

🎓 Learning Roadmap للـ Production Ready ECommerce

Production هذا الملف يحتوي على جميع المصادر والمواضيع التي تحتاج تتعلمها بشأن تطلع المشروع للـ

📋 جدول المحتويات

1. Security & Secrets Management
2. Configuration Management
3. Docker & Containerization
4. Unit Testing
5. Integration Testing
6. Health Checks
7. CI/CD & GitHub Actions
8. Deployment
9. Database Best Practices
10. Monitoring & Logging

1. Security & Secrets Management

✂️ الأسئلة الأساسية التي تحتاج تفهمها:

- خطيرة جداً؟ hard-coded secrets فيه
- development و production secrets ما الفرق بين
- User Secrets و Environment Variables ما الفرق بين
- Azure Key Vault فيه

📖 المصادر الرسمية:

1.1 User Secrets في Development

- الموقع الرسمي: <https://learn.microsoft.com/en-us/aspnet/core/security/app-secrets>

- اقرأ أولاً: How to manage secrets in development
- جرب:

```
dotnet user-secrets init
dotnet user-secrets set "JwtSettings:Secret" "your-secret-key"
```

1.2 Configuration in **ASP.NET Core**

- الموقع الرسمي: <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/configuration/>
- اقرأ: Configuration providers
- اقرأ: How configuration providers work

1.3 Azure Key Vault Integration

- الموقع الرسمي: <https://learn.microsoft.com/en-us/azure/key-vault/general/overview>
- الموقع الرسمي: <https://learn.microsoft.com/en-us/aspnet/core/security/key-vault-configuration>
- المستوى: متقدم (لا تبدأ من هنا)

1.4 Environment Variables

- الموقع الرسمي: <https://learn.microsoft.com/en-us/dotnet/core/extensions/configuration-providers#environment-variable-configuration-provider>
- الفيديو: Environment Variables in .NET - https://www.youtube.com/watch?v=_YYEhfwLvQ

التطبيق العملي:

- (فقط development) في مشروعك User Secrets أنشئ
- وإليه اللي ما يروحش appsettings.json فهم إيه يروح في
- جرب Environment Variables في Docker

2. Configuration Management

الأسئلة الأساسية:

- appsettings.json و appsettings.Production.json ما الفرق بين
- بناءً على البيئة؟ configuration كيف تحمل
- configuration؟ للـ best practices إيه الـ

المصادر:

2.1 Environment-specific Configuration

- الموقع الرسمي: <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/configuration/#environment-variables>
- اقرأ: How **ASP.NET** Core loads configuration based on environment

2.2 Configuration Patterns

- الموقع الرسمي: <https://learn.microsoft.com/en-us/dotnet/core/extensions/configuration>
- اقرأ: Configuration patterns and practices

2.3 Options Pattern

- الموقع الرسمي: <https://learn.microsoft.com/en-us/dotnet/core/extensions/options>
- اقرأ: How to use IOptions pattern

التطبيق العملي:

- جديد appsettings.Production.json أنشئ
- في كودك استخدم IOptions
- User Secrets وإليه في appsettings.json فهم إليه اللي تحط في

3. Docker & Containerization

الأسئلة الأساسية:

- Docker؟ إليه
- Base Image و Runtime Image ما الفرق بين
- Multi-stage builds ما أهمية
- Dockerfile في best practices إليه الـ

المصادر:

3.1 Docker Fundamentals

- الموقع الرسمي: <https://docs.docker.com/get-started/>

- اقرأ: What is Docker?
- جرب: <https://www.katacoda.com/courses/docker/> (interactive labs)

3.2 Docker for .NET

- الموقع الرسمي: <https://learn.microsoft.com/en-us/dotnet/core/docker/building-net-docker-images>
- اقرأ: Building Docker images for .NET applications
- اقرأ: Best practices for writing Dockerfile

3.3 Multi-stage Builds

- الموقع الرسمي: <https://docs.docker.com/build/building/multi-stage/>
- الفيديو: Multi-stage Docker builds - <https://www.youtube.com/watch?v=Ctl2V7w75Ts>

