Chapter 1: Getting Started

REST based Service

Interoperate with service using XML

JAVA

POST

PHP

PUT

DELETE

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

SOAP Message Structure

- SOAP Envelope
  - SOAP Header
  - SOAP Body

- `<Soap: Envelope>`
- `<Soap: Header>`
- `<Soap: Body>`
- `</Soap: Body>`
- `</Soap: Envelope>`
Graphic Bundle
An unhandled exception occurred while processing the request.

AmbiguousActionException: Multiple actions matched. The following actions matched route data and had all constraints satisfied:

DemoECommerceApp.Controllers ValuesController.Get12 (DemoECommerceApp)
DemoECommerceApp.Controllers ValuesController.Get (DemoECommerceApp)

Microsoft.AspNetCore.Mvc.Internal.ActionSelector.SelectBestCandidate(RouteContext context, IReadOnlyList<ActionDescriptor> candidates)
An unhandled exception occurred while processing the request.

AmbiguousActionException: Multiple actions matched. The following actions matched route data and had all constraints satisfied:

DemoECommerceApp.Controllers.ValuesController.Get (DemoECommerceApp)

Microsoft.AspNetCore.Mvc.Internal.ActionSelector.SelectBestCandidate(RouteContext context, IReadOnlyList< IActionResultDescriptor > candidates)

Hello World

Step - 1

http://localhost:57571/api/products

Step - 2
**Graphic Bundle**

```javascript
var settings = {
    async: true,
    crossDomain: true,
    url: "http://localhost:57571/api/products/",
    method: "GET",
    headers: {
        "cache-control": "no-cache",
        "postman-token": "b4c0f3d4-7ed7-012f-96fc-5b373d8b299e"
    }
};
$.ajax(settings).done(function (response) {
    console.log(response);
});
```
**POST**

```plaintext
http://localhost:57571/api/products
```

<table>
<thead>
<tr>
<th>Authorization</th>
<th>Headers (1)</th>
<th>Body</th>
<th>Pre-request Script</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Key</strong></td>
<td><strong>Value</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Content-Type</strong></td>
<td>application/json</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PUT**

```plaintext
http://localhost:57571/api/products/1
```

<table>
<thead>
<tr>
<th>Authorization</th>
<th>Headers (1)</th>
<th>Body</th>
<th>Pre-request Script</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>form-data</strong></td>
<td>x-www-form-urlencoded</td>
<td>raw</td>
<td>binary</td>
</tr>
</tbody>
</table>

```json
{
  "Name": "Noodles",
  "Price": 12.9
}
```

**Body**

| Status: 200 OK | Time: 67 ms |

---

[14]
Monolithic versus SOA

**Monolithic**
- UI
- Business logic
- Data Access Layer
- Database

**SOA**
- UI
- Service
  - Database
- Service
  - Database
- Service
  - Database
- Another External App
Chapter 3: User Registration and Administration
Add Scaffold

- MVC Controller - Empty
- MVC Controller with read/write actions
- MVC Controller with views, using Entity Framework
- API Controller - Empty
- API Controller with read/write actions
- API Controller with actions, using Entity Framework

Click here to go online and find more scaffolding extensions.

Add API Controller with actions, using Entity Framework

- Model class: Customers (DemoECommerceApp.Models)
- Data context class: FlixOneStoreContext (DemoECommerceApp.Models)
- Controller name: CustomersController

Add | Cancel
Register for FlixOneStore

First Name: Tadit
Last Name: Dash
Email: tardictdash@gmail.com
Gender: Male
Date of Birth: 07/05/1990
Mobile Number: 9132994288
Fax: 9132994288
Password: ****
Confirm Password: ****
```javascript
$.ajax({
    url: 'http://localhost:57571/api/Customers',
    type: "POST",
    contentType: "application/json",
    data: JSON.stringify(customer),
    dataType: "json",
    success: function(result) {
        result = {id: "910d4c2f-b394-4578-8d9c-7ca3fd3266e2"}
        alert("A customer record created for: 
        + result.
    },
    error: function(err) {
        alert(err.responseJSON.message);
    }
});
```

