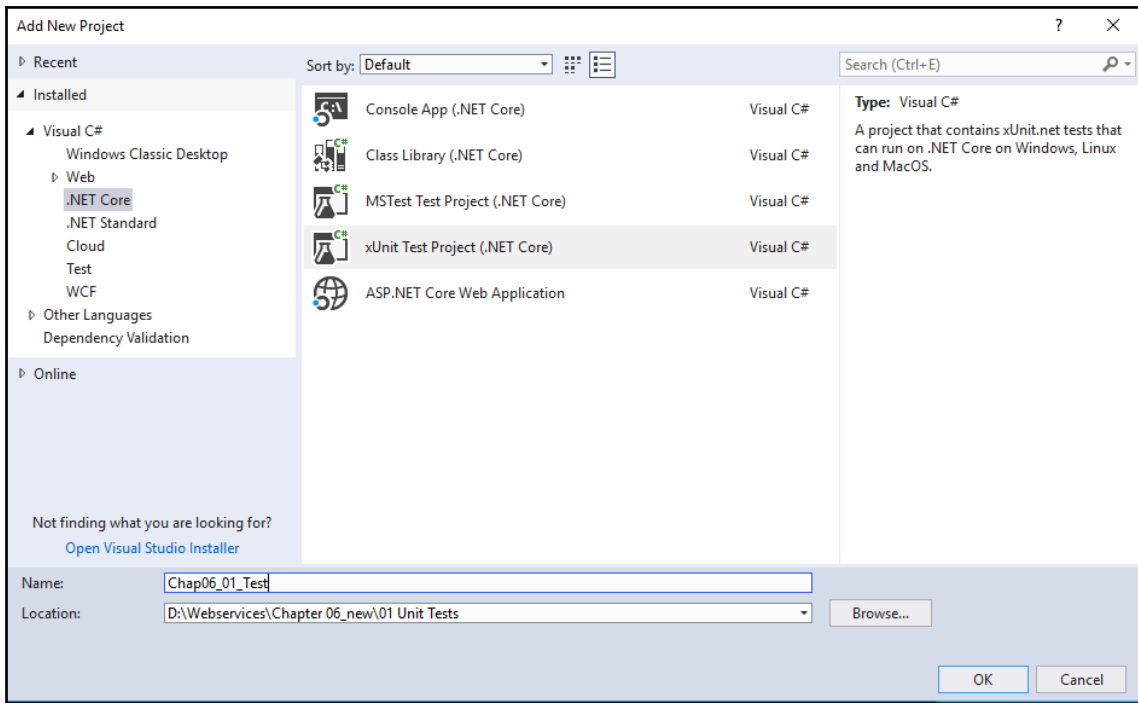
The screenshot displays the Swagger UI for 'Product API V1'. At the top, there is a green header with the Swagger logo and a dropdown menu labeled 'Select a spec' currently set to 'Product API V1'. Below the header, the main title is 'Product APIs' with a 'v1' badge. A link to the Swagger JSON file is provided: </swagger/v1/swagger.json>. The 'Product' category is expanded, showing a list of endpoints:

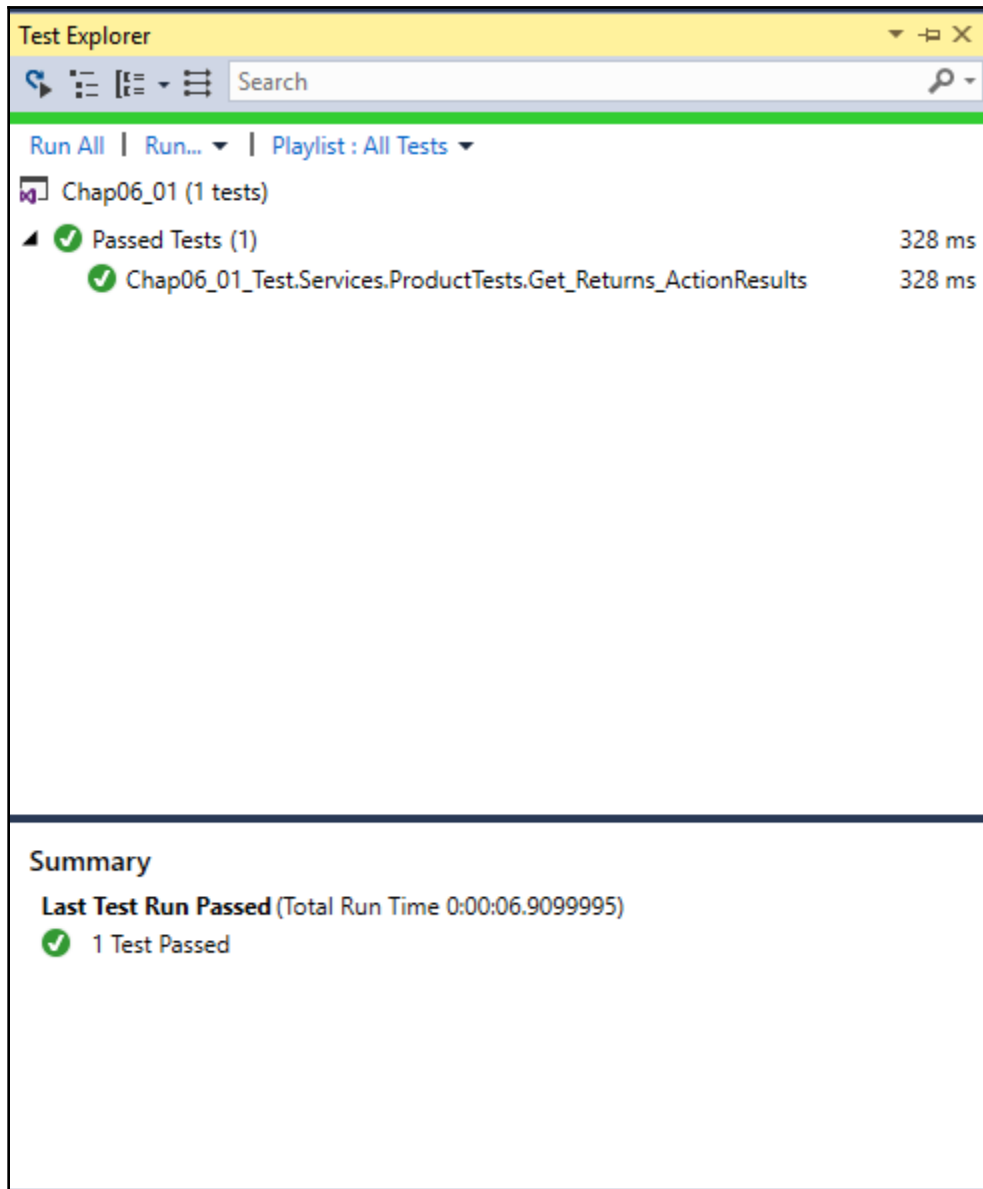
- GET** /api/Product/productlist
- GET** /api/Product/product/{productid}
- POST** /api/Product/addproduct
- PUT** /api/Product/updateproduct/{productid}
- DELETE** /api/Product/deleteproduct/{productid}

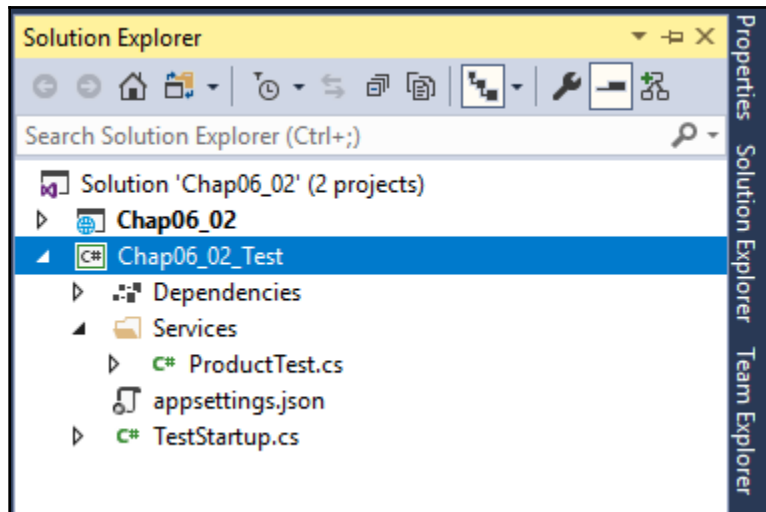
Response body

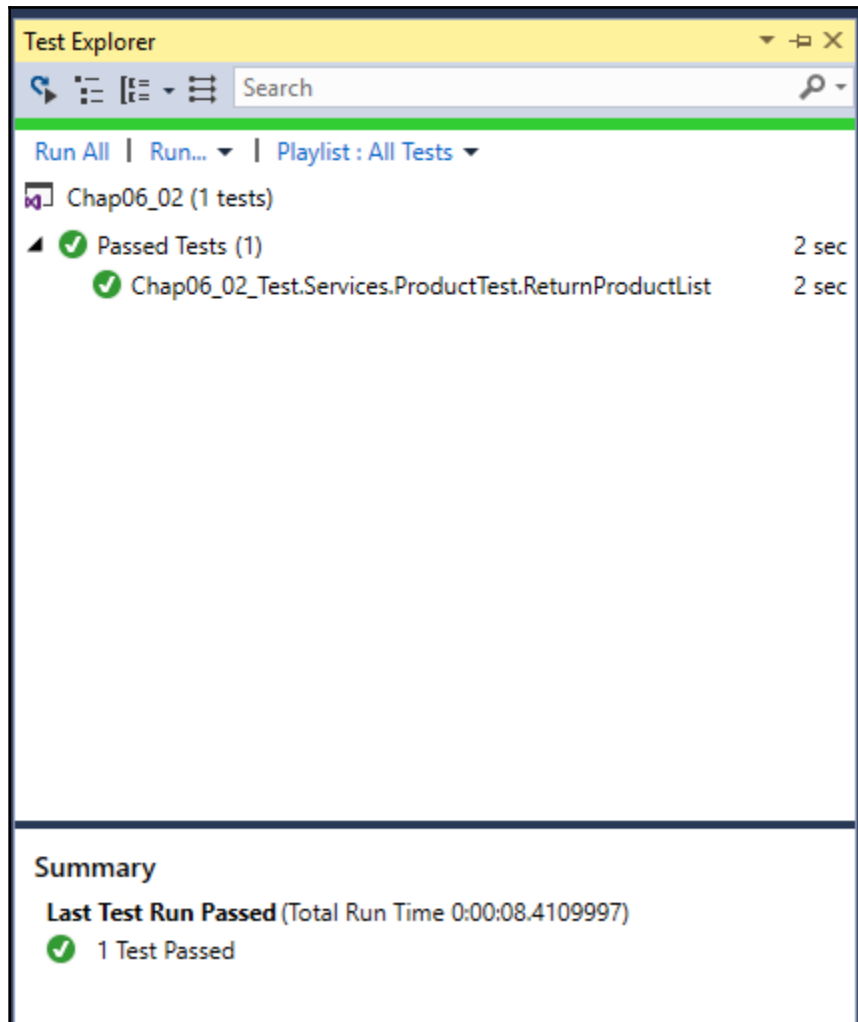
```
[
  {
    "productId": "02341321-c20b-48b1-a2be-47e67f548f0f",
    "productName": "Microservices for .NET",
    "productDescription": "Microservices for .NET Core",
    "productImage": "microservices.jpeg",
    "productPrice": 651,
    "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f",
    "categoryName": "Books",
    "categoryDescription": "Technical Books"
  },
  {
    "productId": "4d261e4a-a657-4add-a0f6-dde6e1464d55",
    "productName": "Learn C#",
    "productDescription": "Learn C# in 7 days",
    "productImage": "csharp.jpeg",
    "productPrice": 520,
    "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f",
    "categoryName": "Books",
    "categoryDescription": "Technical Books"
  }
]
```

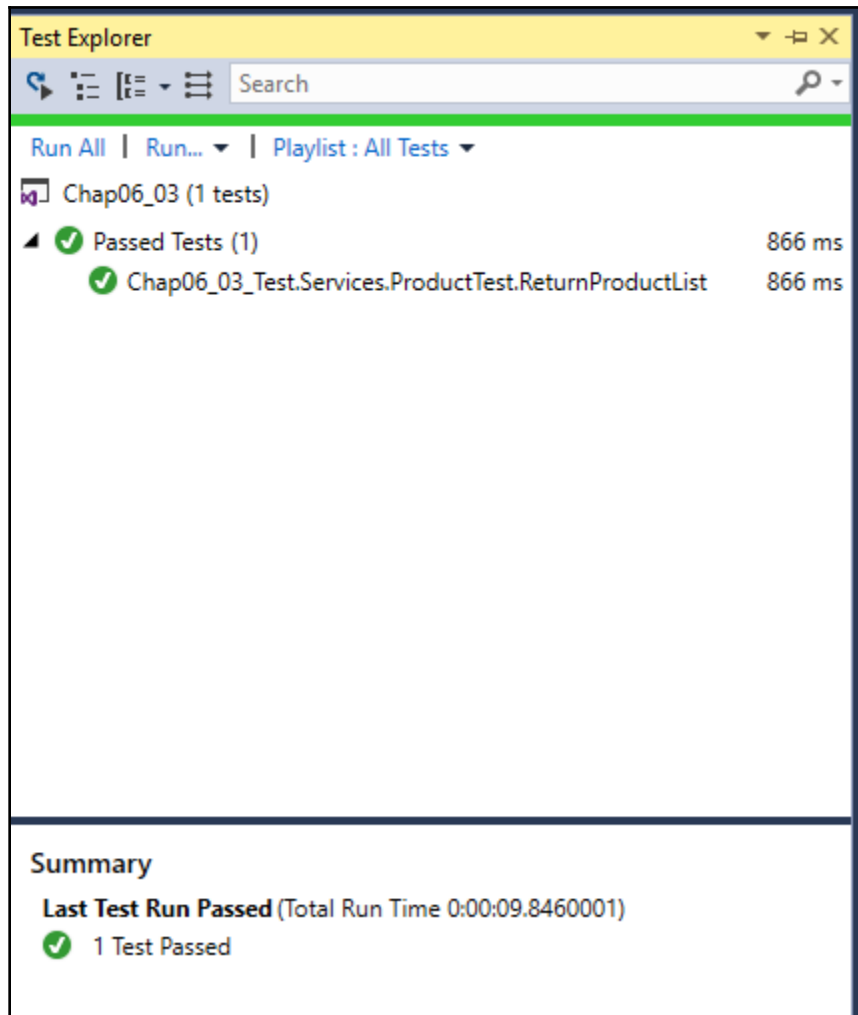









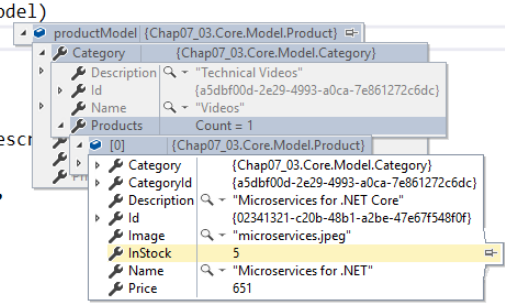
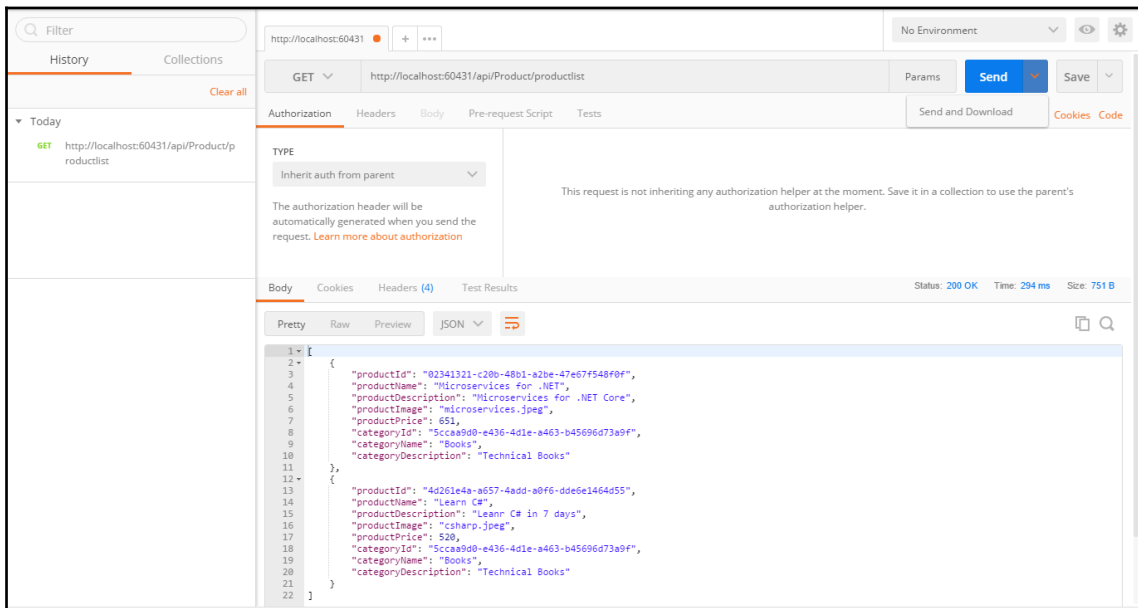




```

2 references | 0 exceptions
private ProductViewModel ToProductVm(Product productModel)
{
    ≤ 2ms elapsed
    return new ProductViewModel
    {
        CategoryId = productModel.CategoryId,
        CategoryDescription = productModel.Category.Description,
        CategoryName = productModel.Category.Name,
        ProductDescription = productModel.Description,
        ProductId = productModel.Id,
        ProductImage = productModel.Image,
        ProductName = productModel.Name,
        ProductPrice = productModel.Price
    };
}

