Chapter 1: An Introduction to Microservices
Chapter 2: Refactoring the Monolith

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\aroraG>dotnet --info
.NET Core SDK (reflecting any global.json):
Version: 3.1.100
Commit: cd82f021f4

Runtime Environment:
OS Name: Windows
OS Version: 10.0.14393
OS Platform: Windows
RID: win10-x64
Base Path: C:\Program Files\dotnet\sdk\3.1.100\

Host (useful for support):
Version: 3.1.0
Commit: 65f04fb6db

PS D:\01 Bitbucket\Hands-On-Microservices-with-C-8-and-.NET-Core-3.0-Third-Edition\Chapter 02> dotnet new winforms
Getting ready...
The template "Windows Forms (WinForms) Application" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on D:\01 Bitbucket\Hands-On-Microservices-with-C-8-and-.NET-Core-3.0-Third-Edition\Chapter 02\Chapter 02.csproj...
Restore completed in 245.33 ms for D:\01 Bitbucket\Hands-On-Microservices-with-C-8-and-.NET-Core-3.0-Third-Edition\Chapter 02\Chapter 02.csproj.
Restore succeeded.

PS D:\01 Bitbucket\Hands-On-Microservices-with-C-8-and-.NET-Core-3.0-Third-Edition\Chapter 02> dotnet run

Form1
The template "WPF Application" was created successfully.

Running 'dotnet restore' on D:\01 Bitbucket\Hands-On-Microservices-with-C-8-and-.NET-Core-3.0-Third-Edition\Chapter 02\WPFSample> dotnet new wpf

Restore completed in 77.46 ms for D:\01 Bitbucket\Hands-On-Microservices-with-C-8-and-.NET-Core-3.0-Third-Edition\Chapter 02\WPFSample\WPFSample.csproj.

Restore succeeded.

PS D:\01 Bitbucket\Hands-On-Microservices-with-C-8-and-.NET-Core-3.0-Third-Edition\Chapter 02\WPFSample> dotnet run

MainWindow
To prevent possible data loss before loading the designer, the following errors must be resolved:

1 Error

Could not load file or assembly 'System.ComponentModel.TypeConverter, Version=4.2.1.0, Culture=neutral, PublicKeyToken=b01f3ffdeadb7e73' or one of its dependencies. The system cannot find the file specified.

Instances of this error (1)
1. Show Call Stack

Help with this error
Could not find an associated help topic for this error. Check Windows Forms Design-Time error list

Forum posts about this error
Search the MSDN Forums for posts related to this error

Options

The following preview features are available for you to try. Your feedback will help us to both make them better and decide whether to include them in the next version of Visual Studio.

- **Use compact menu and search bar (requires restart)**
  - Removes the titlebar and shows Visual Studio with a compact layout.

- **Use the preview Windows Forms designer for .NET Core apps (requires restart)**
  - If checked then the preview Windows Forms Designer for .NET Core apps is used.

- **Vertical document tab layout**
  - Places sortable document tabs to the left or right of the editor window. Adds a Set Tab Layout option in the document tab context menu and a Document Tabs section under Tools > Options > Tabs and Windows.

More Info  Give Feedback

OK  Cancel
### New features of C#8.0

<table>
<thead>
<tr>
<th>Name</th>
<th>index from start</th>
<th>index from end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microservices</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Design Patterns</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Test Driven Development</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Data Sciences</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Functional Programming</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Concurrent Programming</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

### Indices and Ranges

<table>
<thead>
<tr>
<th>Name</th>
<th>Index from start</th>
<th>Index from end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microservices</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Design Patterns</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Test Driven Development</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Data Sciences</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Functional Programming</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Concurrent Programming</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

First element of array (index from start): Books{0} => Microservices
First element of array (index from end): Books{^6} => Microservices
Indices and Ranges

<table>
<thead>
<tr>
<th>Name</th>
<th>Index from start</th>
<th>Index from end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microservices</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Design Patterns</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Test Driven Development</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Data Sciences</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Functional Programming</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Concurrent Programming</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