- **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**
  - 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
GET    http://localhost:57571/api/Customers/3cd92f52-323d-43ca-a26b-0b13576612d9

Body   Cookies   Headers (6)   Test Results

<table>
<thead>
<tr>
<th>1</th>
<th>{}</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>&quot;id&quot;: &quot;3cd92f52-323d-43ca-a26b-0b13576612d9&quot;,</td>
</tr>
<tr>
<td>3</td>
<td>&quot;gender&quot;: &quot;M&quot;,</td>
</tr>
<tr>
<td>4</td>
<td>&quot;firstname&quot;: &quot;Tadit&quot;,</td>
</tr>
<tr>
<td>5</td>
<td>&quot;lastname&quot;: &quot;Dash&quot;,</td>
</tr>
<tr>
<td>6</td>
<td>&quot;dob&quot;: &quot;1990-07-05T00:00:00&quot;,</td>
</tr>
<tr>
<td>7</td>
<td>&quot;email&quot;: &quot;taditdash@<a href="mailto:test1234@gmail.com">test1234@gmail.com</a>&quot;,</td>
</tr>
<tr>
<td>8</td>
<td>&quot;mainaddressid&quot;: null,</td>
</tr>
<tr>
<td>9</td>
<td>&quot;telephone&quot;: &quot;9132994288&quot;,</td>
</tr>
<tr>
<td>10</td>
<td>&quot;fax&quot;: &quot;9132994288&quot;,</td>
</tr>
<tr>
<td>11</td>
<td>&quot;password&quot;: &quot;12345&quot;,</td>
</tr>
<tr>
<td>12</td>
<td>&quot;newsletteropted&quot;: false</td>
</tr>
<tr>
<td>13</td>
<td>}</td>
</tr>
</tbody>
</table>
My Profile

id: 910d4c2f-b394-4578-8d9c-7ca3fd3266e2

gender: M

firstname: Tadit

lastname: Dash

dob: 1990-07-05T00:00:00

email: taditdash@gmail.com

mainaddressid: null

te telephone: 9132994288

fax: 9132994288

password: 12345

newsletteropted: false
**General**
- Request URL: `http://localhost:57571/api/Customer/910D4C2F-B394-4578-8D9C-7CA3FD3266E2`
- Request Method: GET
- Status Code: 401 Unauthorized
- Remote Address: `::1:57571`
- Referrer Policy: no-referrer-when-downgrade

**Response Headers**
- 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"

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

---

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

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

### GET

| GET | http://localhost:57571/api/Customer/910D4C2F-B394-4578-8D9C-7CA3FD23268E2 | Send |

<table>
<thead>
<tr>
<th>Authorization</th>
<th>Headers</th>
<th>Body</th>
<th>Pre-request Script</th>
<th>Tests</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>New key</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Body</th>
<th>Cookies</th>
<th>Headers</th>
<th>Test Results</th>
</tr>
</thead>
</table>

**content-length**: 0

**date**: Mon, 07 May 2018 17:55:59 GMT

**server**: Kestrel

**www-authenticate**: Bearer

**x-powered-by**: ASP.NET

**x-sourcefiles**: =TUTF-8?

RTPcQmFp3VwXERio2tFb38cUGlja3RejUfUVGZbC87Zxi2aWNgYbpbwAtKvUFwNvexlJgM4wXFBByb2pY13RxXERibW9FQ29tsWVYV2VbH8erGrVw0VDb21ZXUjZUFwEFw0GlcQ2V

### POST

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

<table>
<thead>
<tr>
<th>Authorization</th>
<th>Headers</th>
<th>Body</th>
<th>Pre-request Script</th>
<th>Tests</th>
</tr>
</thead>
</table>

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

**grant_type**: password

**scope**: FlixEOneStore.ReadAccess&client_id=tn2.Client&client_secret=secret&username=taditdash@gmail.com&password=123456