```

Filter

History Collections

Clear all

Today

GET http://localhost:60431/api/Product/productlist

Authorization Headers Body Pre-request Script Tests

Send and Download Cookies Code

TYPE

Inherit auth from parent

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies Headers (4) Test Results

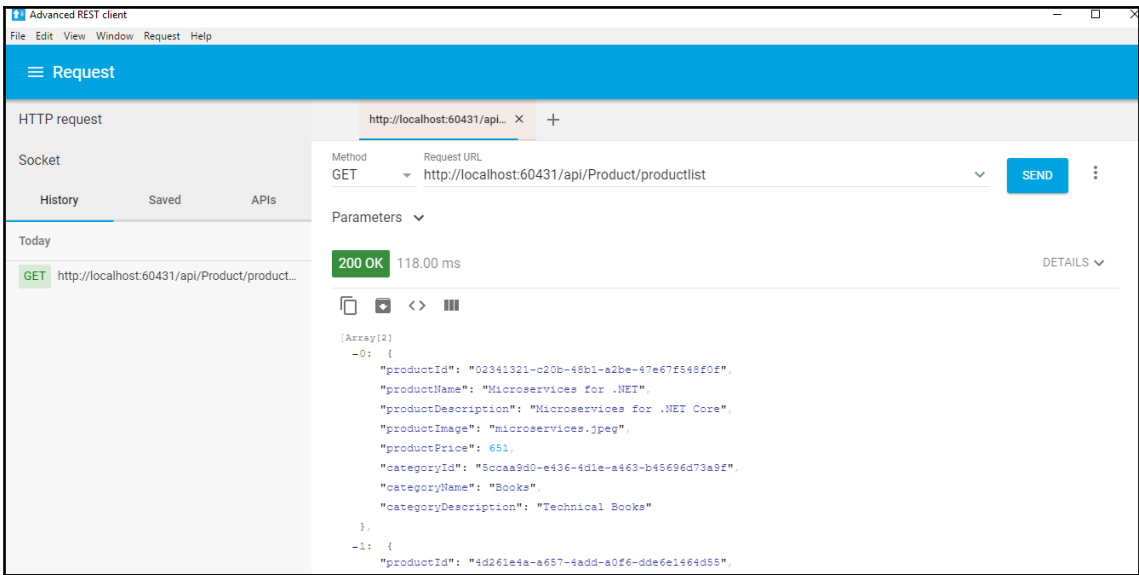
Status: 200 OK Time: 294 ms Size: 751 B

Pretty Raw Preview JSON

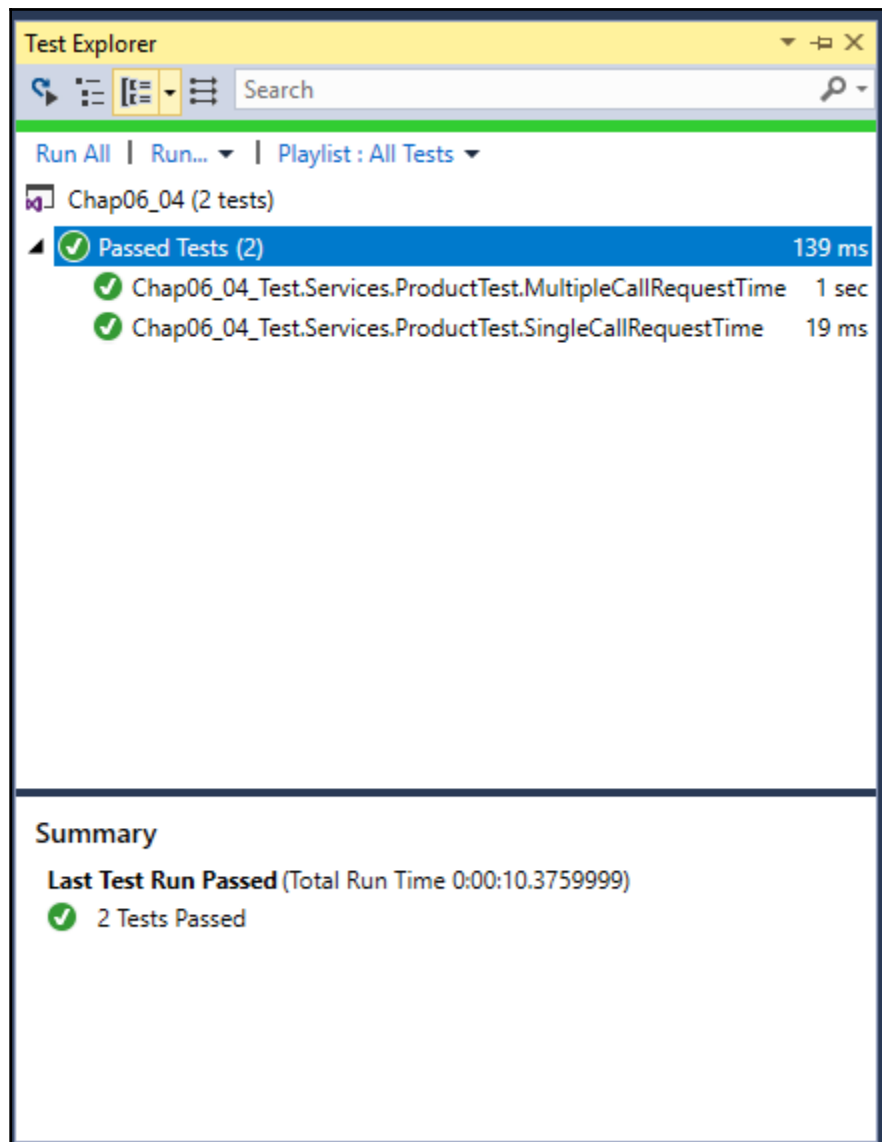
```

1 - [
2 - {
3 -   "productId": "02341321-c20b-48b1-a2be-47e67f548f0f",
4 -   "productName": "Microservices for .NET",
5 -   "productDescription": "Microservices for .NET Core",
6 -   "productImage": "microservices.jpeg",
7 -   "productPrice": 651,
8 -   "categoryId": "5caa89d8-e436-4d1e-a463-b45696d73a9f",
9 -   "categoryName": "Books",
10 -  "categoryDescription": "Technical Books"
11 - },
12 - {
13 -   "productId": "4d261e4a-a657-4add-a0f6-dde6e1464d55",
14 -   "productName": "Learn C#",
15 -   "productDescription": "Learn C# in 7 days",
16 -   "productImage": "csharp.jpeg",
17 -   "productPrice": 520,
18 -   "categoryId": "5caa89d8-e436-4d1e-a463-b45696d73a9f",
19 -   "categoryName": "Books",
20 -   "categoryDescription": "Technical Books"
21 - }
22 - ]

```



```
D:\Webservices\Chapter_06_new\04 Load Tests\Chap06_04>dotnet run
Using launch settings from D:\Webservices\Chapter_06_new\04 Load Tests\Chap06_04\Properties\launchSettings.json...
Hosting environment: Development
Content root path: D:\Webservices\Chapter_06_new\04 Load Tests\Chap06_04
Now listening on: http://localhost:60431
Application started. Press Ctrl+C to shut down.
```

Test Name: Chap06_04_Test.Services.ProductTest.MultipleCallRequestTime

Test Outcome:  Passed

Standard Output

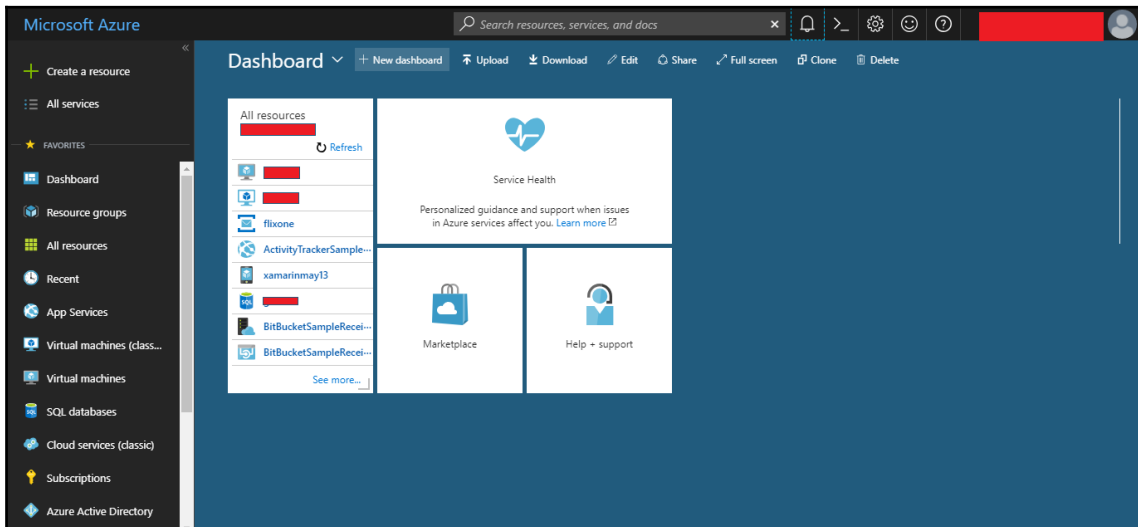
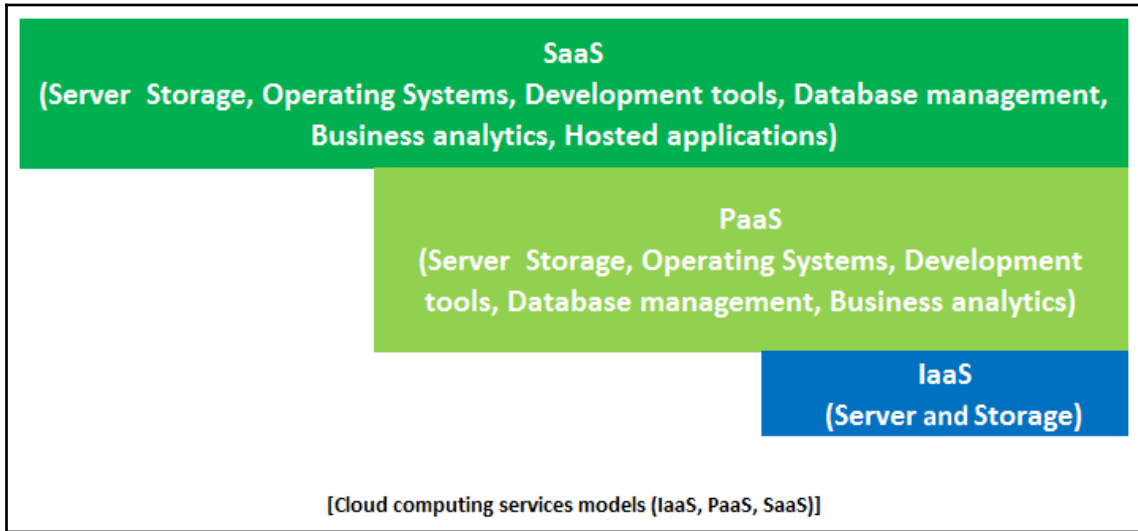
Actual time: 1284.7536 millisecond. Expected time: 115000 millisecond.