3.4 .NET 9 on Docker

- الموقع الرسمي: https://hub.docker.com/_/microsoft-dotnet-aspnet
- اقرأ: Docker tags for .NET 9

3.5 Health Checks in Docker

- الموقع الرسمي: <https://docs.docker.com/engine/reference/builder/#healthcheck>
- اقرأ: HEALTHCHECK instruction

التطبيق العملي:

- جديد من الصفر Dockerfile اكتب
- استخدم multi-stage build
- build locally اختبر
- HEALTHCHECK أضيف

4. Unit Testing

الأسئلة الأساسية:

- Unit Tests؟ ليه
- Unit Test و Integration Test؟ ما الفرق بين
- Mocking؟ ليه
- testable code؟ كيف تكتب

المصادر:

4.1 xUnit Fundamentals

- الموقع الرسمي: <https://xunit.net/docs/getting-started/netcore>
- اقرأ: Getting started with [xUnit.net](https://xunit.net)
- اقرأ: Running tests from the command line

4.2 Unit Testing Best Practices

- الموقع الرسمي: <https://learn.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices-with-nunit>
- اقرأ: Best practices and naming conventions

4.3 Moq - Mocking Library

- الموقع الرسمي: <https://github.com/moq/moq4>
- الوثائق: <https://github.com/moq/moq4/wiki/Quickstart>
- اقرأ: How mocking works

4.4 Test Patterns

- الموقع الرسمي: <https://learn.microsoft.com/en-us/dotnet/core/testing/>
- اقرأ: AAA pattern (Arrange, Act, Assert)

4.5 Unit Testing **ASP.NET** Core

- الفيديو: Unit Testing in C# - <https://www.youtube.com/watch?v=H8w3NG6PGMc>
- المقالة: <https://learn.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices>

التطبيق العملي:

- جديد xUnit أنشئ مشروع: `dotnet new xunit -n ECommerce.Tests`
- اكتب Unit Tests للـ domain models
- اكتب Unit Tests للـ repository pattern
- استخدم Moq للـ mocking

5. Integration Testing

الأسئلة الأساسية:

- Integration testing و Unit ما الفرق بين
- Database interactions كيف تختبر
- API endpoints كيف تختبر

المصادر:

5.1 Integration Testing in **ASP.NET Core**

- الموقع الرسمي: <https://learn.microsoft.com/en-us/aspnet/core/test/integration-tests>
- اقرأ: Overview of integration testing

5.2 WebApplicationFactory

- الموقع الرسمي: <https://learn.microsoft.com/en-us/aspnet/core/test/integration-tests#introduction-to-integration-tests>
- اقرأ: Using WebApplicationFactory to test APIs

5.3 Testing Database Layer

- المقالة: <https://learn.microsoft.com/en-us/ef/core/testing/>
- اقرأ: Testing EF Core applications

5.4 Test Fixtures & Setup

- الفيديو: Integration Testing in .NET - <https://www.youtube.com/watch?v=7roqly9aCEY>

التطبيق العملي:

- أنشئ API endpoints لـ Integration Tests
- استخدم WebApplicationFactory
- اختبر Database interactions
- authentication و authorization اختبر

6. Health Checks

الأسئلة الأساسية:

- Health Checks؟ ليه
- readiness و liveness ما الفرق بين
- orchestration tools كيف يستخدمها الـ

المصادر:

6.1 Health Checks in **ASP.NET Core**

- الموقع الرسمي: <https://learn.microsoft.com/en-us/aspnet/core/host-and-deploy/health-checks>
- اقرأ: Kernal health checks
- اقرأ: Custom health check publishers

6.2 Health Checks with Dependencies

- الموقع الرسمي: <https://learn.microsoft.com/en-us/aspnet/core/host-and-deploy/health-checks#health-checks-with-a-ui>
- اقرأ: Health checks for databases, caches, etc.