<table>
<thead>
<tr>
<th>Body</th>
<th>Cookies</th>
<th>Headers</th>
<th>Test Results</th>
</tr>
</thead>
</table>

**Pretty** | Raw | Preview | JSON |

```
1 - {  
2   "error": "invalid_client"  
3  }
```
Graphic Bundle
public async Task GetProfileDataAsync(ProfileDataRequestContext profileContext)
{
    try
    {
        var customer = await _context.Customers
            .SingleOrDefaultAsync(m => m.Email == profileContext.Identity.Name);
        if (customer != null)
        {
            
            
            }
Chapter 4: Item Catalogue, Cart, and Checkout

![Image of directory structure]

```plaintext
- Models
  - Cart.cs
  - CartAttributes.cs
  - Categories.cs
  - Configuration.cs
  - ConfigurationGroup.cs
  - Countries.cs
  - Currencies.cs
  - CustomerInfo.cs
  - Customers.cs
  - FlixOneStoreContext.cs
  - Manufacturers.cs
  - ManufacturersInfo.cs
  - OrderProductAttributes.cs
  - Orders.cs
  - OrdersProducts.cs
  - OrdersStatus.cs
  - Products.cs
  - ProductsAttributes.cs
  - ProductsDetail.cs
  - ProductsOptions.cs
  - ProductsOptionsValues.cs
  - ProductsOptionsValuesMappings.cs
  - Reviews.cs
  - ReviewsDetail.cs
  - WholsOnline.cs
```
Graphic Bundle

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

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

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

Building RESTful Web services with .NET Core
Leveraging ASP.NET Core MVC and Visual Studio 17
Amazon Link
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

Building RESTful Web services with .NET Core
Leveraging ASP.NET Core MVC and Visual Studio 17
Amazon Link
Price: $39.99

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

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
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);
        }
    });
}

<table>
<thead>
<tr>
<th>id</th>
<th>CustomerId</th>
<th>ProductId</th>
<th>qty</th>
<th>FinalPrice</th>
<th>Addedon</th>
</tr>
</thead>
<tbody>
<tr>
<td>11FD1B87-7B9A-4C4C-B070-3E34F72D8827</td>
<td>910D4C2F-B394-4578-8D9C-7CA3FD3266E2</td>
<td>98A95BB6-C573-450D-A470-0A637E126DD7</td>
<td>4</td>
<td>199.96</td>
<td>2018-05-</td>
</tr>
</tbody>
</table>
### Place Order

#### Shipping Details

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Street Address</td>
<td>Gajapati nagar</td>
</tr>
<tr>
<td>City</td>
<td>Bhubaneswar</td>
</tr>
<tr>
<td>State</td>
<td>Odisha</td>
</tr>
<tr>
<td>Postal Code</td>
<td>751005</td>
</tr>
<tr>
<td>Country</td>
<td>India</td>
</tr>
</tbody>
</table>

[Submit] [Close]
Graphic Bundle
"deliverycountry": "India",
"paymentmethodid": "a1df61de-d9a0-4ca9-8cc2-26a16e8bfe14",
"latsmodified": "2018-05-17T17:53:11.177",
"datepurchased": "2018-05-17T17:53:11.177",
"shippingcost": 0,
"shippingmethodid": "65bf649d-f6d2-41d8-a30e-3d6d8d00882f",
"orderstatus": "Placed",
"orderdatefinished": "2018-05-17T17:53:11.177",
"comments": null,
"currency": "$",
"currencyValue": 0,
"customer": null,
"orderProductAttributes": [],
"ordersProducts": [
    {
        "id": "64c24ba5-791c-46d0-9d05-aec88c4f6540",
        "orderid": "0b4058fd-a6bb-42bc-bc0e-305edf8704a6",
        "productid": "ee41beee-9800-4bee-ac84-20f1d32c4a53",
        "productname": "Building RESTful Web services with .NET Core",
        "productprice": 39.99,
        "finalprice": 39.99,
        "producttax": 0,
        "productqty": 1,
        "product": null,
        "orderProductAttributes": []
    },
    {
        "id": "eb766d628-7df6-4663-81ca-c5947366acef",
        "orderid": "0b4058fd-a6bb-42bc-bc0e-305edf8704a6",
        "productid": "98a95bb6-c573-450d-a470-0a637e126dd7",
        "productname": "Dependency Injection in .NET Core 2.0",
    }
]
Chapter 5: Integrating External Components and Handling

[Example of a typical middleware component]
Graphic Bundle
**Graphic Bundle**

---

```text
(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.
```

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

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

---

This ends the request or short circuit request.