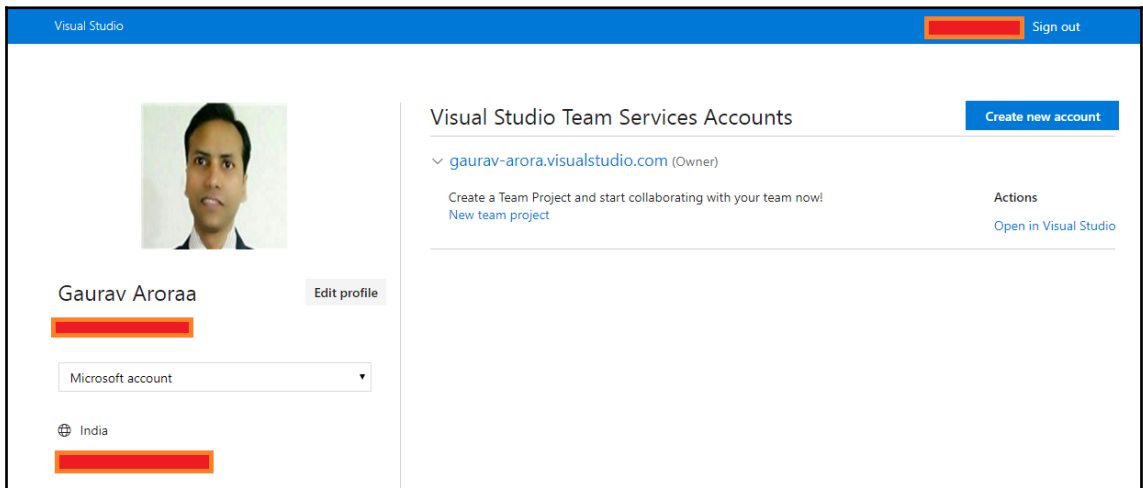
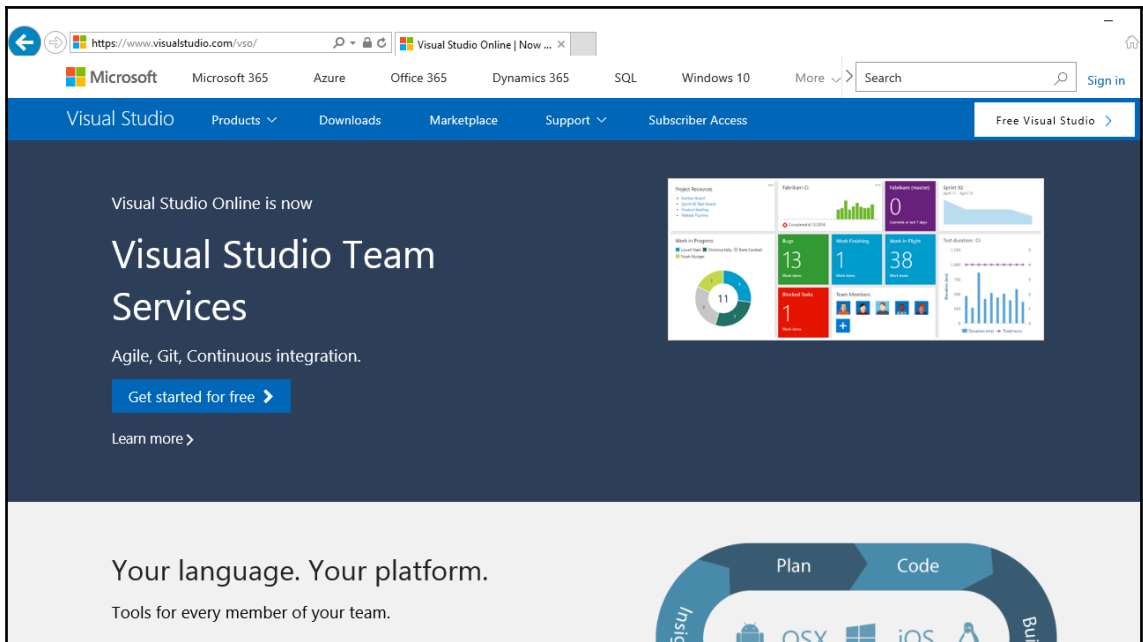
```
Test run for D:\Webservices\Chapter_06_new\04_Load_Tests\Chap06_04_Test\bin\Debug\netcoreapp2
.0\Chap06_04_Test.dll(.NETCoreApp,Version=v2.0)
Microsoft (R) Test Execution Command Line Tool Version 15.6.0
Copyright (c) Microsoft Corporation. All rights reserved.

Starting test execution, please wait...
[xUnit.net 00:00:01.0561900] Discovering: Chap06_04_Test
[xUnit.net 00:00:01.2402047] Discovered: Chap06_04_Test
[xUnit.net 00:00:01.2468461] Starting: Chap06_04_Test
[xUnit.net 00:00:02.9754566] Finished: Chap06_04_Test

Total tests: 2. Passed: 2. Failed: 0. Skipped: 0.
Test Run Successful.
Test execution time: 5.6981 Seconds
D:\Webservices\Chapter_06_new\04_Load_Tests\Chap06_04_Test>_
```

Chapter 7: Continuous Integration and Continuous Deployment





Create new project

Projects contain your source code, work items, automated builds and more.

Project name *

 ✓

Description

Version control

 ▾

Work item process

 ▾

The screenshot shows the VSTS project dashboard for 'FlixOneProductServices'. The top navigation bar includes 'Dashboards', 'Code', 'Work', 'Build and Release', 'Test', and 'Wiki'. The main content area features a 'Use continuous integration' card with a 'Set up Build' button and a 'No README.md found' message. The right sidebar contains sections for 'Members (1)', 'Activity' (showing 2 changesets), 'Build & Release' (no builds yet), and 'Work' (no work items yet).

FlixOneProductServices Team Overview ☆ Edit Refresh

Welcome

Get started using Visual Studio Team Services to make the most of your team dashboard.

- Manage Work**
Add work to your board
- Collaborate on code**
Add code to your repository
- Continuously integrate**
Automate your builds
- Visualize progress**
Learn how to add charts

Work assigned to Gaurav Aroraa (0)

All done with the work assigned to you? Go to your [team backlog](#) to pick up new work.

0

Team Members

It's lonely in here...

[Invite a friend](#)

Sprint Burndown

Set iteration dates to use the sprint burndown widget

[Set iteration dates](#)

New Work Item

Enter title

Bug

[Create](#)

Work in Progress

0

Work items

Visual Studio

[Open in Visual Studio](#)
Requires Visual Studio 2013+

[Get Visual Studio](#)
See Visual Studio downloads

\$/FlixOneProductServices Files Changesets Shelvesets

\$/FlixOneProductServices /

BuildProcessTemplates

Name	Last change	Changesets	
BuildProcessTemplates	3 hours ago	3	Checking in new Team Foundation Build Automation files. Gaurav Aroraa

Work Items Backlogs Queries

Product backlog

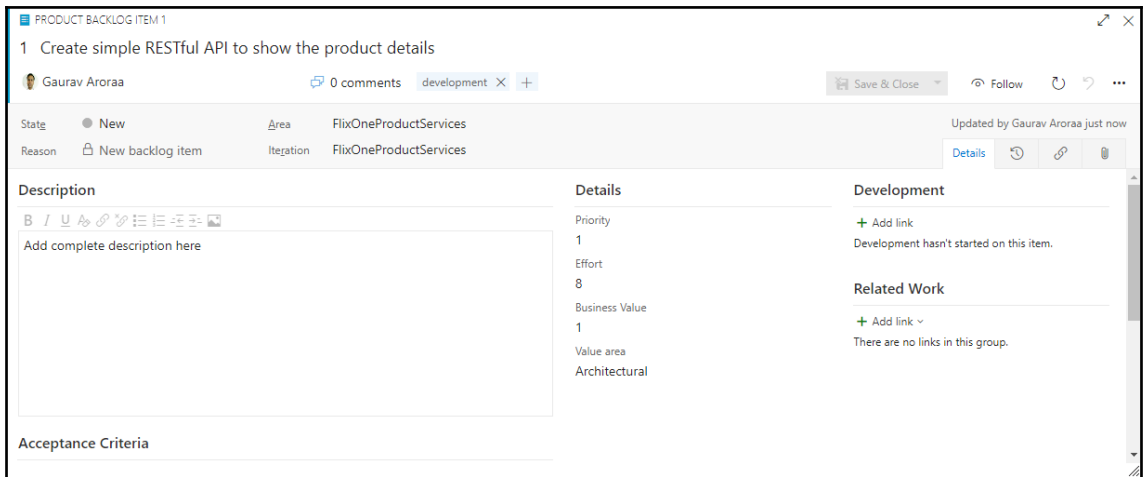
Backlog Board Forecast Off Parents Hide In progress items Show Mapping Off

New Create query Column options

Type: Product Backlog Item

Title: Create simple RESTful API to show the product details Add

Order	Work Item Type	Title	State	Effort	Value Area	Iteration Path	Tags
-------	----------------	-------	-------	--------	------------	----------------	------




✕

Edit iteration


Sprint 1

Iteration name

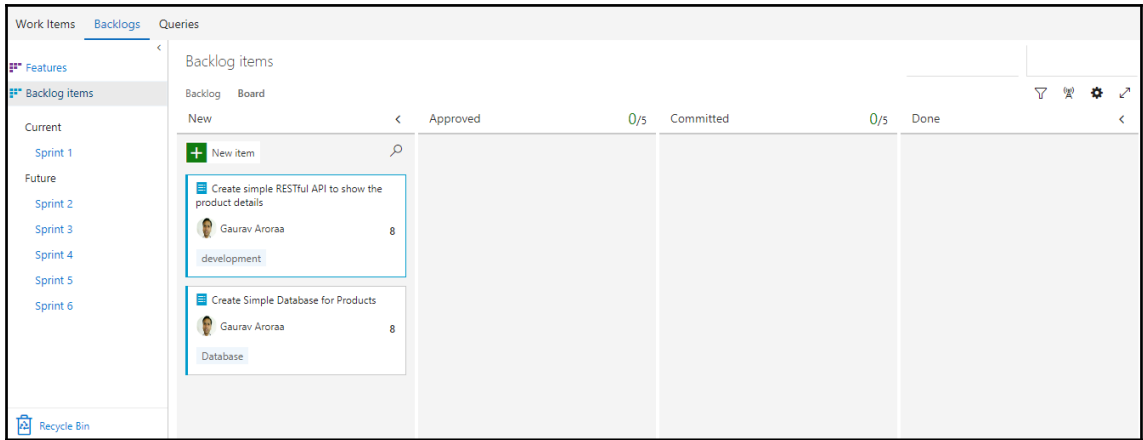
Start date

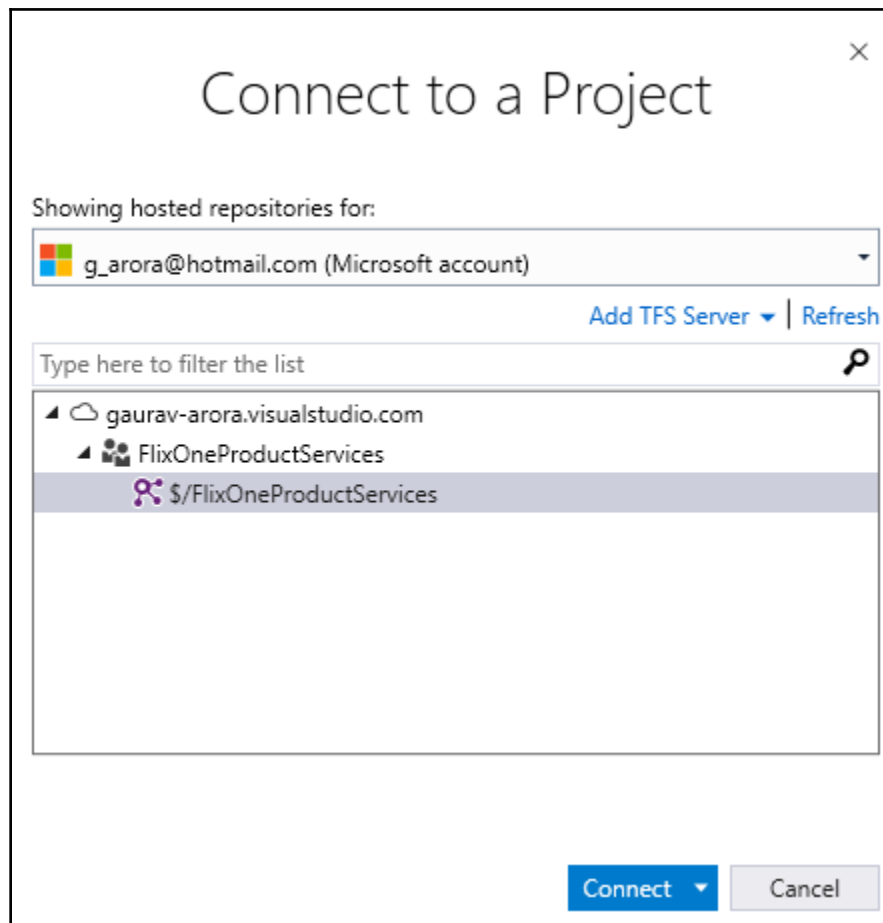
 

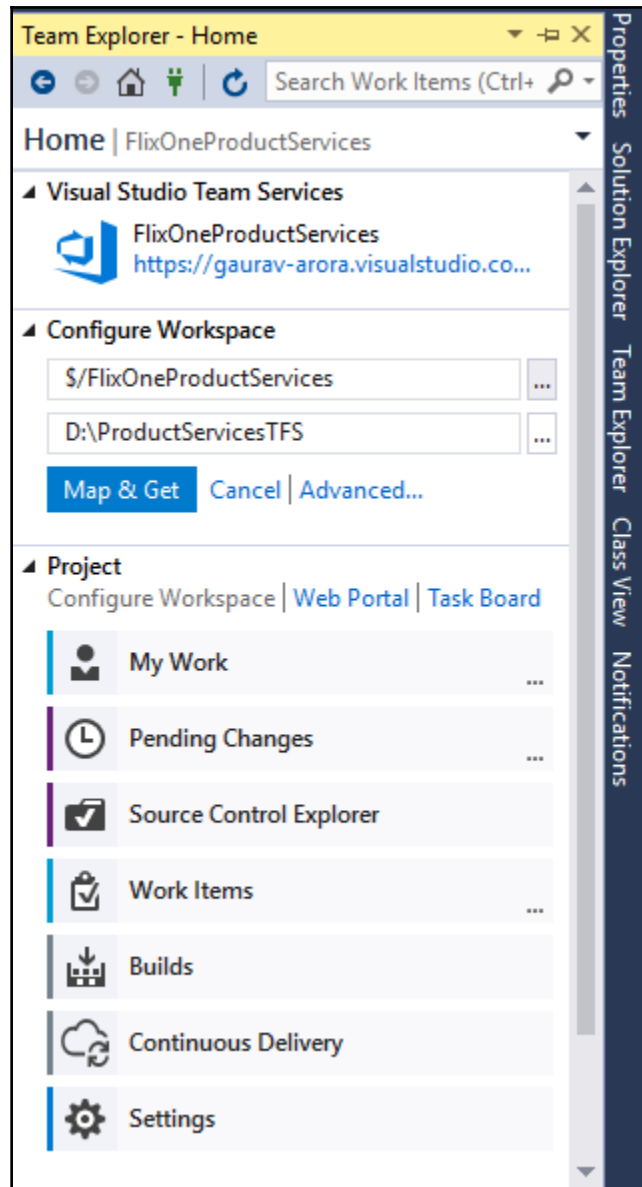
End date

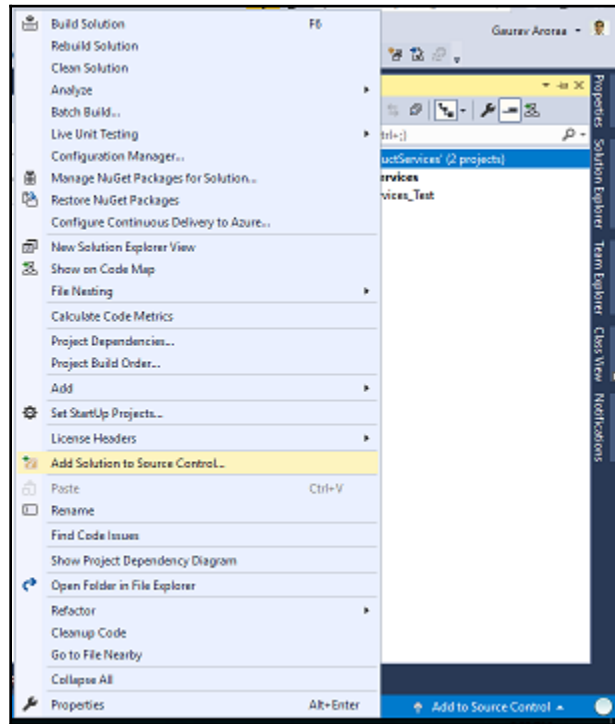
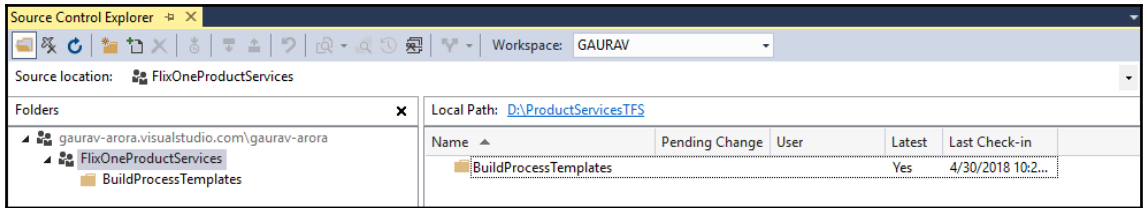
 