First element of array (index from start): Books[0] => Microservices

Indices and Ranges

<table>
<thead>
<tr>
<th>Name</th>
<th>Index from start</th>
<th>Index from end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microservices</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Design Patterns</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Test Driven Development</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Data Sciences</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Functional Programming</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Concurrent Programming</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

First element of array (index from start): Books[0] => Microservices


Design Patterns
Test Driven Development
Data Sciences

Readonly members

Add 5 + 6 = 11
Still a monolith

Database

<table>
<thead>
<tr>
<th>ORDER</th>
<th>PRODUCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OrderID</td>
<td>ProductID</td>
</tr>
<tr>
<td>OrderID</td>
<td>ProductID</td>
</tr>
<tr>
<td>OrderID</td>
<td>ProductID</td>
</tr>
</tbody>
</table>

Removing foreign keys

Database

<table>
<thead>
<tr>
<th>ORDER</th>
<th>PRODUCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OrderID</td>
<td>ProductID</td>
</tr>
<tr>
<td>OrderID</td>
<td>ProductID</td>
</tr>
<tr>
<td>OrderID</td>
<td>ProductID</td>
</tr>
</tbody>
</table>

Book: Design Patterns has Price: 350.00
The image contains a database schema diagram showing the relationship between two tables: `Product` and `Category`.

**Product (product) Table**
- **Id** (uniqueidentifier)
- **Name** (nvarchar(50))
- **Description** (nvarchar(250))
- **Image** (varchar(150))
- **Price** (decimal(14,2))
- **CategoryId** (uniqueidentifier)

**Category (product) Table**
- **Id** (uniqueidentifier)
- **Name**
- **Description**

The schema includes:
- **Keys** (1): PK_Product (Primary Key, Clustered: Id)
- **Check Constraints** (0)
- **Indexes** (0)
- **Foreign Keys** (1): FK_Product_Category (ID)
- **Triggers** (0)
Product service

Micro component

Product repository

Product database
Create a new ASP.NET Core web application

.NET Core ▼ ASP.NET Core 3.1 ▼

API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

Web Application
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

Web Application (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

Angular
A project template for creating an ASP.NET Core application with Angular

React.js
A project template for creating an ASP.NET Core application with React.js

React.js and Redux
A project template for creating an ASP.NET Core application with React.js and Redux

Get additional project templates

Advanced
- Configure for HTTPS
- Enable Docker Support (Requires Docker Desktop)

Author: Microsoft
Source: .NET Core 3.1.0

Solution 'FlixOne.BookStore.ProductService' (1 of 1)
- Connected Services
  - Dependencies
  - Properties
  - wwwroot
  - Controllers
  - Models
  - Views
  - appsettings.json
  - Program.cs
  - Startup.cs
Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.

FlixOne.BookStore.ProductService

Installing:
- Microsoft.AspNetCore.Authentication.Abstractions.2.0.0
- Microsoft.AspNetCore.Authentication.Core.2.0.0
- Microsoft.AspNetCore.Authorization.2.0.0
- Microsoft.AspNetCore.Authorization.Policy.2.0.0
- Microsoft.AspNetCore.Hosting.Abstractions.2.0.0
- Microsoft.AspNetCore.Hosting.Server.Abstractions.2.0.0
- Microsoft.AspNetCore.Http.2.0.0
- Microsoft.AspNetCore.Http.Abstractions.2.0.0
- Microsoft.AspNetCore.Http.Extensions.2.0.0
- Microsoft.AspNetCore.Http.Features.2.0.0
- Microsoft.AspNetCore.JsonPatch.3.0.0-preview3-19153-02
- Microsoft.AspNetCore.Mvc.Abstractions.2.0.0
- Microsoft.AspNetCore.Mvc.Core.2.0.0
- Microsoft.AspNetCore.Mvc.NewtonsoftJson.3.0.0-preview3-19153-02
- Microsoft.AspNetCore.ResponseCaching.Abstractions.2.0.0
- Microsoft.AspNetCore.Routing.2.0.0
- Microsoft.AspNetCore.Routing.Abstractions.2.0.0
- Microsoft.AspNetCore.WebUtilities.2.0.0
- Microsoft.DotNet.PlatformAbstractions.2.0.0