6.3 Health Check UI

- الموقع الرسمي: <https://github.com/Xabaril/AspNetCore.Diagnostics.HealthChecks>
- المشروع: AspNetCore.Diagnostics.HealthChecks

التطبيق العملي:

- أضف MapHealthChecks في Program.cs
- أنشئ custom health check لـ database
- أنشئ custom health check لـ Redis

7. CI/CD & GitHub Actions

الأسئلة الأساسية:

- CI/CD؟ ما الـ

- ليه GitHub Actions؟
- كيف تبني و تختبر تلقائياً؟

المصادر:

7.1 GitHub Actions Basics

- الموقع الرسمي: <https://docs.github.com/en/actions>
- اقرأ: Introduction to GitHub Actions
- اقرأ: Workflow syntax

7.2 .NET with GitHub Actions

- الموقع الرسمي: <https://docs.github.com/en/actions/use-cases-and-examples/building-and-testing/building-and-testing-dotnet>
- اقرأ: Building and testing .NET
- اقرأ: Publishing packages with GitHub Actions

7.3 Docker with GitHub Actions

- الموقع الرسمي: <https://docs.github.com/en/actions/publishing-packages/publishing-docker-images>
- اقرأ: Publishing Docker images

7.4 Secrets in GitHub Actions

- الموقع الرسمي: <https://docs.github.com/en/actions/security-guides/encrypted-secrets>
- اقرأ: Using secrets in workflows

التطبيق العملي:

- أنشئ `.github/workflows/build.yml`
- أضف build step
- أضف test step
- أضف docker build step
- استخدم GitHub Secrets

8. Deployment

✂️ **Railway أو Azure ابدأ بـ) اختر واحد من الثلاثة:**

8.1 Azure App Service (الأسهل والأكثر احترافية)

المصادر:

- الموقع الرسمي: <https://learn.microsoft.com/en-us/azure/app-service/quickstart-dotnetcore>
- التوثيق الكاملة: <https://learn.microsoft.com/en-us/azure/app-service/>
- الفيديو: Deploy ASP.NET Core to Azure - <https://www.youtube.com/watch?v=3KfKxMFv0T8>
- اقرأ: Deployment best practices
- اقرأ: Scaling App Service

الخطوات:

1. أنشئ Azure Account (free tier متاح)
2. أنشئ Resource Group
3. أنشئ App Service Plan
4. أنشئ Web App
5. Deploy باستخدام Git أو Visual Studio

8.2 Railway (الأسهل والأسرع - مجاني للبداية)

المصادر:

- الموقع الرسمي: <https://railway.app/>
- التوثيق: <https://docs.railway.app/>
- اقرأ: Getting started with Railway
- اقرأ: Deploying .NET apps
- الفيديو: Deploy .NET to Railway - <https://www.youtube.com/watch?v=g8ov1BZk-cE>

الخطوات:

1. أنشئ Railway Account (GitHub login)
2. Connect GitHub repo
3. Deploy (drag & drop)
4. Set environment variables

8.3 AWS ECS (الأكثر قوة لكن معقد)

المصادر:

- الموقع الرسمي: <https://aws.amazon.com/ecs/>
- المقالة: <https://aws.amazon.com/blogs/developer/deploying-net-applications-on-aws/>
- التوثيق: <https://docs.aws.amazon.com/AmazonECS/latest/developerguide/>
- اقرأ: ECS Task Definitions
- اقرأ: ECS Services

التطبيق العملي:

اختر واحد:

- Deploy و Free tier Azure Account أنشئ
- Deploy و Railway اسجل في (!الأسهل)
- AWS Free tier (advanced features لو بتحتاج) استخدم

9. Database Best Practices

الأسئلة الأساسية:

- كيف تحمي البيانات؟
- backup strategy؟
- migrations؟ كيف تدير

المصادر:

9.1 SQL Server Best Practices

- الموقع الرسمي: <https://learn.microsoft.com/en-us/sql/relational-databases/tables/primary-and-foreign-key-constraints>
- اقرأ: Security best practices
- اقرأ: Backup and recovery

9.2 Entity Framework Core

- الموقع الرسمي: <https://learn.microsoft.com/en-us/ef/core/>
- اقرأ: Migrations
- اقرأ: Performance tuning