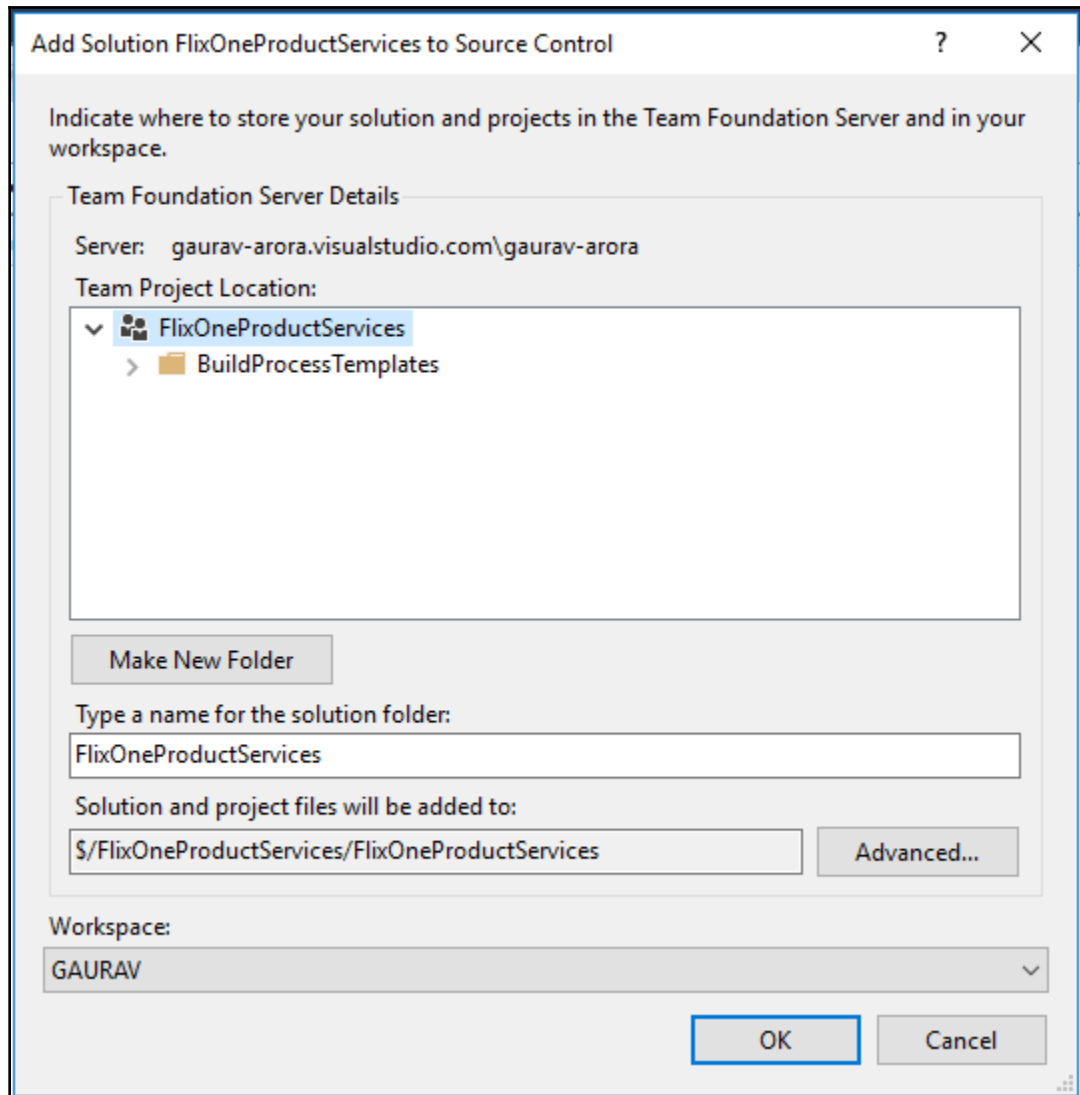
Location

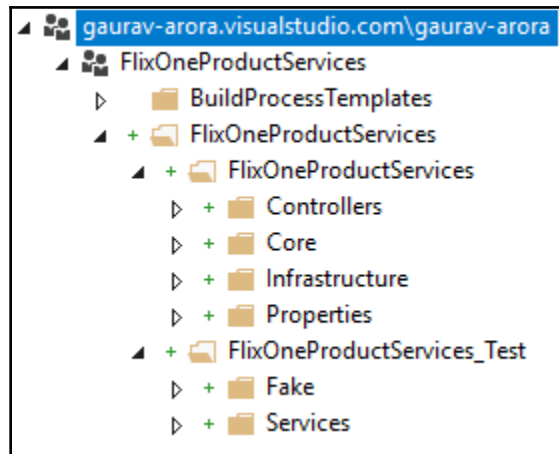


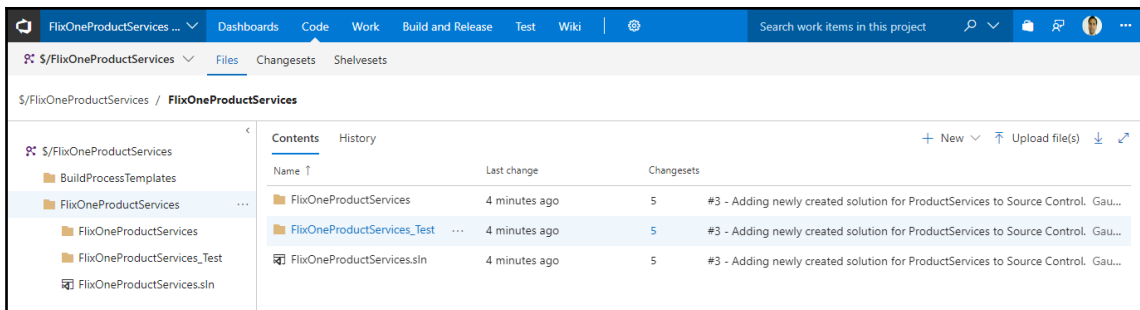
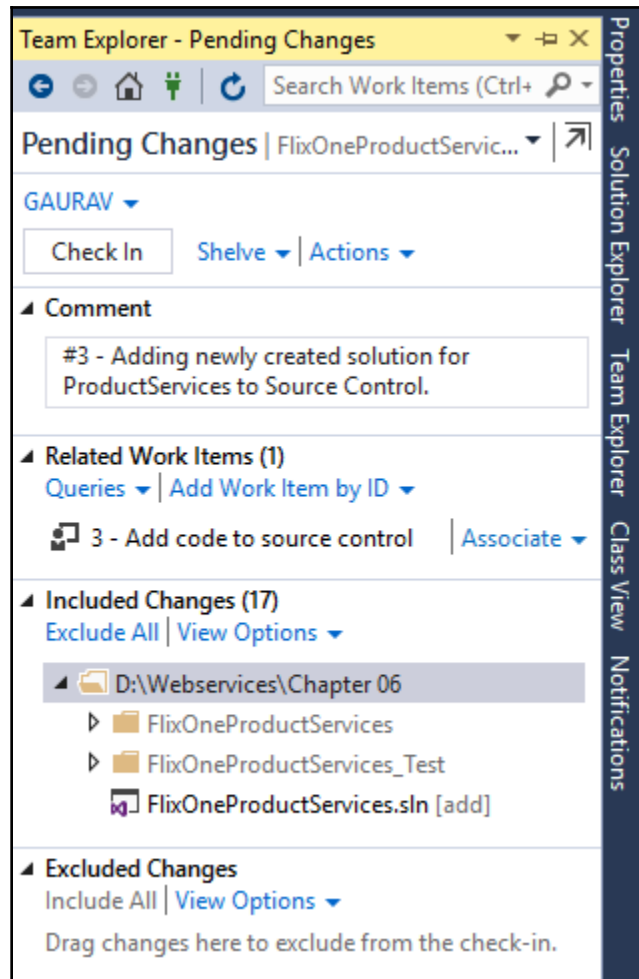


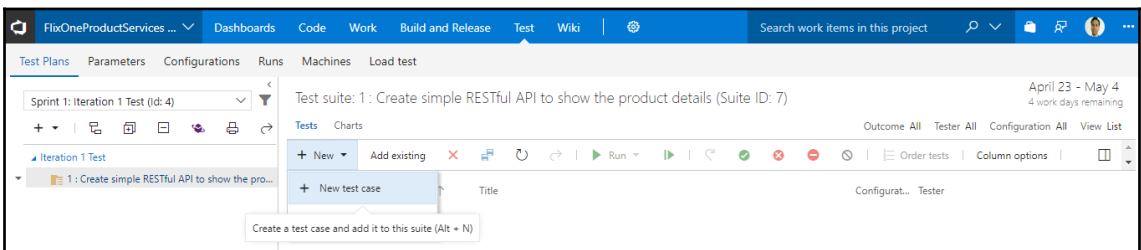
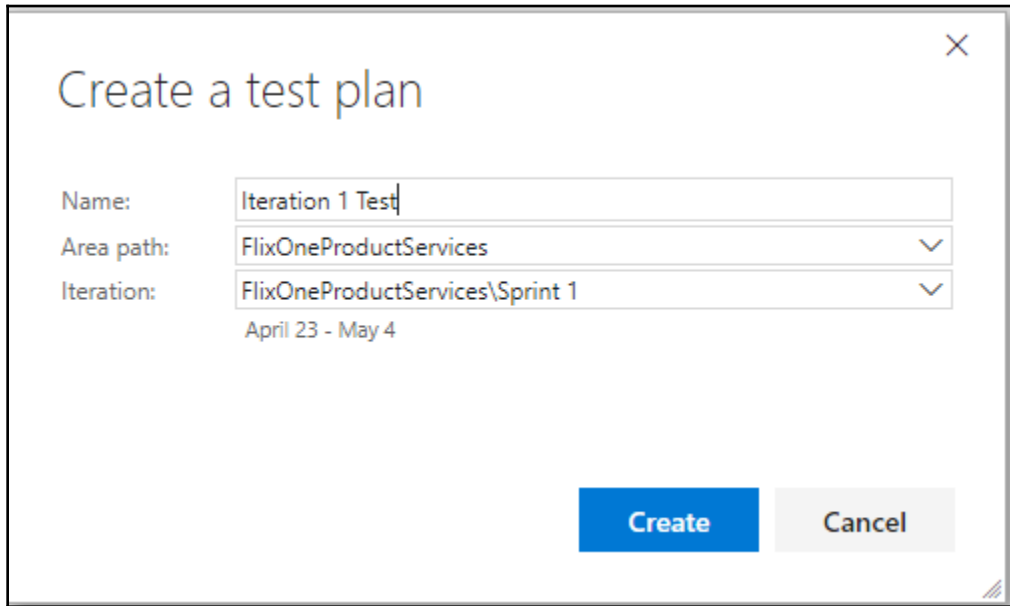
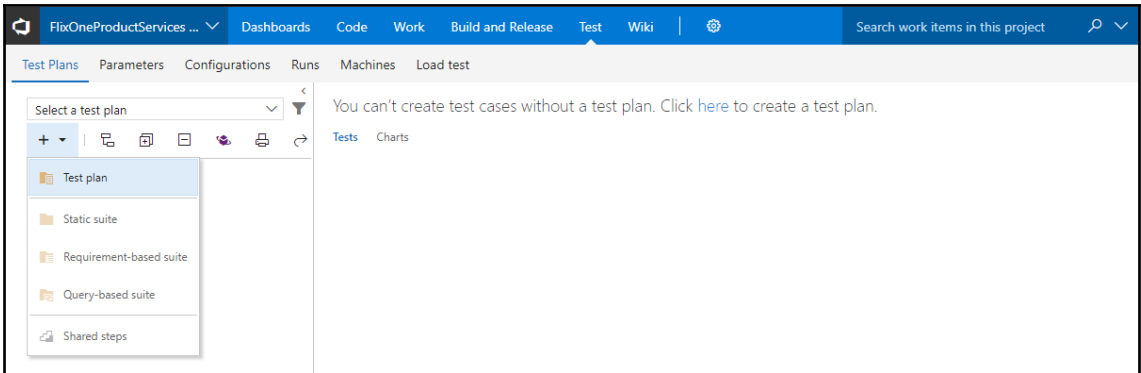


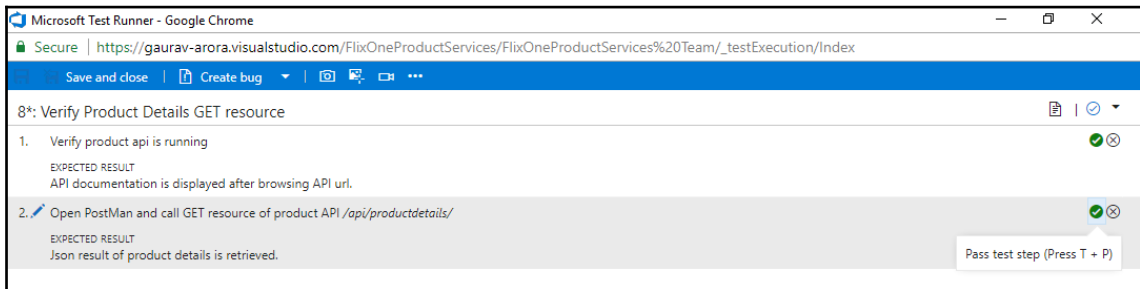
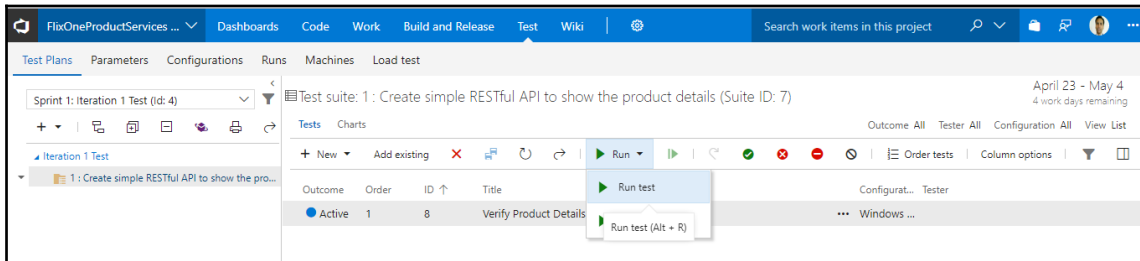
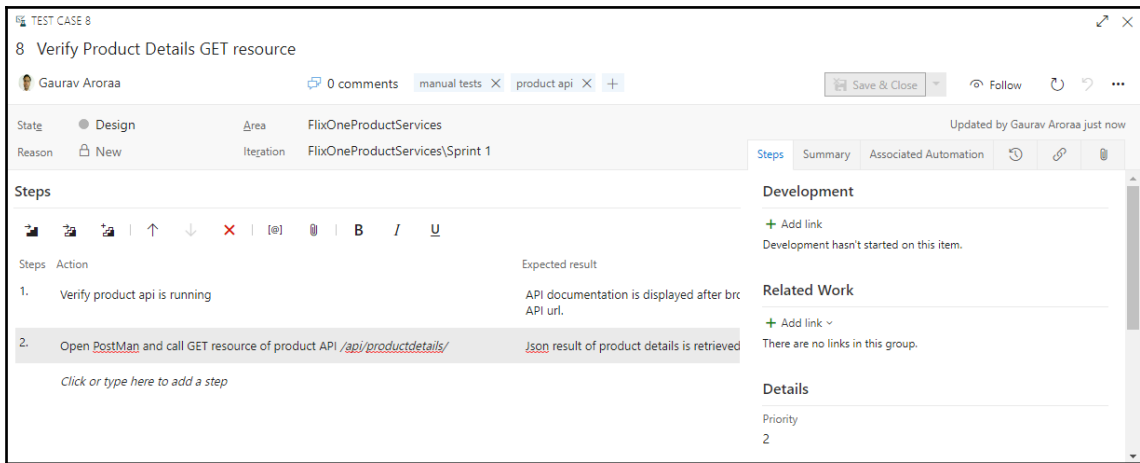