Do not show this again  OK  Cancel
<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>
Chapter 3: Effective Communication between Services

Rest API - Pictorial overview of sync communication
Pictorial view: Process of purchasing a book

1. Customer (Searches for book)
2. Book selection (Customers select a book)
3. Proceed to order (Customer adds, selected book to cart)
4. Less available book count
5. Checkout
6. Proceed for payment
7. Generate invoice and send to customer
8. Order tracking
Pictorial overview: Inter-service communication for order process
Pictorial overview: API Gateway
PICTORIAL OVERVIEW: Event-Driven pattern

PICTORIAL OVERVIEW: How OrderService Event system works
Application instance and it generates messages

Message Queue

Consumer service and it processes messages

Pictorial Overview: Message queue
Pictorial overview: Azure Service Bus
Starting...
Ending...
Press any key...
Starting...
Ending...
Press any key...
Chapter 4: Testing Microservices with the Microsoft Unit Testing Framework

Testing pyramid

Unit tests
Service tests
System tests
Recording consumer contract
Replaying consumer contract
Add a new project

Recent project templates
A list of your recently accessed templates will be displayed here.

Configure your new project

xUnit Test Project (.NET Core)  C#  Windows  Linux  macOS  Test

Project name

Location

Back  Create
Chapter 5: Deploying Microservices with Docker

Standard monolith release cycle

Microservices release cycle - Share nothing architecture
Microservice Deployment Workflow with Docker using Azure DevOps
Hyper-V feature is not enabled.
Do you want to enable it for Docker to be able to work properly?
Your computer will restart automatically.

Settings

General

Shared Drives

Select the local drives you want to be available to your containers.

<table>
<thead>
<tr>
<th>Shared</th>
<th>Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>C</td>
</tr>
</tbody>
</table>

Microsoft PowerShell

> docker run --rm -v c:/Users:/data alpine ls /data

Reset credentials...

Apply
Create a new project

Recent project templates

- ASP.NET Core Web Application
- xUnit Test Project (.NET Core)
- ASP.NET Core Web Application
- gRPC Service
- Razor Class Library
- ASP.NET Web Application (.NET Framework)

Configure your new project

ASP.NET Core Web Application

Project name
FlixOne.BookStore.ProductService

Location
D:\101 Bitbucket\Hands-On-Microservices-with-C#-and-.NET-Core-3.0-Third-Edition\Chapter 05\...

Solution name
FlixOne.BookStore.ProductService

Place solution and project in the same directory
Create a new ASP.NET Core web application

.NET Core ➔ ASP.NET Core 3.1

API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

Web Application
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

Web Application (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

Angular
A project template for creating an ASP.NET Core application with Angular

React.js
A project template for creating an ASP.NET Core application with React.js

React.js and Redux
A project template for creating an ASP.NET Core application with React.js and Redux

Get additional project templates

Authentication
No Authentication
Change

Advanced
- Configure for HTTPS
- Enable Docker Support
  (Requires Docker Desktop)

Author: Microsoft
Source: .NET Core 3.1.0

Back ➔ Create
Chapter 6: Securing Microservices Using Azure Active Directory
1. Fetch the resource

4. Authorize me to fetch The resource (Authorize request)

5. You have been authorize to fetch the resource (Authorize agent)

2. Allow me to fetch resource

3. Take permission from resource owner

6. Here is authorization grant

7. Get the access token

8. Allow me to fetch resource, I have access token with me

9. Okay, here is the access token from client, provide the protected resource

10. This is protected resource

11. Provide the resource to end-user

I assume, you are authenticated i.e. trusted by end-user
Azure Active Directory

Free training from Microsoft
Secure your application by using OpenID Con... 6 units · 50 min

Useful links
Overview
Get Started
Pricing
Register an application

* Name
The user-facing display name for this application (this can be changed later).

FlikOne.BookStore.ProductService

Supported account types
Who can use this application or access this API?

- Accounts in this organizational directory only (Default Directory only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

Help me choose...

