



**.NET
Minimal
API**

**Теоретичні засади та
методологія розробки
вебзастосунків**

.NET Minimal API:
Теоретичні засади та методологія розробки вебзастосунків

Томка Ю.Я.

Міністерство освіти і науки України
Чернівецький національний університет імені Юрія
Федьковича

Ю.Я. Томка

.NET Minimal API:
Теоретичні засади та методологія розробки
вебзастосунків

Чернівці
Мехнодрук
2025

УДК 004.432.2

Т-56

Рекомендовано до друку вченою радою Навчально-наукового інституту фізико-технічних та комп'ютерних наук Чернівецького національного університету імені Юрія Федьковича (протокол № 17 від 18 грудня 2025)

Рецензенти:

Литвин В.В., доктор технічних наук, професор, професор кафедри інформаційних систем і мереж Національного університету «Львівська політехніка»

Газдюк К.П., доктор філософії (PhD), доцент, завідувач кафедри програмного забезпечення комп'ютерних систем Чернівецького національного університету імені Юрія Федьковича

Томка Ю.Я.

Т-56 .NET minimal API. Теоретичні засади та методологія розробки веб-застосунків / Ю.Я. Томка – Чернівці: Технодрук, 2025. – 256с.

Посібник присвячений вивченню принципів створення вебзастосунків на базі Minimal API у .NET. У ньому розглядаються основи побудови REST-сервісів, робота з маршрутизацією, HTTP-методами, залежностями, прив'язуванням параметрів та middleware. Значна увага приділена взаємодії з базами даних, використанню Entity Framework Core, асинхронності, кешуванню, оптимізації продуктивності та питанням безпеки.

Окремі розділи охоплюють тестування, сумісність із новими версіями .NET та розгортання застосунків у Azure, Docker і на Kestrel. Видання призначене для студентів спеціальності ФЗ “Комп’ютерні науки” та всіх, хто бажає опанувати сучасні підходи до розроблення Minimal API.

УДК 004.432.2

©Чернів. нац. ун-т, 2025

©ПВКФ «Технодрук», 2025

©Томка Ю.Я., 2025

Зміст

Частина 1. Вступ до Minimal Apis	8
1. Початок роботи із Minimal API.....	9
1.1. Розуміння minimal APIs.....	10
1.2. Порівняння мінімальних API з традиційними підходами API.....	11
1.3. Значення мінімальних API у сучасній розробці.....	14
1.4. Встановлення необхідних інструментів та залежностей..	15
1.5. Встановлення Visual Studio для Windows	17
1.6. Налаштування середовища розробки	18
1.7. Створення проекту у Visual Studio.....	19
Питання для самоконтролю	21
2. Створення вашого першого Minimal API.....	23
2.1. Структура та організація проекту	23
2.2. Визначення кінцевих точок (endpoint-ів) і маршрутів.....	26
2.3. Створення API.....	31
Creating the first endpoint.....	33
2.4. Обробка HTTP запитів	36
Питання для самоконтролю	38
3. Анатомія Minimal API	40
3.1 Анатомія Minimal API.....	40
3.2. Компоненти застосунку з Minimal API	42
3.2. Життєвий цикл API-запиту	44
Питання для самоконтролю	46
Частина 2. Потік даних та виконання	48
4.1. Handling HTTP Methods and Routing.....	49
4.2. Обробка запитів	49
4.3. Визначення endpoint-ів у Todo API	50
4.4. Керування параметрами маршруту.....	56
4.5. Валідація запитів і обробка помилок	59
Мануальна валідація	59
Валідація за допомогою анотацій даних та прив'язки моделей	60
Валідація за допомогою фільтрів	64
Підсумок	65

Питання для самоконтролю	66
5 Конвеєр Middleware.....	68
5.1. Вступ до посередників (middleware).....	68
Просування через middleware	69
5.2. Налаштування middleware-конвеєра	71
Класи проміжного програмного забезпечення (Middleware classes).....	72
Вбудоване (inline) проміжне програмне забезпечення (middleware).....	73
Стандартне middleware	74
5.3. Створення кастомного middleware	75
Термінальний middleware	78
Обробка помилок у межах middleware-конвеєра	82
Підсумок	84
Питання для самоконтролю	85
6 Прив'язування параметрів.....	87
6.1. Джерела прив'язування параметрів	87
Значення маршруту	88
Рядки запиту (Query strings)	90
Перевірка наявності параметрів рядка запиту.....	90
Необов'язкові параметри рядка запиту.....	90
Заголовки	92
Строго типізоване зв'язування об'єктів	93
Значення форм.....	94
Явне зв'язування за допомогою атрибутів.....	95
Зв'язування параметрів через впровадження залежностей (Dependency Injection).....	96
6.2. Порядок пріоритетності прив'язки (Binding precedence).98	
6.3. Створення власної логіки прив'язки (Custom Binding Logic).....	99
Підсумок	102
Питання для самоконтролю	102
6. Ін'єкція залежностей у Minimal APIs.....	105
7.1. Розуміння DI.....	106
Контейнер DI	106