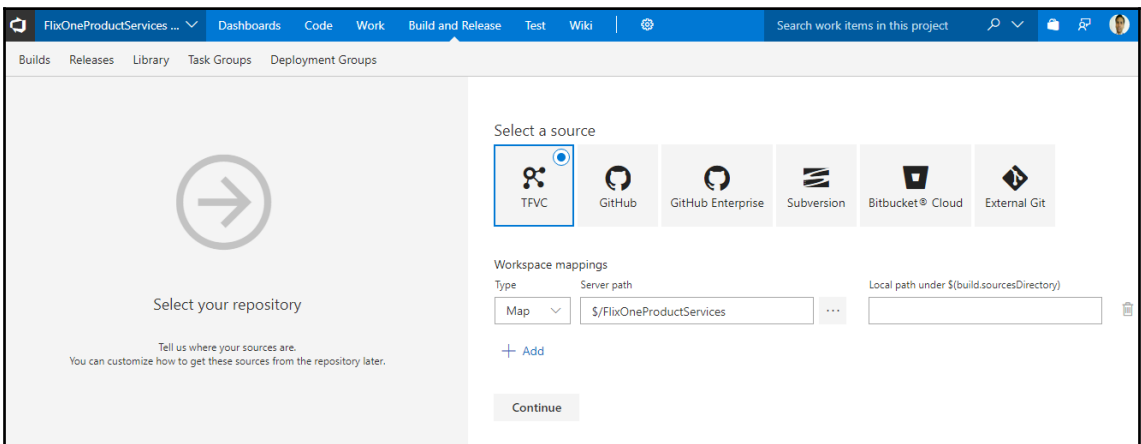
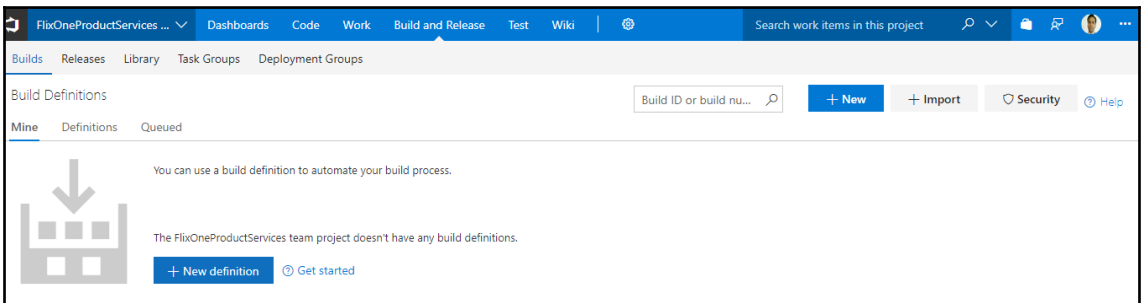
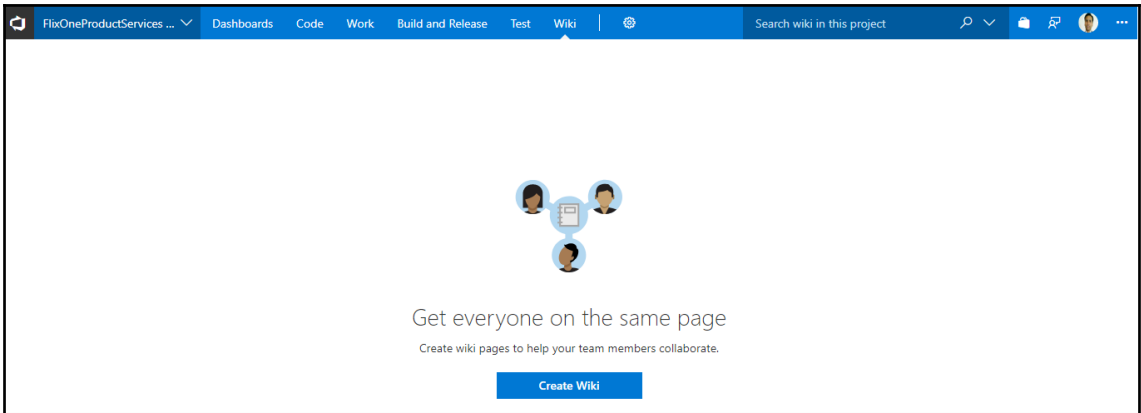


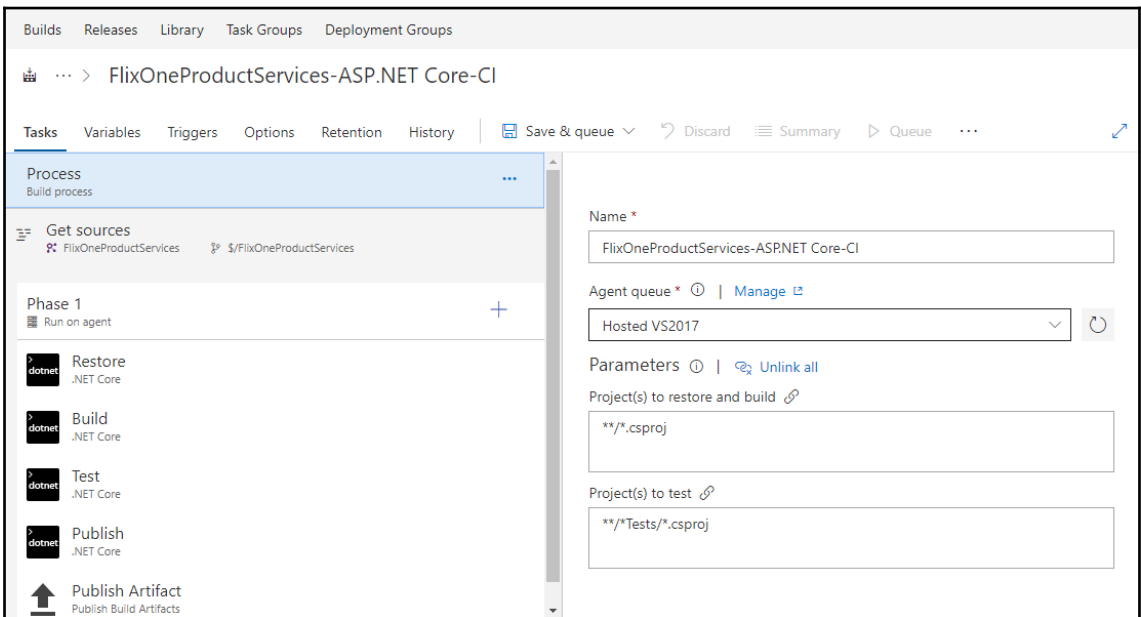
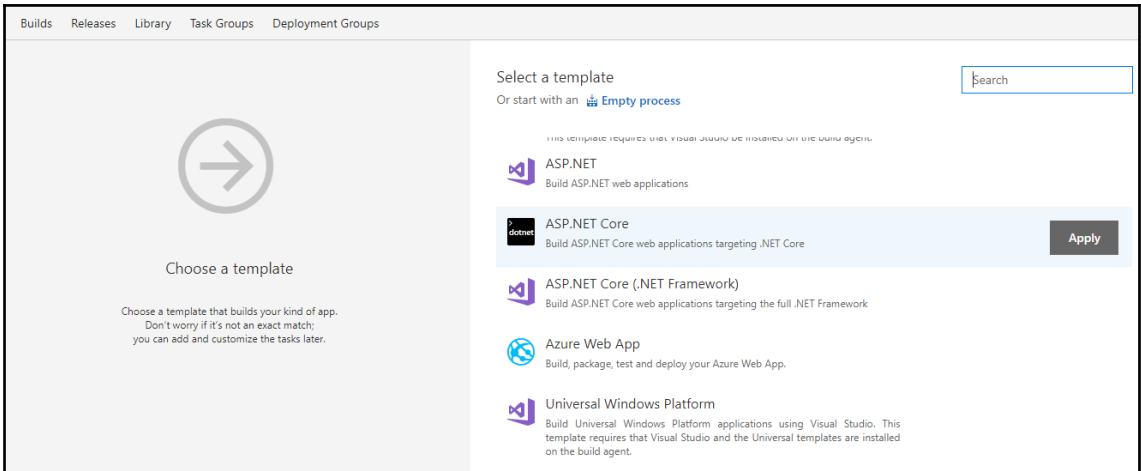












Save build definition and queue

Agent queue

Hosted VS2017

Source version ⓘ

C5

Shelveset name

Variables Demands


BuildConfiguration	release
BuildPlatform	any cpu
system.debug	false

+ Add

Save & queue Cancel

- ✓ Initialize Agent
- ✓ Initialize Job
- ✓ Get Sources
- ✓ Restore
- ✓ Build
- ✓ Test

Builds **Releases** Library Task Groups Deployment Groups



Release Management helps you automate the deployment and testing of your software in multiple environments. You can either fully automate the delivery of your software all the way to production, or set up semi-automated processes with approvals and on-demand deployments.

Start by creating a new release definition.

[+ New definition](#) [? Getting started](#) [🛡 Security](#)

Builds Releases Library Task Groups Deployment Groups

All definitions > **New Release Definition**

Pipeline Tasks Variables Retention Options History

Artifacts | + Add

+ Add artifact

Schedule not set

Environments | + Add






Environment 1
Select template


Select a Template

Or start with an [Empty process](#)


Search

Featured


-  **Azure App Service Deployment**
Deploy your application to Azure App Services. Choose from Web App On Windows, Linux, Containers, Function Apps or Web Jobs [Apply](#)
-  **Deploy Node.js App to Azure App Service**
Deploy your Node.js application to Azure Web App
-  **Deploy PHP App to Azure App Service**
Deploy your PHP Application to Azure Web App
-  **IIS Website and SQL Database Deployment**
Deployment Group: Deploy ASP.NET, ASP-Net core web applications to IIS Website and SQL Database
-  **Azure Cloud Service Deployment**
Deploy an Azure Cloud Service




Build



Git



GitHub



✓ Team Fou...

3 more artifact types ▾

Project * ⓘ

FlixOneProductServices ▾

Source (repository) * ⓘ

\$/FlixOneProductServices ▾

Default version * ⓘ

Specify at the time of release creation ▾

Source alias ⓘ

_FlixOneProductServices

ⓘ The artifacts published by each version will be available for deployment in Release Management. The last successful version of **\$/FlixOneProductServices** published the following artifacts: **BuildProcessTemplates, FlixOneProductServices, README.md.**

Add

Builds Releases Library Task Groups Deployment Groups

All definitions > New Release Definition Save + Release View releases

Pipeline **Tasks** Variables Retention Options History

Development
Deployment process

Run on agent
Run on agent +

Deploy Azure App Service
Azure App Service Deploy

Environment name
Development

Parameters | Unlink all

Azure subscription * | Manage

Scope: [Redacted]

App type
Web App

App service name *
DeployAzureAppService

New Release Definition / Release-1

Summary Environments Artifacts Variables General Commits Work items Tests Logs History

Deploy Save Abandon Send Email

Details

Release deployment

Manually created by Gaurav Aroraa 2 minutes ago

\$/FlixOneProductServices / Changeset 5 (Team Foundation Version Control)

Work items

No associated work items found.

Tags

azure app x dev x Add...

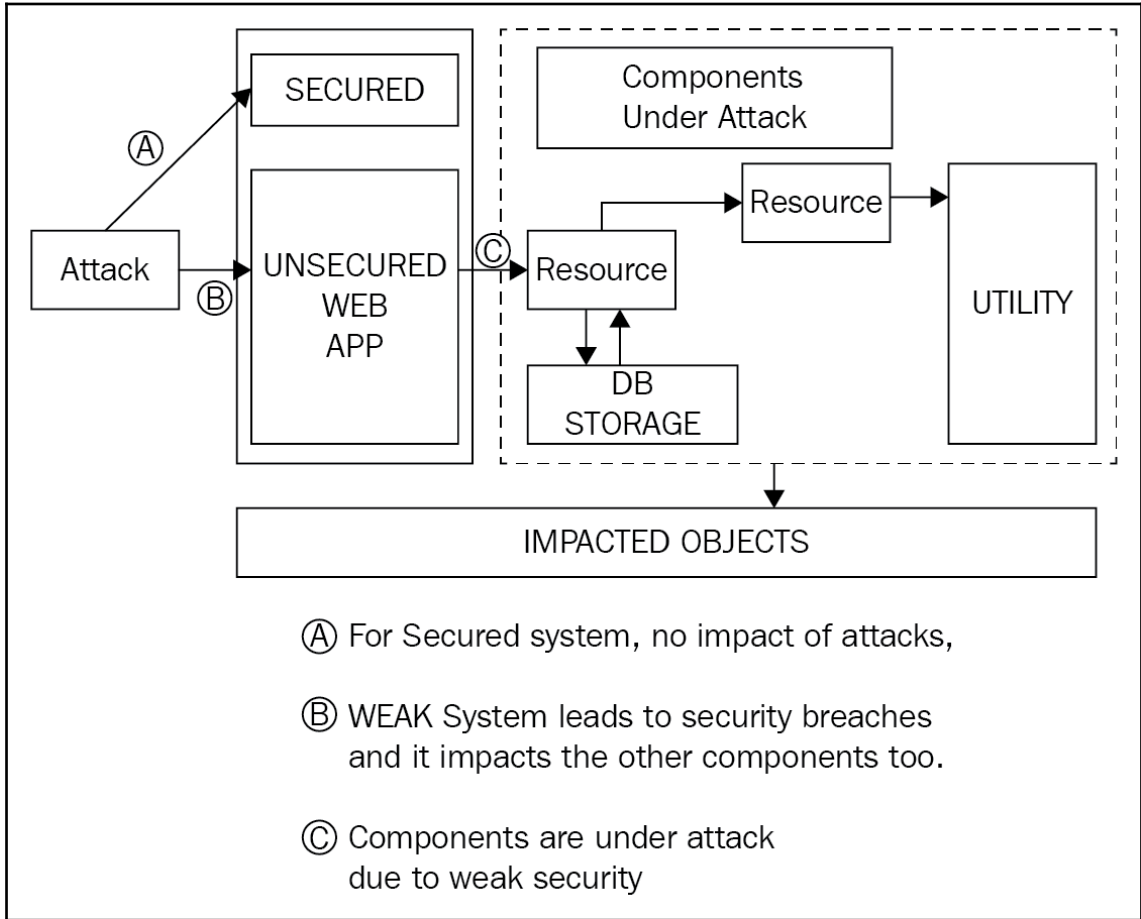
Environments

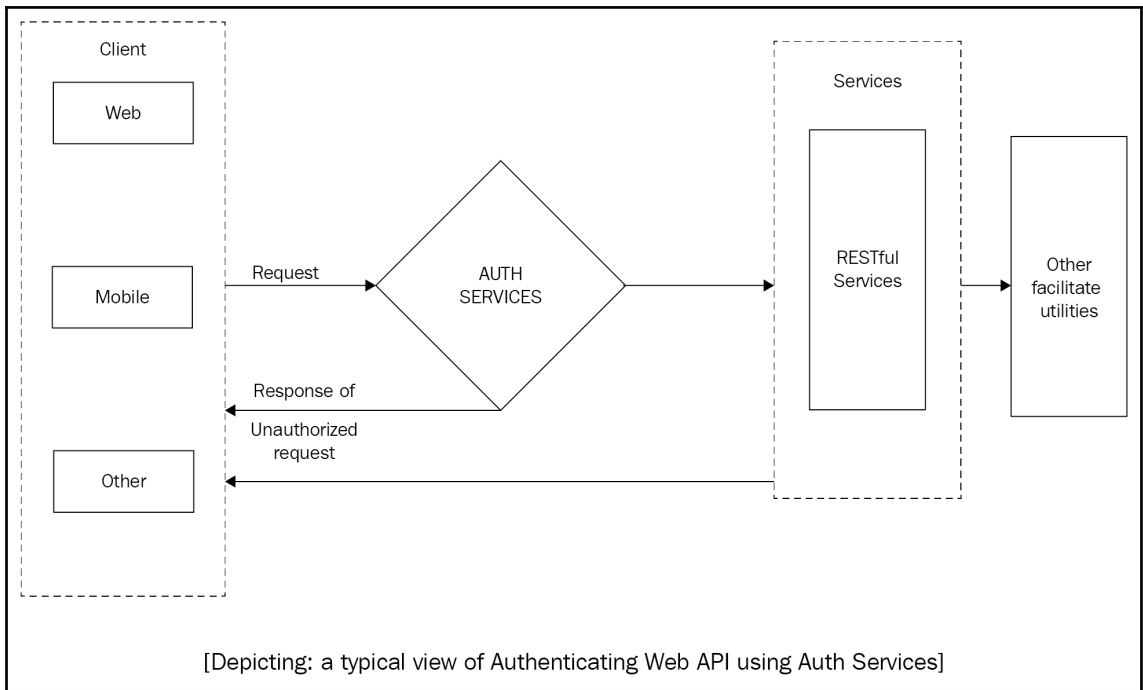
Environm...	Actions	Deployment status	Triggered	Completed	Tests
Develop...	...	NOT DEPLOYED			No tests

Issues

No issues reported in this release.