By proceeding, you agree to the Microsoft Platform Policies 📚

Register
Add a client secret

Description
AppKey

Expires
- In 1 year
- In 2 years
- Never

[Add] [Cancel]

Client secrets
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

<table>
<thead>
<tr>
<th>Description</th>
<th>Expires</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppKey</td>
<td>11/6/2020</td>
<td></td>
</tr>
</tbody>
</table>
Welcome

This is a landing page.
You have to sign-in to see Product Listing. Click here to sign-in. Before that make sure you have setup Authorization mechanism.
Chapter 7: Monitoring Microservices
Create storage account

Basics  Networking  Advanced  Tags  Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group *

Create new

Instance details

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. Choose classic deployment model

Storage account name *

Location *

Performance

Account kind

Review + create  < Previous  Next : Networking >
Create an Application Insights resource to monitor your live web application. With Application Insights, you have full observability into your application across all components and dependencies of your complex distributed architecture. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms including .NET, Node.js and Java EE, hosted on-premises, hybrid, or any public cloud. Learn More

PROJECT DETAILS
Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ○

Resource Group * ○

(INew) FlixOne

Create new

INSTANCE DETAILS
Name * ○

FlixOneWeb

Region * ○

(US) East US

Review + create
Gain insights through telemetry, analytics and smart detection

Detect
and diagnose exceptions and application performance issues

Monitor
websites on Azure, hosted containers, on-premises and with other cloud providers

Integrate
with your DevOps pipeline using Visual Studio, VSTS, GitHub, and web hooks

Get Started
Register your app with Application Insights

Account
Microsoft account

Subscription

Resource
FlixOneWeb (Existing resource)

Configure settings...

Pricing
Visit our pricing page for details.

Register
Application Insights

Register your app with Application Insights

Account
Microsoft account

Subscription

Resource
FlexOneWeb (Existing resource)

Configure settings...

Pricing
Visit our pricing page for details.

Register

Learn more...

Adding Application Insights to project...

App Map
Understand component dependencies

Smart Detection
Detects anomalies and alerts you

VS Search
Search and analyze your telemetry in VS

Analytics
Query terabytes of data in just seconds

CodeLens
View method performance inline with your code in VS

Live Metrics
See the effects of your web app in real time
Application Insights Configuration

Resource Settings

Sending telemetry to

FlixOneWeb in FlixOne

Reading telemetry from

FlixOneWeb as Gaurav Arora

Configured 100%

- SDK added
- App registered with Application Insights
```csharp
public class Startup
{
    public Startup(IConfiguration configuration)
    {
        Configuration = configuration;
    }

    public IConfiguration Configuration { get; }

    // This method gets called by the runtime. Use this method to add services to the container.
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddControllersWithViews();
        services.AddApplicationInsightsTelemetry();
    }
}
```
Average page load time breakdown:

- Page load network time: 4.5 ms
- Client processing time: 68.25 ms
- Send request time: 514.33 ms
- Receiving response time: 58.08 ms

Server response time:

- Server response time (avg): 63.04 ms

Failed requests:

- 0 failed requests

Completed. Showing results from the last 24 hours.