9.3 Connection Pooling

- الموقع الرسمي: <https://learn.microsoft.com/en-us/sql/connect/ado-net-connection-pooling>
- اقرأ: How connection pooling works

9.4 Backup & Recovery

- الموقع الرسمي: <https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/back-up-and-restore-of-sql-server-databases>
- اقرأ: Backup strategies



:التطبيق العملي

- في مشروعك Migrations فهم
- جرب Seeding data
- فهم Connection string options

10. Monitoring & Logging



:الأسئلة الأساسية

- structured logging؟ ليه
- و مكتبات أخرى؟ Serilog إيه الفرق بين
- Application في Production كيف تراقب الـ



:المصادر

10.1 Serilog Documentation

- الموقع الرسمي: <https://serilog.net/>
- التوثيق الكاملة: <https://github.com/serilog/serilog/wiki>
- اقرأ: Structured logging

- اقرأ: Sinks (Seq, Application Insights, etc.)

10.2 Seq Server

- الموقع الرسمي: <https://datalust.co/seq>
- اقرأ: Getting started with Seq

10.3 Application Insights

- الموقع الرسمي: <https://learn.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview>
- اقرأ: What is Application Insights?
- اقرأ: Integration with .NET

10.4 Performance Monitoring

- الموقع الرسمي: <https://learn.microsoft.com/en-us/dotnet/core/diagnostics/>
- اقرأ: Diagnostic tools



:التطبيق العملي

- في مشروعك Serilog فهم
- جرب Seq locally
- أنشئ custom Serilog sinks

خريطة التعلم الموصى بها

أسبوع 1:

- └ Security & Secrets (يوميين)
- └ Configuration Management (يوم)

أسبوع 2:

- └ Docker & Containerization (يوميين)
- └ Health Checks (نص يوم)
- └ ملخص وتطبيق عملي (نص يوم)

أسبوع 3-4:

- └ Unit Testing (أسبوع كامل)
- └ Integration Testing (أسبوع كامل)

أسبوع 5:

- └ CI/CD & GitHub Actions (أيام 3)
- └ Database Best Practices (يوميين)

أسبوع 6:

- └ Deployment (أيام 3)
- └ Monitoring & Logging (يوميين)
- └ تطبيق في مشروعك الفعلي

نصائح هامة أثناء التعلم

1. لا تتسرع: خذ وقتك مع كل موضوع
2. اقرأ وجرب: ما تقرأ فقط، لازم تجرب الكود
3. اكتب تعليقات: وثق فهمك في الكود
4. "ابحث عن الـ "ليه": مش مهم "كيف"، المهم "ليه"
5. احفظ المصادر: قد تحتاجها لاحقاً
6. عن ما تعلمته blog post ساهم في مجتمع: اكتب

مصادر إضافية عامة

- **Microsoft Learn** (مجاني): <https://learn.microsoft.com>
- **Stack Overflow**: <https://stackoverflow.com/questions/tagged/c%23>
- **GitHub (examples)**: <https://github.com/dotnet/samples>
- **YouTube Channels**:
 - Code with Tim: <https://www.youtube.com/@CodeWithTimCorey>
 - Microsoft Developer: <https://www.youtube.com/@MicrosoftDeveloper>
 - Raw Coding: <https://www.youtube.com/@RawCoding>

Checklist للـ Learning

- ☐ Security & Secrets
- ☐ Configuration Management
- ☐ Docker Basics
- ☐ Dockerfile Best Practices
- ☐ Unit Testing Fundamentals
- ☐ Moq & Mocking
- ☐ Integration Testing
- ☐ Health Checks
- ☐ GitHub Actions Basics
- ☐ .NET with GitHub Actions
- ☐ Choose & Learn Deployment Platform
- ☐ Serilog & Logging
- ☐ Database Best Practices
- ☐ تطبيق كل ما تعلمته في مشروعك

تاريخ التحديث: January 20, 2026

ملاحظة: هذا الملف حي ويمكن تحديثه مع الوقت. أضف ملاحظاتك واكتشافاتك هنا