Chapter 8: Securing RESTful Web Services





Create

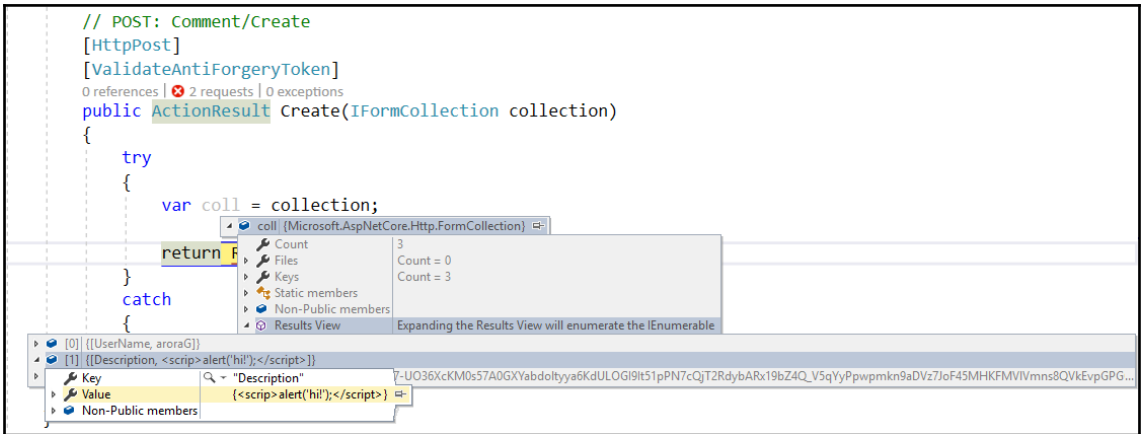
Comment

UserName

Description

Create

```
// POST: Comment/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create(IFormCollection collection)
{
    try
    {
        var coll = collection;
        return F
    }
    catch
    {
    }
}
```



Chap08_01 Home Comment About Cor localhost:36944 says

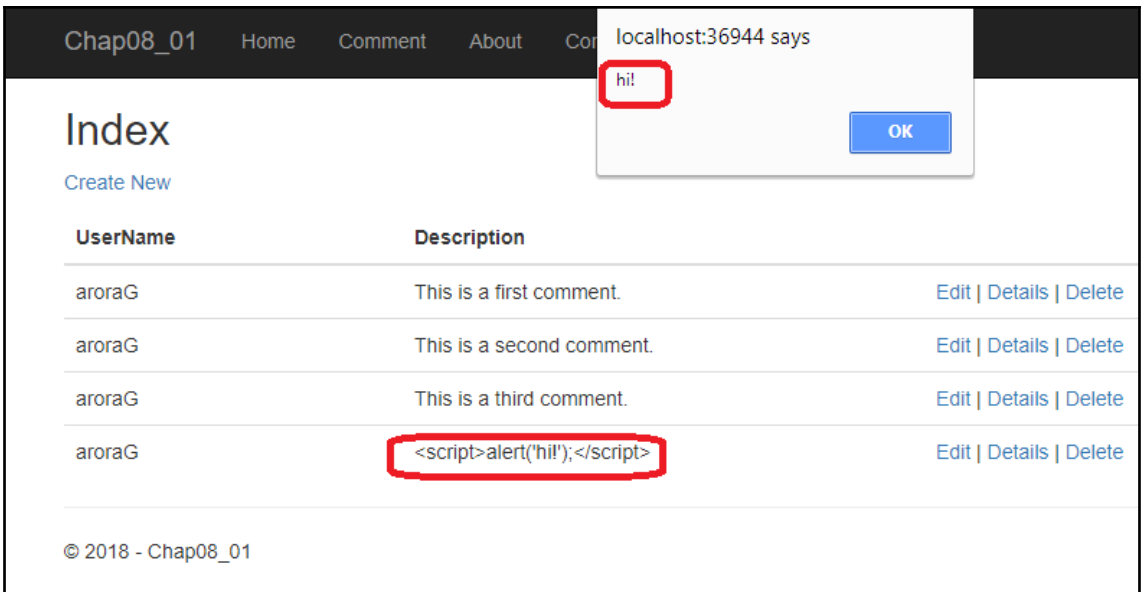
hi!

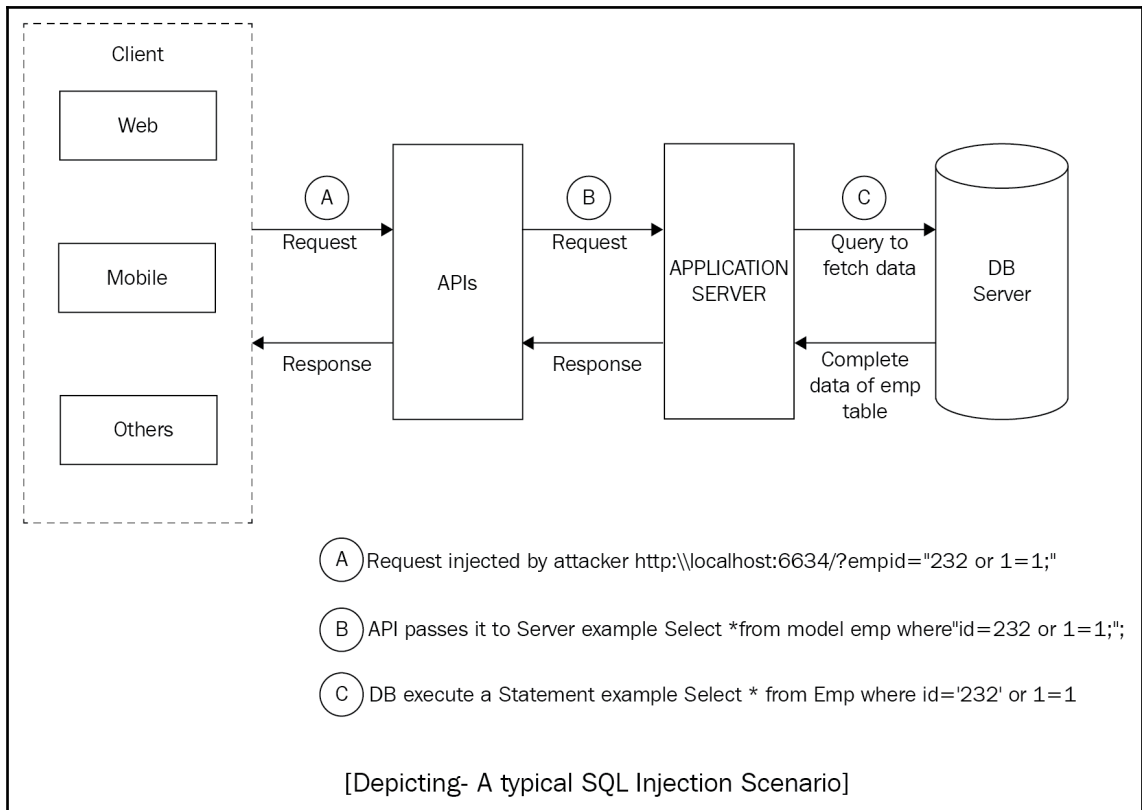
Index

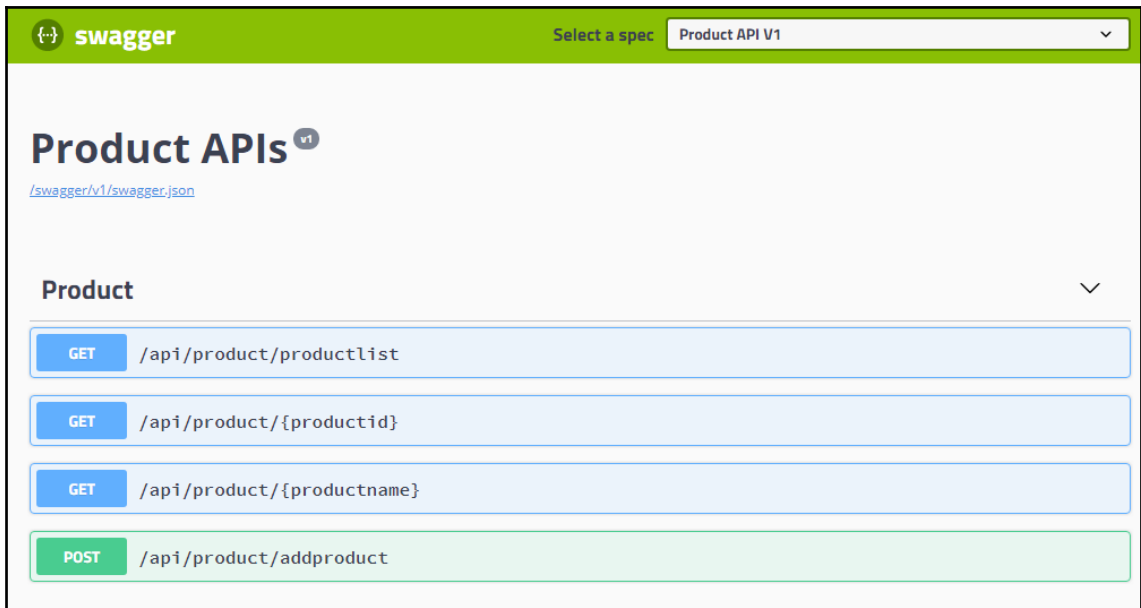
Create New

UserName	Description	
aroraG	This is a first comment.	Edit Details Delete
aroraG	This is a second comment.	Edit Details Delete
aroraG	This is a third comment.	Edit Details Delete
aroraG	<script>alert('hi!');</script>	Edit Details Delete

© 2018 - Chap08_01

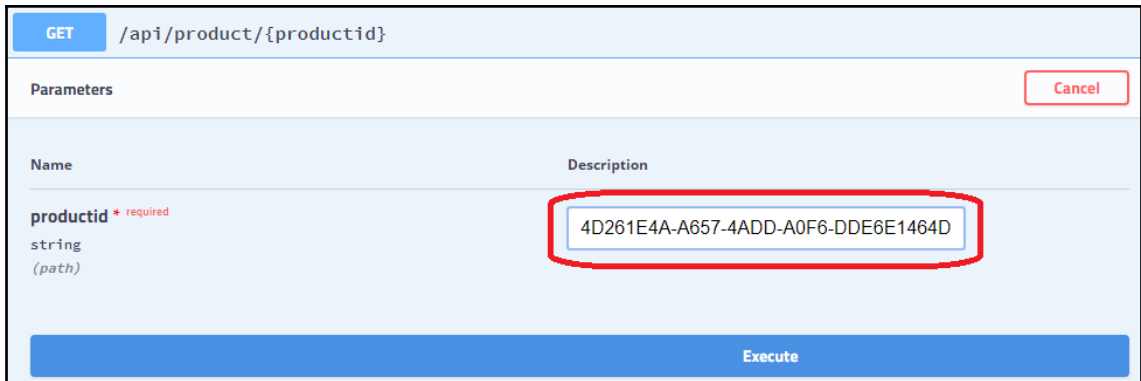






The image shows the Swagger UI interface for 'Product API V1'. The header includes the Swagger logo and a dropdown menu for 'Select a spec' with 'Product API V1' selected. The main heading is 'Product APIs' with a version indicator 'v1' and a link to the Swagger JSON file. Below this, a 'Product' section is expanded to show a list of endpoints:

- GET /api/product/productlist
- GET /api/product/{productid}
- GET /api/product/{productname}
- POST /api/product/addproduct



The image shows the configuration for the GET endpoint /api/product/{productid}. The 'Parameters' section is active, and a 'Cancel' button is visible. The parameter table is as follows:

Name	Description
productid * required string (path)	4D261E4A-A657-4ADD-A0F6-DDE6E1464D

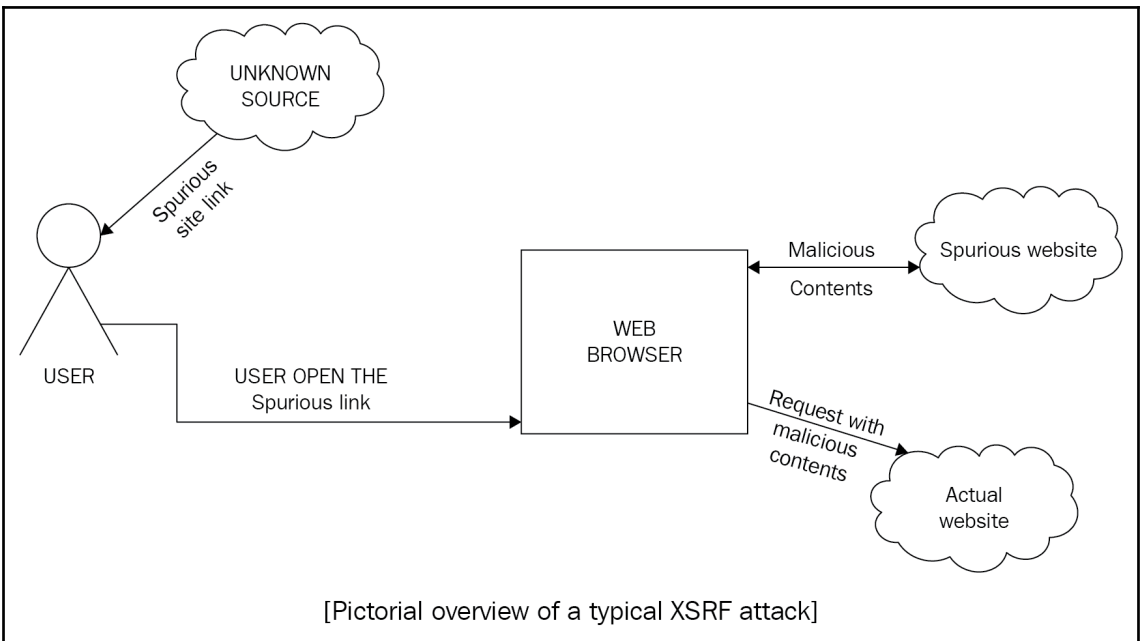
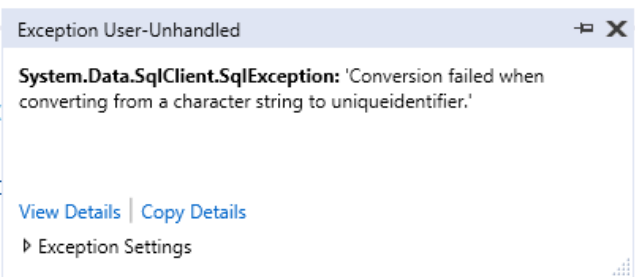
An 'Execute' button is located at the bottom of the configuration panel.