---

```text
(extension) IApplicationBuilder IApplicationBuilder.Map(PathString pathMatch, Action<> 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.
```

---

This ends the request or short circuit request.

---

This is called from testroute. This ends the request or short circuit request.
Graphic Bundle

Product APIs

```
Product

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/api/Product/productlist</td>
</tr>
<tr>
<td>GET</td>
<td>/api/Product/product/{productid}</td>
</tr>
<tr>
<td>POST</td>
<td>/api/Product/addproduct</td>
</tr>
<tr>
<td>PUT</td>
<td>/api/Product/updateproduct/{productid}</td>
</tr>
<tr>
<td>DELETE</td>
<td>/api/Product/deleteproduct/{productid}</td>
</tr>
</tbody>
</table>
```
Graphic Bundle

```json
[
{
    "productId": "02341321-c20b-48b1-a2be-47e67f548f0f",
    "productName": "Microservices for .NET",
    "productDescription": "Microservices for .NET Core",
    "productImage": "microservices.jpeg",
    "productPrice": 651,
    "categoryId": "5caca9d0-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": "5caca9d0-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)
Graphic Bundle

```plaintext
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.

Build succeeded.
0 Warning(s)
0 Error(s)
Time Elapsed 00:00:02.96
```

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

Body

```
```
D:\Websevices\Chapter05\02 Logging\ Chap05_02\Properties\launchSettings.json

Using launch settings from D:\Websevices\Chapter05\02 Logging\ Chap05_02\Properties\launchSettings.json...

User profile is available. Using 'C:\Users\gjuran\AppData\Local\ASP.NET\DataProtection-Keys' as key repository and Windows DPAPI to encrypt keys at rest.

Hosting environment: Development

Now listening on: http://localhost:60431

Application started. Press Ctrl+G to shut down.

Request starting HTTP/1.1 GET http://localhost:60431/api/Product/productlist

Executing action method Chap05_02.Controllers.ProductController.Getlist ( Chap05_02) with arguments ((null))

- ModelState is Valid

Executing DbCommand ( sms) [Parameters=[]] CommandType='Text', CommandType='storedProcedure'

SELECT [c].[Id], [c].[CategoryId], [c].[Description], [c].[Image], [c].[Name], [c].[Price], [c].[Category].[Id]
FROM [Products] AS [c]
INNER JOIN [Categories] AS [c.Category] ON [c].[CategoryId] = [c.Category].[Id]


Executed action Chap05_02.Controllers.ProductController.Getlist ( Chap05_02) in 411.8216 ms

Graphic Bundle
Graphic Bundle
Graphic Bundle
Chapter 6: Testing RESTful Web Services

**Requirement**
Gather the requirement for Book - Building RESTful Webservices using .NET Core 2.0

**Outline**
Layout outline of the book, as per initial proposal of the book

**Agreement**
Start process of agreement between publisher and author

**Preliminary Drafts**
Start draft chapters from authors within the timeline

**Final Drafts**
After content editor accept the final drafts

**Technical Review**
Appoint a technical consultant

---

Hierarchical representation of Use case, Test Scenarios, and Test Case
GUI Tests (REST Client Tests)

RESTful services tests (Acceptance tests)

Unit tests
localhost:60431/swagger/
Product APIs

/ swagger/v1/swagger.json

Product

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/api/Product/productlist</td>
</tr>
<tr>
<td>GET</td>
<td>/api/Product/product/{productid}</td>
</tr>
<tr>
<td>POST</td>
<td>/api/Product/addproduct</td>
</tr>
<tr>
<td>PUT</td>
<td>/api/Product/updateproduct/{productid}</td>
</tr>
<tr>
<td>DELETE</td>
<td>/api/Product/deleteproduct/{productid}</td>
</tr>
</tbody>
</table>
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"
  }
]

Graphic Bundle
Test Explorer

Chap06_01 (1 tests)

- Passed Tests (1)
  - Chap06_01_Test.Services.ProductTests.Get_Returns_ActionResults
  - 328 ms

Summary

Last Test Run Passed
(Total Run Time 0:00:06.9099995)

1 Test Passed
Graphic Bundle
Graphic Bundle
private ProductViewModel ToProductViewModel(Product productModel)
{
    s2ms elapsed
    return new ProductViewModel
    {
        CategoryId = productModel.CategoryId,
        CategoryDescription = productModel.Category.Description,
        CategoryName = productModel.Category.Name,
        ProductId = productModel.Id,
        ProductImage = productModel.Image,
        ProductName = productModel.Name,
        ProductPrice = productModel.Price
    };
}

GET http://localhost:8043/api/ProductList

Authorization

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

Body

