<https://www.tutorialsteacher.com/mvc/mvc-model>

<https://dir.by/developer/mvc/render_body/>

https://www.codeguru.com/csharp/razor-pages-using-c/

**Архітектура ASP.NET MVC**

**ASP.NET** – технологія створення вебзастосунків і вебсервісів від компанії Майкрософт. Вона є складовою частиною платформи Microsoft.NET і розвитком старішої технології Microsoft ASP. На цей час останньою версією цієї технології є ASP.NET Core 6.0

Архітектурний шаблон MVC існує в розробці програмного забезпечення протягом тривалого часу. Майже всі мови використовують MVC з невеликими варіаціями, але концептуально він залишається незмінним.

Давайте розберемося з архітектурою MVC, яка підтримується в ASP.NET.

MVC означає Model, View і Controller. MVC розділяє програму на три компоненти – модель, представлення та контролер.

**Модель**: модель представляє форму даних. Клас у C# використовується для опису моделі. Об’єкти моделі зберігають дані, отримані з бази даних.

**Модель представляє дані**.

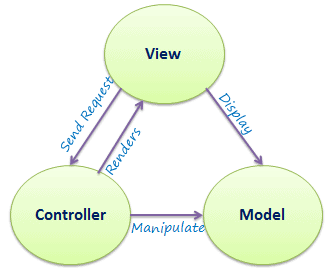
**Представлення:** Представлення у MVC – це інтерфейс користувача. Представляє відображення даних моделі для користувача, а також дозволяє їм змінювати їх. Представлення у ASP.NET MVC — це HTML, CSS і деякий спеціальний синтаксис (синтаксис Razor), який полегшує взаємодію з моделлю та контролером.

**Представлення — це інтерфейс користувача.**

**Контролер**: контролер обробляє запит користувача. Як правило, користувач використовує представлення та створює HTTP-запит, який оброблятиметься контролером. Контролер обробляє запит і повертає відповідне представлення як відповідь.

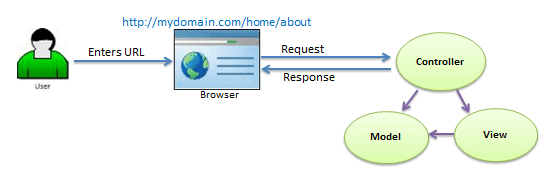
**Контролер — це обробник запитів.**

Наступний малюнок ілюструє взаємодію між моделлю, представленням і контролером.

[](https://www.tutorialsteacher.com/Content/images/mvc/mvc-architecture.png)

Архітектура MVC

Наступний малюнок ілюструє потік запиту користувача в ASP.NET MVC.

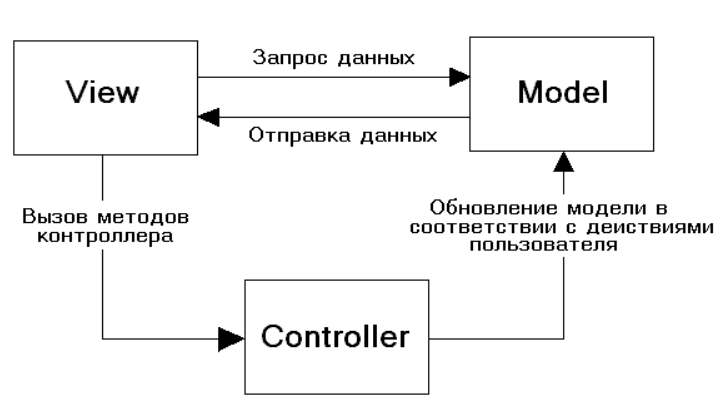
[](https://www.tutorialsteacher.com/Content/images/mvc/request-handling-in-mvc.png)

Потік запитів в архітектурі MVC

Згідно з наведеним вище малюнком, коли користувач вводить URL-адресу в браузері, вона переходить на веб-сервер і направляється до контролера. Контролер виконує пов’язане представлення та моделі для цього запиту, створює відповідь і надсилає її назад у браузер.

