

這本《ASP.NET Core 9 Web API 開發實戰》書籍將以**實際案例引導學習**，適合 **有 C# 基礎但不熟悉 Web API 開發的讀者**。書中將從零開始，循序漸進地帶領讀者構建符合 **REST 精神** 的 API，並涵蓋 **資料庫存取、身份驗證、權限控管、測試、CI/CD 與容器化** 等主題，確保讀者學會開發可落地的 Web API。

ASP.NET Core 9 Web API 開發實戰 - 章節大綱

第一部分：基礎概念與專案準備

1. Web API 是什麼？為什麼需要 Web API？

- Web API 的概念與應用場景
- Web API vs. MVC vs. gRPC
- RESTful API 原則與最佳實踐
- HTTP 動詞與狀態碼

2. 環境設置與開發工具

- 介紹 .NET 9 SDK 與 Visual Studio / VS Code
- 版本管理與 LTS 策略
- 使用 Postman 或 Swagger 進行 API 測試
- 設置 SQL Server / PostgreSQL / SQLite / NoSQL 環境 (依書籍案例需求)

3. 專案架構與解決方案規劃

- Web API 專案的基本結構
 - 方案 (Solution) 與專案 (Project) 的分工
 - 資料夾與命名空間的規劃 (Domain、Application、Infrastructure、API)
 - 介紹 Clean Architecture、DDD (Domain-Driven Design) 與分層架構
-

第二部分：建立 Web API 服務

4. 建立第一個 Web API

- 使用 `dotnet new webapi` 指令建立專案
- 了解 `Program.cs`、`appsettings.json` 配置
- 介紹 Controller 與 Endpoint
- 建立 `GET /api/health` 進行基本測試

5. Controller、Service、Repository 設計

- Controller 負責接收請求與回應
- Service 層負責業務邏輯
- Repository 模式實作資料存取
- 使用 **Dependency Injection (DI)** 管理物件生命週期

6. 符合 REST 精神的 API 設計

- RESTful API 設計規範與最佳實踐
- 路由設計 (`[Route]`、`[HttpGet]`、`[HttpPost]`)

- 避免 "胖 Controller"，拆分適當的 Service
 - 設計 DTO (Data Transfer Object) 提升 API 穩定性
 - 錯誤處理與統一回應格式 (`ProblemDetails`, `ApiResult`)
-

第三部分：資料庫存取

7. 使用 Entity Framework Core 進行資料存取

- 設計 Database Model (`DbContext`、Entity)
- 使用 Code-First 與 Migration 建立資料庫
- 基本 CRUD 操作 (`Add`, `Update`, `Delete`, `Query`)
- 使用 `AsNoTracking` 與 `LazyLoading` 進行效能調優

8. 使用 Dapper 進行高效能查詢

- Dapper vs. EF Core
 - 撰寫 SQL 指令與查詢參數
 - 透過 `IDbConnection` 進行資料存取
 - 動態 SQL 查詢與物件映射
-

第四部分：身份驗證與權限管理

9. 身份驗證 (Authentication)

- JWT (JSON Web Token) 與 OAuth 2.0
- 使用 ASP.NET Core Identity
- 登入、登出、Token 生成與驗證
- 設定 Refresh Token 機制

10. 權限管理 (Authorization)

- 角色 (Roles) 與權限 (Claims-Based Authorization)
 - 使用 Policy-Based Authorization
 - API 限制存取 (`[Authorize]`、`[AllowAnonymous]`)
 - 根據權限篩選 API 回應內容
-

第五部分：API 安全性與最佳實踐

11. 保護 API 安全性

- 使用 `Rate Limiting` 防止濫用請求
- 使用 `CORS` 限制跨域請求
- 設定 Content Security Policy (CSP)
- 使用 HTTPS 與 HSTS 確保安全傳輸

12. 記錄與監控

- 使用 `Serilog` 進行記錄與日誌管理
- 記錄 API 訪問日誌與請求內容

- 監控 API 健康狀態 (`HealthChecks`)
 - 使用 `OpenTelemetry` 進行分散式追蹤
-

第六部分：測試與 CI/CD

13. 測試 Web API

- 單元測試 (`xUnit` / `MSTest`)
- 整合測試 (`TestServer`)
- 使用 `Mock` 進行依賴模擬 (`Moq`, `FakeItEasy`)

14. CI/CD 部署

- 使用 GitHub Actions / Azure DevOps 進行 CI/CD
 - 自動化測試與建置 (Build & Test)
 - 使用 Docker 容器化 Web API
 - 部署到雲端 (Azure / AWS / GCP)
-

第七部分：進階應用與最佳實踐

15. API 版本管理

- API 版本控制策略
- 使用 `Microsoft.AspNetCore.Mvc.Versioning`

16. GraphQL vs. REST API

- GraphQL 基本概念
- 何時選擇 GraphQL ?

17. 事件驅動架構與背景任務

- 使用 `Hangfire` 或 `BackgroundService`
- Web API 如何與 Event Bus (如 Kafka, RabbitMQ) 整合

18. Elasticsearch / Redis 快取

- 使用 `MemoryCache` 與 `DistributedCache`
 - 利用 Redis 進行 API 快取優化
 - Elasticsearch 快速搜尋應用
-

第八部分：專案實作

19. 完整專案案例

- 需求分析與系統規劃
- 建立 Web API 服務
- 身份驗證與權限控管
- API 安全性與測試

- CI/CD 自動化部署

20. 總結與未來展望

- Web API 發展趨勢
- API Gateway 與微服務架構的發展
- 如何進一步提升 Web API 的效能與安全性

結語

這本書的設計目標是讓讀者能夠 **從零開始**，實戰開發一個完整的 **ASP.NET Core 9 Web API 專案**，並學會 **從架構規劃、資料庫存取、身份驗證、權限管理、測試、CI/CD 到 API 最佳實踐**。讓讀者不僅能夠開發 API，還能 **讓 API 維運更高效、擴展更容易、安全性更高**。

這份大綱是否符合你的需求？如果有特定案例或技術細節希望深入探討，可以進一步調整章節內容！