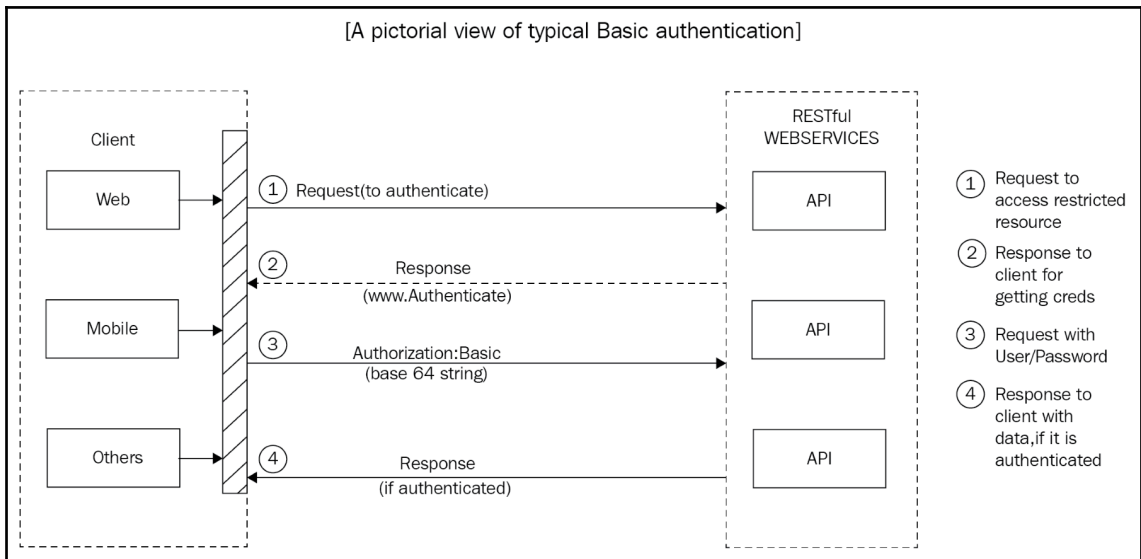
Request URL	
<code>http://localhost:42918/api/product/'4D261E4A-A657-4ADD-A0F6-DDE6E1464D55'</code>	
Server response	
Code	Details
200	Response body
	<pre>[{ "productId": "4d261e4a-a657-4add-a0f6-dde6e1464d55", "productName": "Learn C#", "productDescription": "Learn C# in 7 days", "productImage": "csharp.jpeg", "productPrice": 520, "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f", "categoryName": "Books", "categoryDescription": "Technical Books" }]</pre>

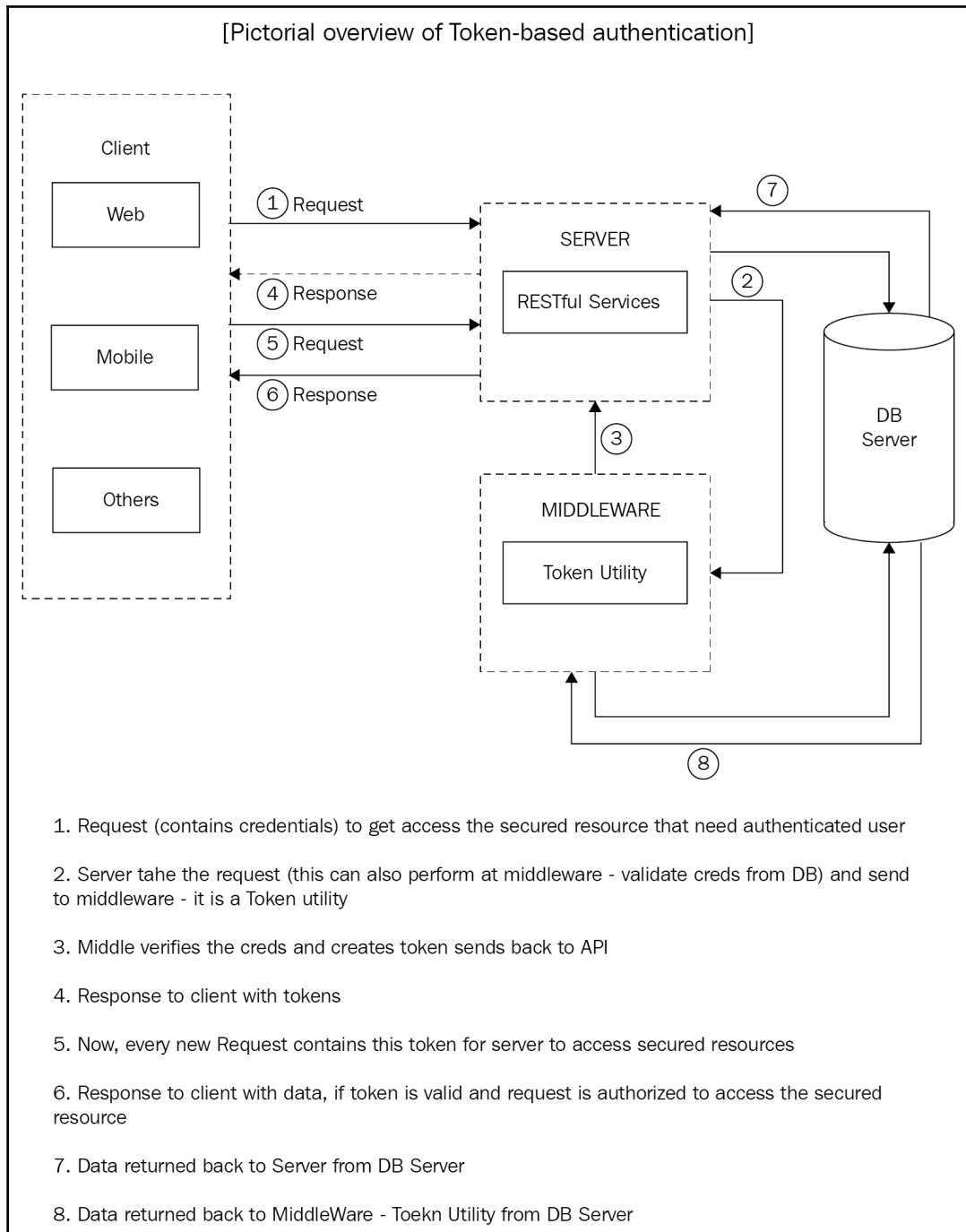
```
[
  {
    "productId": "02341321-c20b-48b1-a2be-47e67f548f0f",
    "productName": "Microservices for .NET",
    "productDescription": "Microservices for .NET Core",
    "productImage": "microservices.jpeg",
    "productPrice": 651,
    "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f",
    "categoryName": "Books",
    "categoryDescription": "Technical Books"
  },
  {
    "productId": "4d261e4a-a657-4add-a0f6-dde6e1464d55",
    "productName": "Learn C#",
    "productDescription": "Learn C# in 7 days",
    "productImage": "csharp.jpeg",
    "productPrice": 520,
    "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f",
    "categoryName": "Books",
    "categoryDescription": "Technical Books"
  }
]
```

```
Request URL
http://localhost:42918/api/product/'4D261E4A-A657-4ADD-A0F6-DDE6E1464D55'%20or%201%3D1
```

```
public IEnumerable<Product> GetByProduct(string id) => _context.Products
    .FromSql("SELECT * FROM dbo.Products WHERE id={0}", id)
    .Include(p => p.Category)
    .ToList();
```







POST /api/product/addproduct

Parameters Cancel

Name	Description
productvm <i>(body)</i>	<p>Example Value Model</p> <pre>{ "productId": "84974c9b-bb96-4553-8d56-6f6b197b6719", "productName": "Learn Python in 7-days", "productDescription": "Learning series on Python", "productImage": "microservices.jpeg", "productPrice": 651, "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f", "categoryName": "Books", "categoryDescription": "Technical Books" }</pre> <p>Cancel</p>

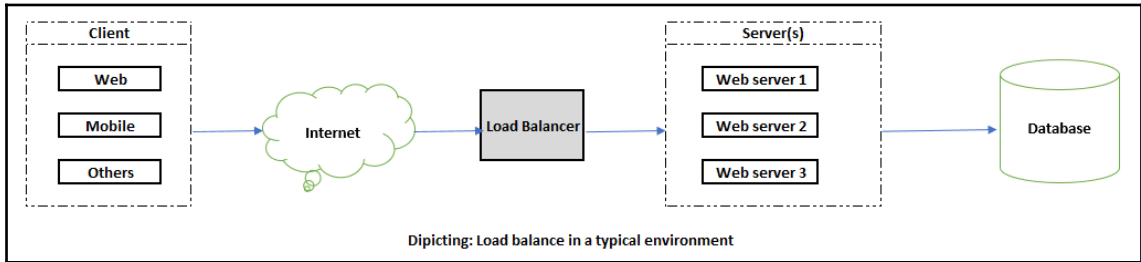
POST /api/product/addproduct

Parameters Cancel

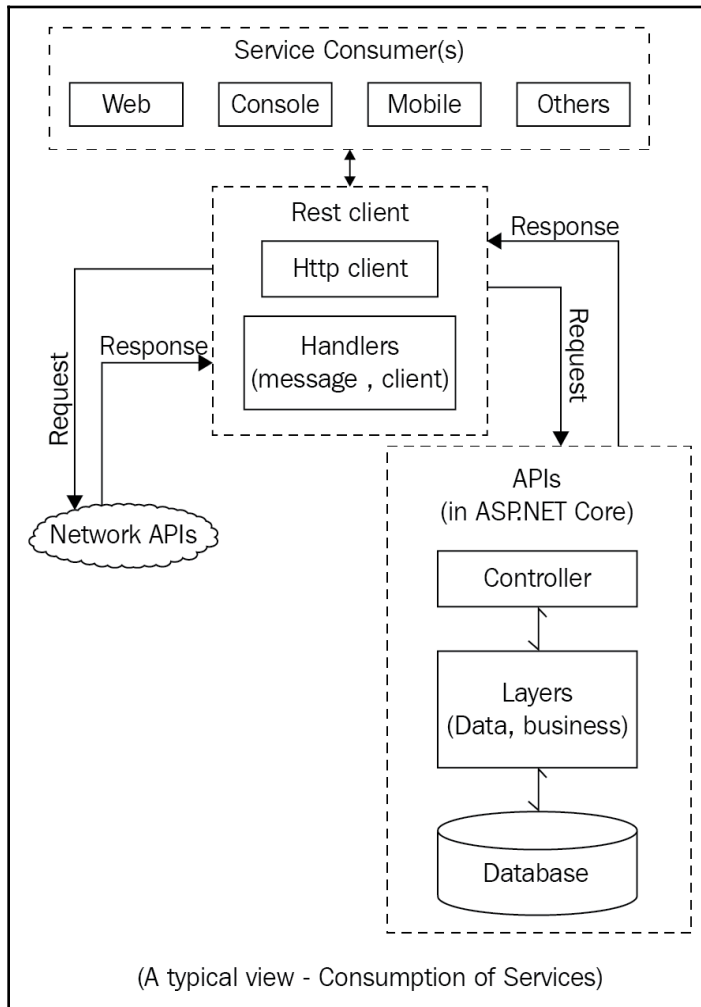
Name	Description
productvm <i>(body)</i>	<p>Example Value Model</p> <pre>{ "productId": "84974c9b-bb96-4553-8d56-6f6b197b6719", "productName": "", "productDescription": "Learning series on Python", "productImage": "microservices.jpeg", "productPrice": 651, "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f", "categoryName": "Books", "categoryDescription": "Technical Books" }</pre>

Server response	
Code	Details
400 <i>Undocumented</i>	Error: Bad Request Response body <pre>{ "ProductName": ["Product Name shoud not be empty."] }</pre>

Chapter 9: Scaling RESTful Services (Performance of Web Services)



Chapter 10: Building a Web Client (Consuming Web Services)



Product APIs ^{v1}

</swagger/v1/swagger.json>

The Api is to deal with Products.

[Terms of service](#)