Шаблон MVC описує простий спосіб побудови структури програми, метою якого є відокремлення бізнес-логіки від інтерфейсу користувача. В результаті додаток легше масштабується, тестується, супроводжується і звичайно ж реалізується.

В архітектурі MVC модель надає дані та правила бізнес-логіки, подання відповідає за інтерфейс користувача, а контролер забезпечує взаємодію між моделлю та поданням.



Типову послідовність роботи MVC-програми можна описати так:

1. Коли користувач заходить на веб-ресурс, скрипт ініціалізації створює екземпляр програми та запускає його на виконання. При цьому відображається вигляд, наприклад, головної сторінки сайту.
2. Програма отримує запит від користувача та визначає запитаний контролер та дію. У випадку головної сторінки виконується дія за замовчуванням (index).
3. Додаток створює екземпляр контролера та запускає метод дії, в якій, наприклад, містяться виклики моделі, які зчитують інформацію з бази даних.
4. Після цього дія формує подання з даними, отриманими з моделі і виводить результат користувачеві.

Модель містить бізнес-логіку програми і включає методи вибірки (це можуть бути методи ORM), обробки (наприклад, правила валідації) і надання конкретних даних, що часто робить її дуже товстою, що цілком нормально.

Модель не має безпосередньо взаємодіяти з користувачем. Усі змінні, що стосуються запиту користувача, повинні оброблятися в контролері.

Модель не повинна генерувати HTML або інший код відображення, який може змінюватись залежно від потреб користувача. Такий код має оброблятися у представленні.

Одна й та сама модель, наприклад: модель аутентифікації користувачів може використовуватися як у користувальницькій, і у адміністративній частині додатку. У такому випадку можна винести загальний код в окремий клас і успадковуватись від нього, визначаючи в спадкоємцях специфічні для додатків методи.

Вигляд — використовується для визначення зовнішнього відображення даних, отриманих з контролера та моделі.

Види містять HTML-розмітку та невеликі вставки PHP-коду для обходу, форматування та відображення даних.

Не повинні безпосередньо звертатися до бази даних. Цим мають займатися моделі.

Не повинні працювати з даними, отриманими із запиту користувача. Це завдання має виконувати контролер.

Може безпосередньо звертатися до властивостей та методів контролера або моделей, щоб отримати готові до виведення дані.

Види зазвичай поділяють на загальний шаблон, що містить розмітку, загальну для всіх сторінок (наприклад, шапку і підвал) та частини шаблону, які використовують для відображення даних, що виводяться з моделі або відображення форм введення даних.

Контролер — зв'язуюча ланка, що з'єднує моделі, представлення та інші компоненти робочого додатку. Контролер відповідає за обробку запитів користувача. Контролер не повинен містити SQL-запитів. Їх краще тримати у моделях. Контролер не повинен містити HTML та іншу розмітку. Її варто виносити у види.

У добре спроектованому MVC-додатку контролери зазвичай дуже тонкі і містять лише кілька десятків рядків коду. Чого не скажеш про Stupid Fat Controllers (SFC) у CMS Joomla. Логіка контролера досить типова і більшість її виноситься в базові класи.

Моделі, навпаки, дуже товсті та містять більшу частину коду, пов'язану з обробкою даних, так як структура даних та бізнес-логіка, що міститься в них, зазвичай досить специфічна для конкретної програми.

# Історія версій ASP.NET MVC

ASP.NET — це безкоштовна веб-платформа для створення веб-сайтів і веб-додатків на .NET Framework за допомогою HTML, CSS і JavaScript. ASP.NET MVC 5 — це веб-фреймворк на основі архітектури Model-View-Controller (MVC).

Microsoft представила ASP.NET MVC у .NET 3.5, з того часу було додано багато нових функцій.