```
{
    "productModel": [{
        "id": 1,
        "name": "Microservices for .NET Core",
        "description": "Microservices for .NET Core",
        "image": "microservices.jpg"
    },
    {"id": 2,
        "name": "Books",
        "description": "Technical Books"
    }]
}
```

History

Collections

Clear all

Today

GET http://localhost:8043/api/ProductList
Graphic Bundle

Test Explorer

Run All | Run... | Playlist: All Tests

Chap06_04 (2 tests)

Passed Tests (2) 139 ms

✓ Chap06_04_Test.Services.ProductTest.MultipleCallRequestTime 1 sec
✓ Chap06_04_Test.Services.ProductTest.SingleCallRequestTime 19 ms

Summary

Last Test Run Passed (Total Run Time 0:00:10.3759999)
✓ 2 Tests Passed
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

Test Run Successful.

Total tests: 2. Passed: 2. Failed: 0. Skipped: 0.

Test execution time: 5.0981 Seconds

D:\Webservices\Chapter 06_new\04 Load Tests\ Chap06_04_Test>
Chapter 7: Continuous Integration and Continuous Deployment

[SaaS
(Server Storage, Operating Systems, Development tools, Database management, Business analytics, Hosted applications)]

[PaaS
(Server Storage, Operating Systems, Development tools, Database management, Business analytics)]

[IaaS
(Server and Storage)]

[Cloud computing services models (IaaS, PaaS, SaaS)]
Create new project

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

Project name *

FlixOneProductServices

Description

This project is for demonstration of .NET Core Web services book to show case the power of VSTS

Version control

Team Foundation Version Control

Work item process

Scrum
Graphic Bundle

Create simple RESTful API to show the product details

Description

Add complete description here

Acceptance Criteria

Details

Development

Related Work

There are no links in this group.
Edit iteration

Sprint 1
Iteration name
Sprint 1

Start date
4/23/2018

End date
5/4/2018

Location
FlixOneProductServices

Save and close  Cancel
Graphic Bundle
Add Solution FlixOneProductServices to Source Control

Indicate where to store your solution and projects in the Team Foundation Server and in your workspace.

Team Foundation Server Details

Server: gaurav-arora.visualstudio.com\gaurav-arora

Team Project Location:

- FlixOneProductServices
- BuildProcessTemplates

Make New Folder

Type a name for the solution folder:
FlixOneProductServices

Solution and project files will be added to:
$/FlixOneProductServices/FlixOneProductServices

Workspace:
GAURAV

[OK] [Cancel]
Graphic Bundle
Create a test plan

Name: Iteration 1 Test
Area path: FlixOneProductServices
Iteration: FlixOneProductServices\Sprint 1
April 23 - May 4

Create Cancel
Graphic Bundle
Graphic Bundle
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.

Select a Template
Or start with an Empty process

Featured
- Azure App Service Deployment
- Deploy Node.js App to Azure App Service
- Deploy PHP App to Azure App Service
- IIS Website and SQL Database Deployment
- Azure Cloud Service Deployment
Graphic Bundle

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
Graphic Bundle
Chapter 8: Securing RESTful Web Services

① For Secured system, no impact of attacks,

② WEAK System leads to security breaches and it impacts the other components too.

③ Components are under attack due to weak security
[Depicting: a typical view of Authenticating Web API using Auth Services]
Graphic Bundle