Чому варто використовувати DI	107
7.2. Налаштування DI в Minimal APIs.....	108
Налаштування проєкту з DI зі scoped-залежностями	110
Створення проєкту з singleton-DI.....	115
7.3. DI найкращі практики.....	118
Уникайте патерна Service Locator	119
Реєстрація сервісів за допомогою методів-розширень	120
Вибір коректного часу життя сервісу	120
Підсумок	121
Питання для самоконтролю	121
8. Інтеграція мінімального API з джерелами даних	124
8.1. Розуміння інтеграції даних у minimal API	125
8.2. Підключення та інтеграція з SQL-базами даних	127
Налаштування підключення до бази даних та отримання записів	127
Додавання записів про співробітників	131
Виконання транзакцій з API-ендпоінтів	132
8.3. Підключення до MongoDB.....	133
Підсумок	141
Питання для самоконтролю	142
9. Об'єктно-реляційне відображення з Entity Framework Core та Dapper.....	144
9.1. Введення в ORM	145
Налаштування Dapper у проєктах мінімального API	146
Виконання CRUD-операцій з Dapper	148
Налаштування Entity Framework у проєктах з мінімальними API.....	154
9.2. Виконання CRUD-операцій з Entity Framework	159
Підсумок	162
Питання для самоконтролю	163
Частина 3. Оптиміальний мінімальний API.....	165
10. Профілювання та ідентифікація вузьких місць	166
10.1. Введення в профілювання та моніторинг продуктивності	166
Знайомство з профайлером.....	167

Метрики продуктивності	168
10.2. Інструменти та техніки профілювання.....	169
Профілювання в Visual Studio.....	170
Бенчмаркінг з BenchmarkDotNet.....	174
10.3. Загальні вузькі місця продуктивності.....	179
Підсумок	181
Питання для самоконтролю	182
11. Використання асинхронного програмування для масштабованості	184
11.1. Розуміння та впровадження асинхронних патернів у мінімальному API	185
11.2. ТАР з async/await.....	187
11.3. Патерн асинхронної обробки (Asynchronous processing pattern)	188
11.4. Типові помилки та виклики	191
Підсумок	194
Питання для самоконтролю	195
12. Стратегії кешування для підвищення продуктивності.....	197
12.1. Введення в кешування в мінімальних API.....	198
12.2. Техніки кешування в пам'яті	199
12.3. Стратегії розподіленого кешування	203
12.4. Кешування відповідей	205
Підсумок	206
Питання для самоконтролю	207
Частина 4.....	209
Найкращі практики, дизайн,	209
і розгортання	209
13. Найкращі практики для стійкості Minimal API	210
13.1. Організація та структура коду.....	211
13.2. Дослідження структур папок	212
Модульна структура на основі функцій	212
Шарована модульна структура	213
13.3. Шаблони проектування (Design patterns).....	215
Шаблон фабрики (The factory pattern)	215
Шаблон репозиторію (The repository pattern).....	217

Шаблон стратегії (Strategy Pattern)	218
13.4. Обробка помилок.....	222
13.5. Міркування щодо безпеки.....	224
Аутентифікація	225
Авторизація	225
Обмеження швидкості	228
Підсумок	229
Питання для самоконтролю	230
14. Модульне тестування, сумісність та розгортання Minimal APIs	233
Технічні вимоги.....	Опшибка! Закладка не определена.
14.1. Модульне тестування та інтеграційне тестування для minimal APIs.....	234
14.2. Сумісність та міграція minimal APIs до .NET 9.....	241
14.3. Розгортання мінімальних API	244
Розгортання в Microsoft Azure App Service (хмарне розгортання)	245
Розгортання в контейнер Docker.....	248
Розгортання на місці з Kestrel.....	251
Підсумок	252
Питання для самоконтролю	253

Частина 1. Вступ до Minimal Apis

Навчальне видання

Томка Юрій Ярославович

.NET Minimal API:
Теоретичні засади та методологія розробки
вебзастосунків

Літературний редактор: О. В. Лукул

Папір офсетний. Формат 60х84/16.
Умов. друк. арк. 31,5. Обл.- вид. арк. 29,3. Тираж – 50.

Видавець та виготівник: ПВКФ «Технодрук»
Свідоцтво суб'єкта видавничої справи ДК №1841 від 10.06.2004 р.
58000, м. Чернівці, вул. І. Франка, 20, оф.18, тел. (0372) 55-05-85