<table>
<thead>
<tr>
<th>timestamp (UTC)</th>
<th>id</th>
<th>name</th>
<th>url</th>
<th>success</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/18/2018, 2:33:54.02 PM</td>
<td>c5c32c3a9b4b9b8c5dbf74d7f677f12c1c55ebd08b4b</td>
<td>GET /lib/jquery/dist/jquery.min.js</td>
<td><a href="http://localhost:31540/lib/jquery/dist/jquery.min.js">http://localhost:31540/lib/jquery/dist/jquery.min.js</a></td>
<td>True</td>
</tr>
<tr>
<td>11/18/2018, 2:33:54.02 PM</td>
<td>t7b8b265b9b7734b793b5bce6e255b6b6b7f1b8b265b9b8c5dbf74d7f677f12c1c55ebd08b4b</td>
<td>GET /js/ide.js</td>
<td><a href="http://localhost:31540/js/ide.js">http://localhost:31540/js/ide.js</a></td>
<td>True</td>
</tr>
<tr>
<td>11/18/2018, 2:33:54.02 PM</td>
<td>d1d8b265b9b7734b793b5bce6e255b6b6b7f1b8b265b9b8c5dbf74d7f677f12c1c55ebd08b4b</td>
<td>GET /</td>
<td><a href="http://localhost:31540/">http://localhost:31540/</a></td>
<td>True</td>
</tr>
<tr>
<td>11/18/2018, 2:33:54.02 PM</td>
<td>113954221c273612c6d69ba9a4d72c2a5adeb9f1444</td>
<td>GET /lib/bootstrap/dist/css/bootstrap.min.css</td>
<td><a href="http://localhost:31540/lib/bootstrap/dist/css/bootstrap.min.css">http://localhost:31540/lib/bootstrap/dist/css/bootstrap.min.css</a></td>
<td>True</td>
</tr>
<tr>
<td>11/18/2018, 2:33:54.02 PM</td>
<td>23572a712d7a47b25710775745056d97430b904e82cbeb64c</td>
<td>GET /favicon.ico</td>
<td><a href="http://localhost:31540/favicon.ico">http://localhost:31540/favicon.ico</a></td>
<td>True</td>
</tr>
</tbody>
</table>
Chapter 8: Scaling Microservices with Azure

Scale Cube

Martin L. Abbott; Michael T. Fisher
Microservices infrastructure

Cluster orchestration tools (Apache Mesos, Docker Swarm, Kubernetes)

VM Extensions

VMs and ScaleSets

Azure stack

Azure public cloud
Chapter 9: Introduction to Reactive Microservices
<table>
<thead>
<tr>
<th>Event</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login</td>
<td>Auth/Login</td>
</tr>
<tr>
<td>UpdateProduct</td>
<td>Product/Update</td>
</tr>
</tbody>
</table>

UpdateProduct += \{BaseUrl\}/Product/Update
### Order

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Client ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1704AC</td>
<td>Cancel</td>
<td>#</td>
</tr>
<tr>
<td>1704ZA</td>
<td>Placed</td>
<td>#</td>
</tr>
<tr>
<td>....</td>
<td>....</td>
<td>....</td>
</tr>
</tbody>
</table>

### Details

<table>
<thead>
<tr>
<th>Order ID</th>
<th>Item ID</th>
<th>Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1704ZA</td>
<td>HP_lap_24</td>
<td>2</td>
</tr>
<tr>
<td>1704AC</td>
<td>....</td>
<td>....</td>
</tr>
</tbody>
</table>
Create a new ASP.NET Core web application

.NET Core

API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

Web Application
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

Web Application (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

Angular
A project template for creating an ASP.NET Core application with Angular

React.js
A project template for creating an ASP.NET Core application with React.js

React.js and Redux
A project template for creating an ASP.NET Core application with React.js and Redux

Get additional project templates

Authentication
No Authentication
Change

Advanced
Configure for HTTPS
Select Enable Docker Support
(Requires Docker Desktop)

Author: Microsoft
Source: .NET Core 3.1.0

Browse Installed Updates
System.Reactive.Core

NuGet Package: System.Reactive.Core
Version: 4.2.0-preview.625
Latest preview version 4.2.0-preview.625

Description
Reactive Extensions (Rx) for .NET v3 compatibility facade for

<table>
<thead>
<tr>
<th>Version</th>
<th>Latest preview</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.0-preview.625</td>
<td>Install</td>
</tr>
</tbody>
</table>

Options

Licenses
Apache 2.0

Date: Monday, August 19, 2019 (8/19/2019)
Project URL: https://github.com/dotnet/reactive
Tags:

System.Reactive
System.Reactive.PlatformServices

Dependencies
.NET Framework Version 4.6
System.Reactive (version 4.2.0-preview.625)
System.Threading.Tasks.Extensions (>= 4.5.3)
Chapter 10: Design Patterns and Best Practices
Chapter 11: Building a Microservice Application

![Diagram showing the components of a microservice application: User Interface, Module - Direct Sale, Module - Inventory, and Database. Arrows indicate connections between the components.]
Chapter 12: Microservices Architecture Summary
Graphics Bundle Ends Here