Product



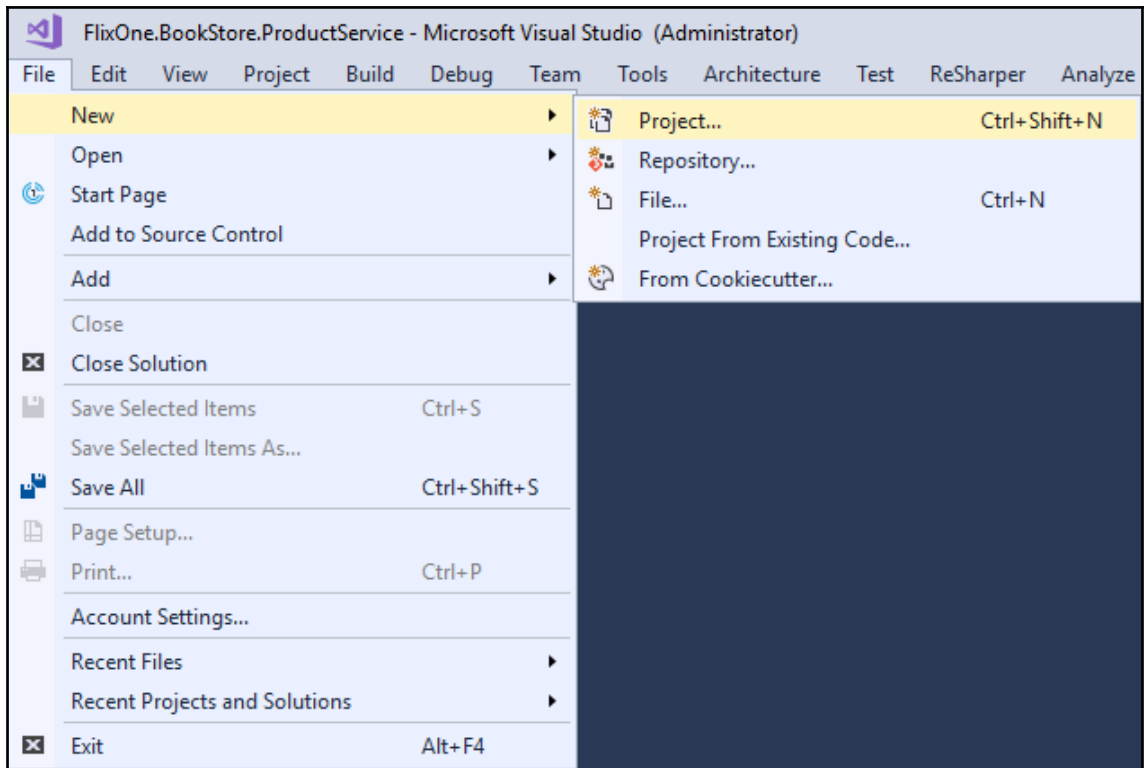
GET /api/product/productlist

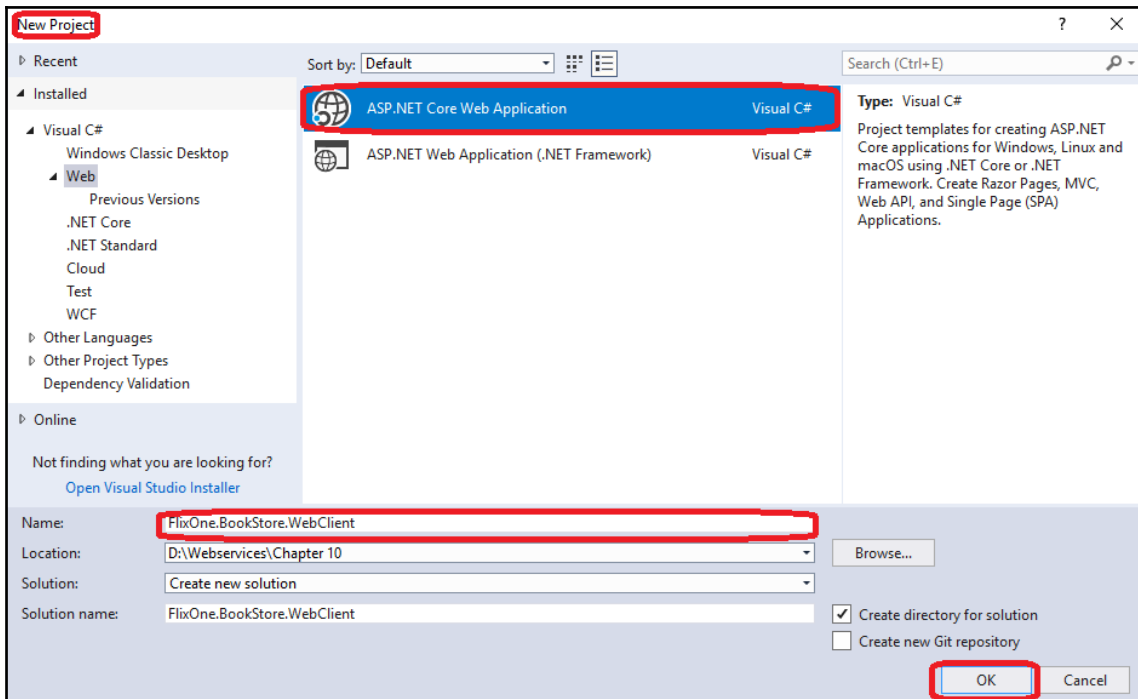
GET /api/product/product/{productid}

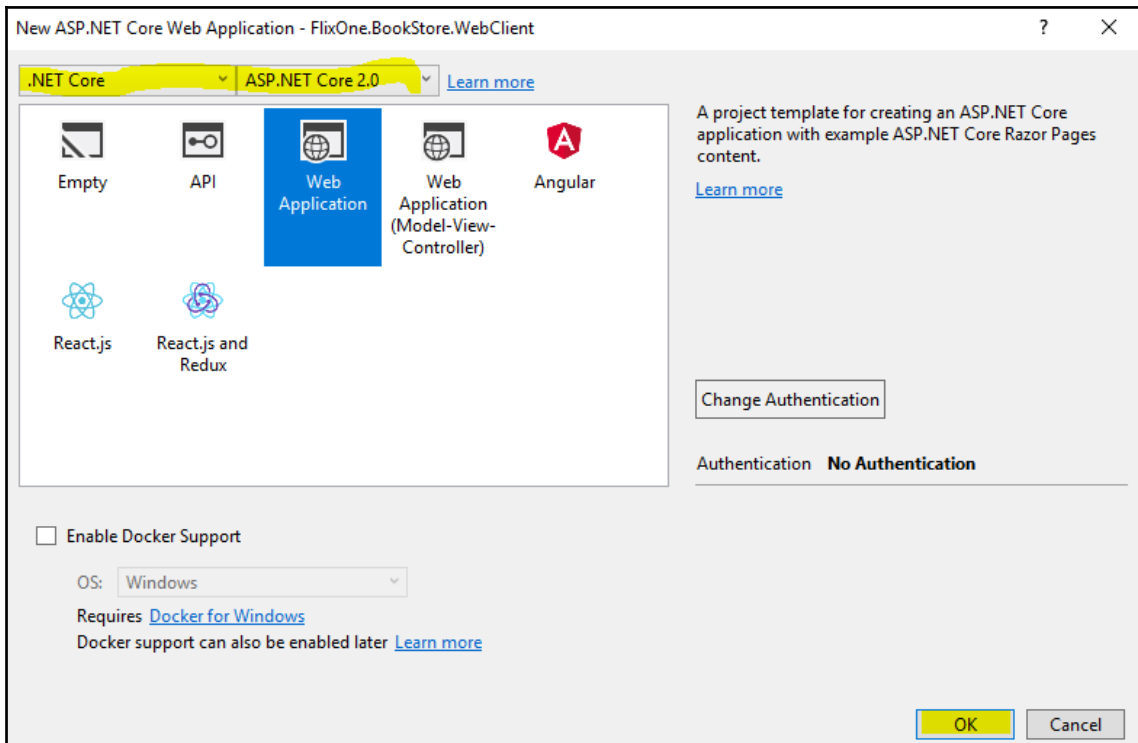
POST /api/product/addproduct

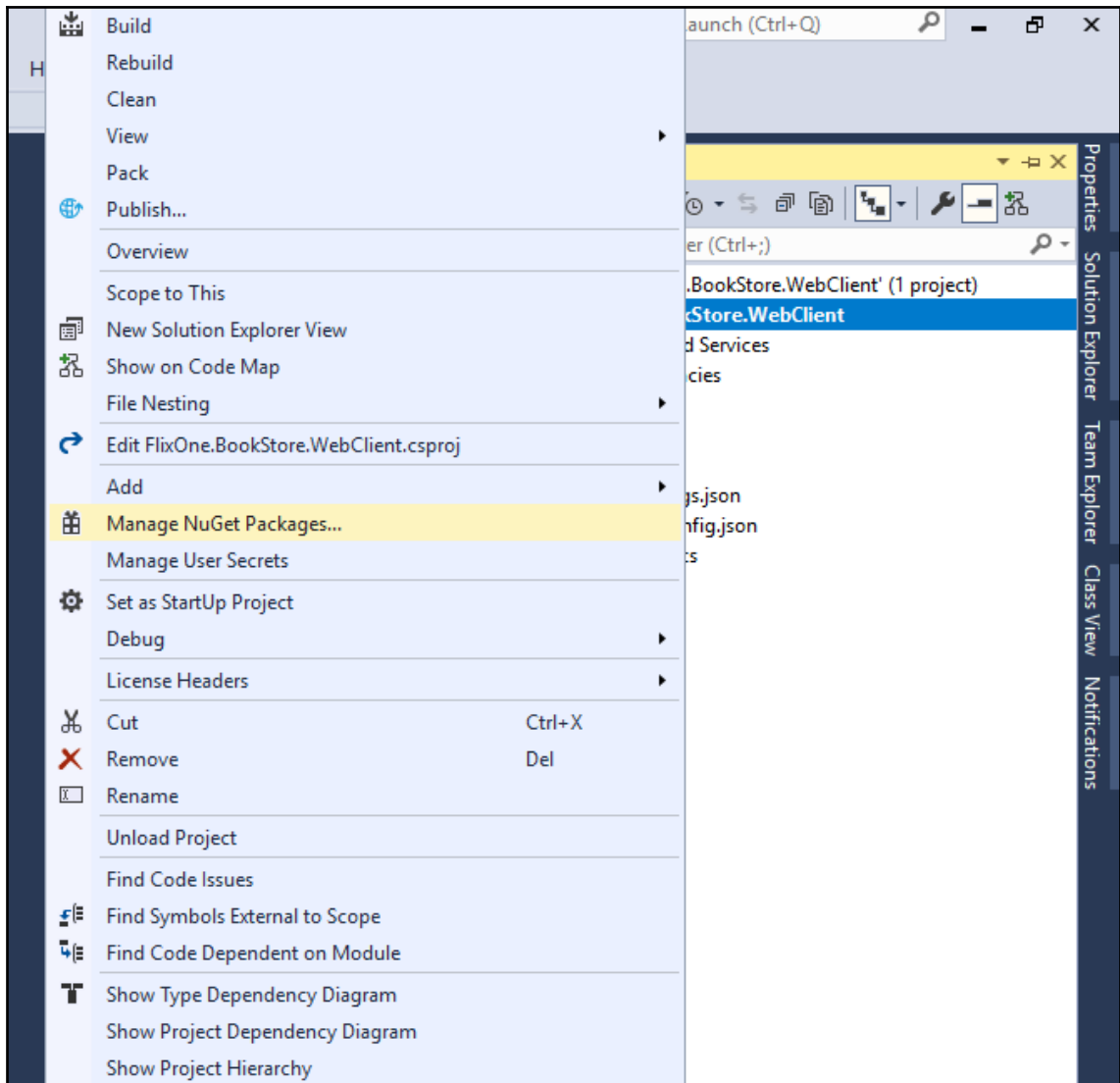
PUT /api/product/updateproduct/{productid}

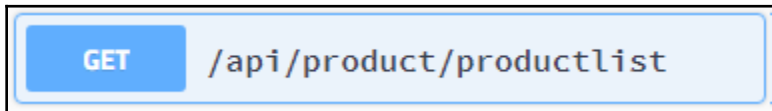
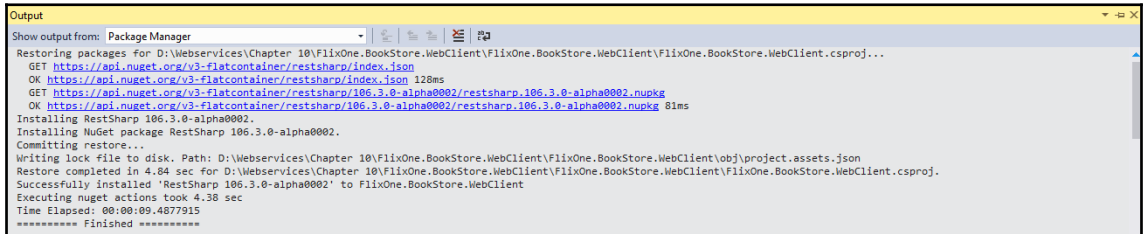
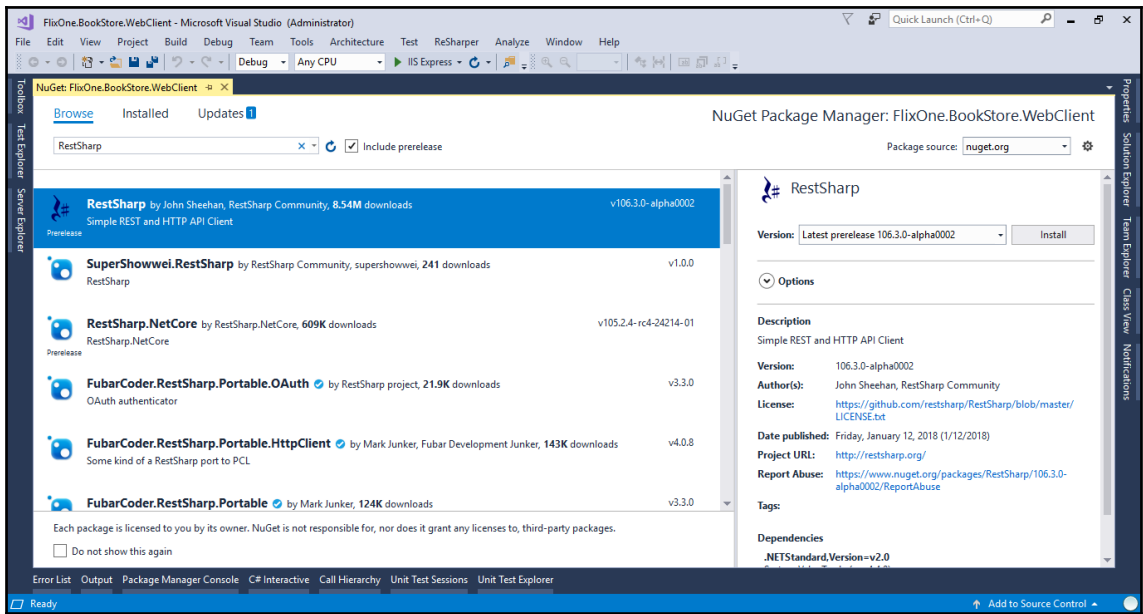
DELETE /api/product/deleteproduct/{productid}









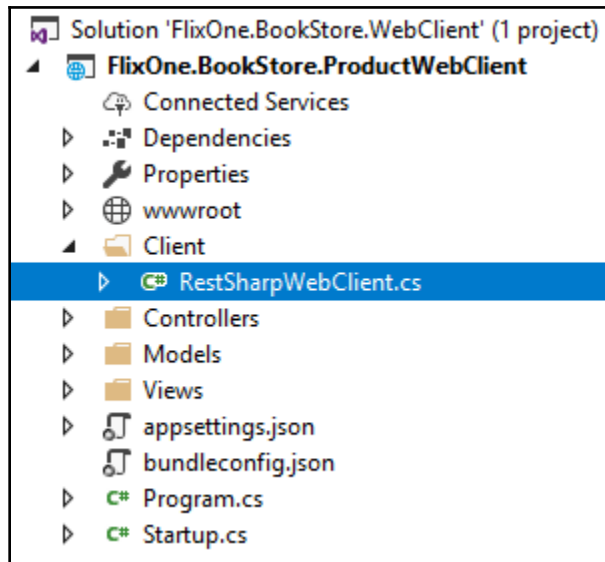


Request URL

```
http://localhost:10065/api/product/productlist
```

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "productId": "8dd81d72-3859-4376-b7de-a55d7a09627e", "productName": "Learn C# in 7 days", "productDescription": "Start learning C# in 7 days", "productImage": "Not available", "productPrice": 115, "categoryId": "880ac2ad-112b-4cac-a60f-348eefe7bbcc", "categoryName": "Tech books", "categoryDescription": "Technical books on various technology stacks." }, { "productId": "2bcfa359-8d49-4531-821d-40f4e04cc4df", "productName": "Next day after death", "productDescription": "Next day after death", "productImage": "Not available", "productPrice": 655, "categoryId": "9703106a-b473-47ea-b9d8-11681e33ad8c", "categoryName": "Story books", "categoryDescription": "Story books." }]</pre>



FlixOne.BookStore.ProductWebClient Home About Contact **Web Client**

Visual Studio | HTML5 | CSS3 | JS

There are powerful new features in Visual Studio for building modern web apps. [Learn More](#)

Application uses	How to	Overview	Run & Deploy
<ul style="list-style-type: none">• Sample pages using ASP.NET Core MVC• Theming using Bootstrap	<ul style="list-style-type: none">• Add a Controller and View• Manage User Secrets using Secret Manager.• Use logging to log a message.• Add packages using NuGet.	<ul style="list-style-type: none">• Conceptual overview of what is ASP.NET Core• Fundamentals of ASP.NET Core such as Startup and middleware.• Working with Data	<ul style="list-style-type: none">• Run your app• Run tools such as EF migrations and more• Publish to Microsoft Azure Web Apps

FlixOne.BookStore.ProductWebClient Home About Contact Web Client

Products List

[Create New](#)

ProductId	ProductName	ProductDescription	ProductImage	ProductPrice	CategoryId	CategoryName	CategoryDescription	
8dd81d72-3859-4376-b7de-a55d7a09627e	Learn C# in 7 days	Start learning C# in 7 days	Not available	115.00	880ac2ad-112b-4cac-a60f-348eefe7bbcc	Tech books	Technical books on various technology stacks.	Edit Details Delete
2bcfa359-8d49-4531-821d-40f4e04cc4df	Next day after death	Next day after death	Not available	655.00	9703106a-b473-47ea-b9d8-11681e33ad8c	Story books	Story books.	Edit Details Delete

© 2018 - FlixOne.BookStore.ProductWebClient

Chapter 11: Introduction to Microservices