```csharp
// POST: Comment/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create(IFormCollection collection)
{
    try
    {
        var coll = collection;
    }
    catch
    {
    }
    return 1;
}
```

Index

Create New

<table>
<thead>
<tr>
<th>UserName</th>
<th>Description</th>
<th>Edit</th>
<th>Details</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>aroraG</td>
<td>This is a first comment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>aroraG</td>
<td>This is a second comment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>aroraG</td>
<td>This is a third comment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>aroraG</td>
<td><code>&lt;script&gt;alert('hi!');&lt;/script&gt;</code></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2018 - Chap08_01

localhost:36944 says

hi!

OK
A) Request injected by attacker http:\\localhost:6634/?empid="232 or 1=1;"

B) API passes it to Server example Select * from model emp where'id=232 or 1=1;";

C) DB execute a Statement example Select * from Emp where id='232' or 1=1

[Depicting- A typical SQL Injection Scenario]
Request URL


Server response

<table>
<thead>
<tr>
<th>Code</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td></td>
</tr>
</tbody>
</table>

Response body

```json
[
  {
    "productId": "4d261e4a-a657-4add-a8f6-dde6e1464d55",
    "productName": "Learn C#",
    "productDescription": "Learn C# in 7 days",
    "productImage": "csharp.jpeg",
    "productPrice": 528,
    "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f",
    "categoryName": "Books",
    "categoryDescription": "Technical Books"
  }
]
```

```json
[
  {
    "productId": "02341321-c20b-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-a8f6-dde6e1464d55",
    "productName": "Learn C#",
    "productDescription": "Learn C# in 7 days",
    "productImage": "csharp.jpeg",
    "productPrice": 528,
    "categoryId": "5ccaa9d0-e436-4d1e-a463-b45696d73a9f",
    "categoryName": "Books",
    "categoryDescription": "Technical Books"
  }
]
```
public IEnumerable<Product> GetByProduct(string id) => _context.Products
.FromSql("SELECT * FROM dbo.Products WHERE id={0}", id)
.Include(p => p.Category)
.ToList();

System.Data.SqlClient.SqlException: 'Conversion failed when converting from a character string to uniqueidentifier.'

[Pictorial overview of a typical XSRF attack]
[A pictorial view of typical Basic authentication]

Client
- Web
  1. Request (to authenticate)
  2. Response (www.Authenticate)
- Mobile
  3. Authorization: Basic (base 64 string)
- Others
  4. Response (if authenticated)

RESTful WEB SERVICES

1. Request to access restricted resource
2. Response to client for getting creds
3. Request with User/Password
4. Response to client with data, if it is authenticated
1. Request (contains credentials) to get access the secured resource that need authenticated user

2. Server take the request (this can also perform at middleware - validate creds from DB) and send to middleware - it is a Token utility

3. Middleware verifies the creds and creates token sends back to API

4. Response to client with tokens

5. Now, every new Request contains this token for server to access secured resources

6. Response to client with data, if token is valid and request is authorized to access the secured resource

7. Data returned back to Server from DB Server

8. Data returned back to MiddleWare - Token Utility from DB Server
Graphic Bundle

POST /api/product/addproduct

Parameters

Name | Description
--- | ---
productvm | (body)

Example Value | Model
--- | ---
{
"productId": "84974c9b-bb56-4553-8d56-6f6b197b6719",
"productName": "Learn Python in 7-days",
"productDescription": "Learning series on Python",
"productImage": "microservices.jpeg",
"productPrice": 651,
"categoryId": "5cda9d6-e436-4d1e-a463-b45696d73a9f",
"categoryName": "Books",
"categoryDescription": "Technical Books"
}

POST /api/product/addproduct

Parameters

Name | Description
--- | ---
productvm | (body)

Example Value | Model
--- | ---
{
"productId": "84974c9b-bb56-4553-8d56-6f6b197b6719",
"productName": "Learn Python in 7-days",
"productDescription": "Learning series on Python",
"productImage": "microservices.jpeg",
"productPrice": 651,
"categoryId": "5cda9d6-e436-4d1e-a463-b45696d73a9f",
"categoryName": "Books",
"categoryDescription": "Technical Books"
}
<table>
<thead>
<tr>
<th>Code</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Error: Bad Request</td>
</tr>
</tbody>
</table>

**Response body**

