

Learning Roadmap لـ Production Ready ECommerce

هذا الملف يحتوي على جميع المصادر والمواضيع اللي تحتاج تعلمها عشان تطلع المشروع لـ Production

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 اللي تحتاج تفهمها:

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

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=_YYEhfwLvfQ

التطبيق العملي:

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

2. Configuration Management

الأسلحة الأساسية:

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



: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 أنشئ appsettings.Production.json
- في كودك IOptions استخدم
- وابه في appsettings.json فهم إيه اللي تحط في User Secrets

3. Docker & Containerization



:الأسئلة الأساسية

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



: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=CtI2V7w75Ts>

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 اكتب Dockerfile
- استخدم multi-stage build
- اختبر build locally
- أضف HEALTHCHECK

4. Unit Testing

❖ الأسئلة الأساسية:

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

4.1 xUnit Fundamentals

- الموقع الرسمي: <https://xunit.net/docs/getting-started/netcore>
- اقرأ: Getting started with [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:

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

5. Integration Testing

❖ الأسئلة الأساسية:

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

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

6. Health Checks

❖ الأسئلة الأساسية:

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

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:

- ما هي CI/CD؟

- GitHub Actions؟ ليه
- 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

❖ ابدأ بـ Azure أو Railway: 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 Git أو Visual Studio باستخدام

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:

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

9. Database Best Practices

الأسئلة الأساسية:

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

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 فهم Migrations
- جرب Seeding data
- فهم Connection string options

10. Monitoring & Logging

الأسئلة الأساسية:

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

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

خريطة التعلم الموصى بها

1: أسبوع :

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

2: أسبوع :

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

4-3: أسبوع :

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

5: أسبوع :

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

6: أسبوع :

- └ Deployment (3 أيام)
- └ Monitoring & Logging (يومين)
- └ تطبيق في مشروعك الفعلى

نصائح هامة أثناء التعلم

1. لا تتسرع: خذ وقتاً 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 تعلنته 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 تعلمه في مشروعك

تاریخ التحديث: January 20, 2026

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