```json
{
  "ProductName": [
    "Product Name should not be empty."
  ]
}
```
Chapter 9: Scaling RESTful Services (Performance of Web Services)

Diagram: Load balancing in a typical environment
Chapter 10: Building a Web Client (Consuming Web Services)

(A typical view - Consumption of Services)
Product APIs

/swagger/v1/swagger.json

The API is to deal with Products.

Terms of service

**Product**

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/api/product/productlist</td>
</tr>
<tr>
<td>GET</td>
<td>/api/product/product/{productid}</td>
</tr>
<tr>
<td>POST</td>
<td>/api/product/addproduct</td>
</tr>
<tr>
<td>PUT</td>
<td>/api/product/updateproduct/{productid}</td>
</tr>
<tr>
<td>DELETE</td>
<td>/api/product/deleteproduct/{productid}</td>
</tr>
</tbody>
</table>
Graphic Bundle
GET /api/product/productlist
<table>
<thead>
<tr>
<th>Code</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td></td>
</tr>
</tbody>
</table>

```
[{
  "productId": "8dd8d1d72-3859-4376-b7de-a55d7a09627e",
  "productName": "Learn C# in 7 days",
  "productDescription": "Start learning C# in 7 days",
  "productImage": "Not available",
  "productPrice": 115,
  "categoryId": "880ac2ad-112b-4cac-a69f-348cefc7bbcc",
  "categoryName": "Tech books",
  "categoryDescription": "Technical books on various technology stacks."
},
{
  "productId": "2bca5539-8d49-4531-821d-404e04cc4df",
  "productName": "Next day after death",
  "productDescription": "Next day after death",
  "productImage": "Not available",
  "productPrice": 655,
  "categoryId": "983106a-b473-47ea-b9d8-11e686e33ad8c",
  "categoryName": "Story Books",
  "categoryDescription": "Story books."
}]
```
<table>
<thead>
<tr>
<th>ProductId</th>
<th>ProductName</th>
<th>ProductDescription</th>
<th>ProductImage</th>
<th>ProductPrice</th>
<th>CategoryId</th>
<th>CategoryName</th>
<th>CategoryDescription</th>
<th>Details</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>8dd81d72-3859-4076-b70e-a55d7a09927e</td>
<td>Learn C# in 7 days</td>
<td>Not available</td>
<td></td>
<td>115.00</td>
<td>860ae2ad-112b-4cc-a65f-341e6f780c8c</td>
<td>Tech books</td>
<td>Technical books on various technology stacks.</td>
<td>Edit</td>
<td>Details</td>
</tr>
<tr>
<td>2bcsa359-8d5a-4531-621d-40f4e04cc4df</td>
<td>Next day after death</td>
<td>Next day after death</td>
<td>Not available</td>
<td>655.00</td>
<td>970310fa-b473-47ea-b6db-11581e33adlic</td>
<td>Story books</td>
<td>Story books.</td>
<td>Edit</td>
<td>Details</td>
</tr>
</tbody>
</table>

Chapter 11: Introduction to Microservices

Why to go with Microservices

- Scalability
- Technology Adoption
- Deployment
- Business
Graphic Bundle

(Depicting - a typical Microservices Architectural Style)