У квітні 2009 року корпорація Майкрософт зробила фреймворк ASP.NET MVC відкритим вихідним кодом. Вихідний код було випущено за публічною ліцензією Microsoft (MS-PL). Ви можете переглядати, завантажувати або робити внесок у вихідний код у [репозиторії github для AspNetWebStack](https://github.com/aspnet/AspNetWebStack) .

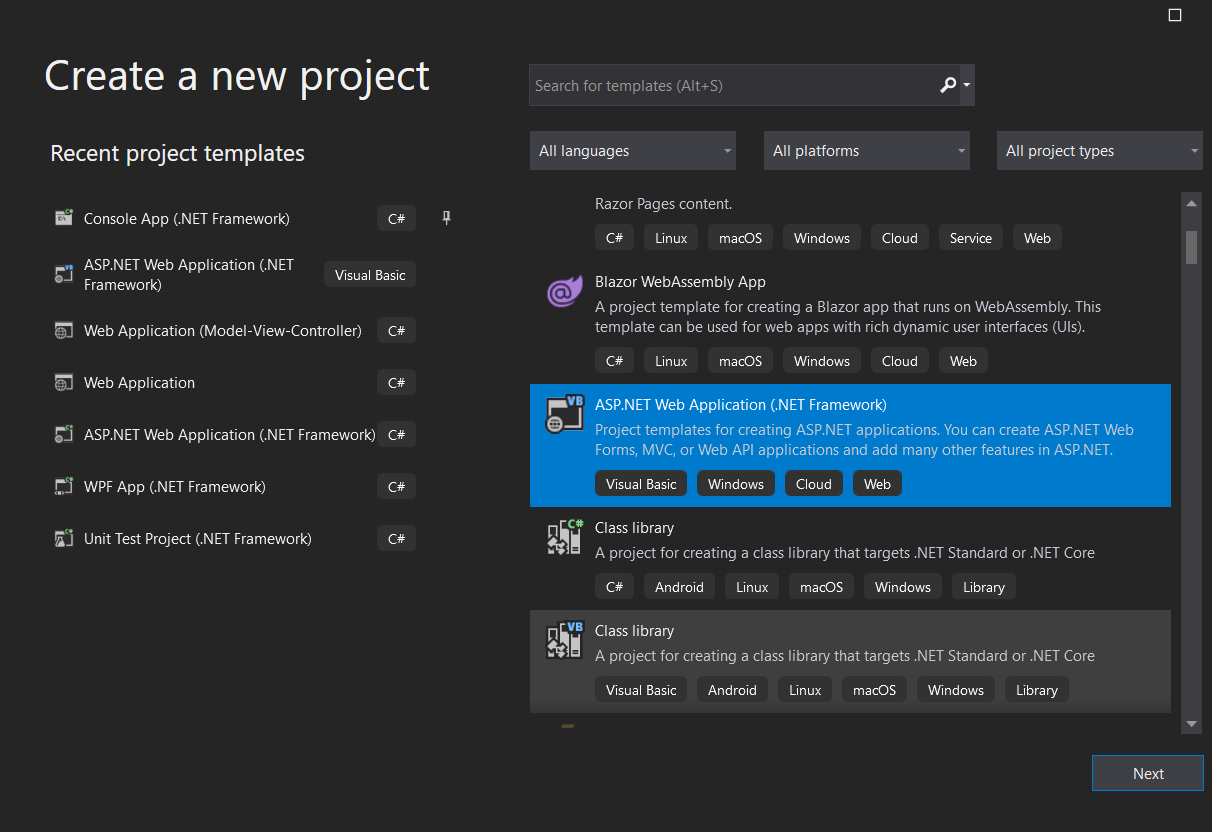
ASP.NET MVC framework можна встановити як пакет NuGet. Відвідайте [nuget.org](https://www.nuget.org/packages/Microsoft.AspNet.Mvc#versions-body-tab) , щоб дізнатися подробиці про кожну версію.

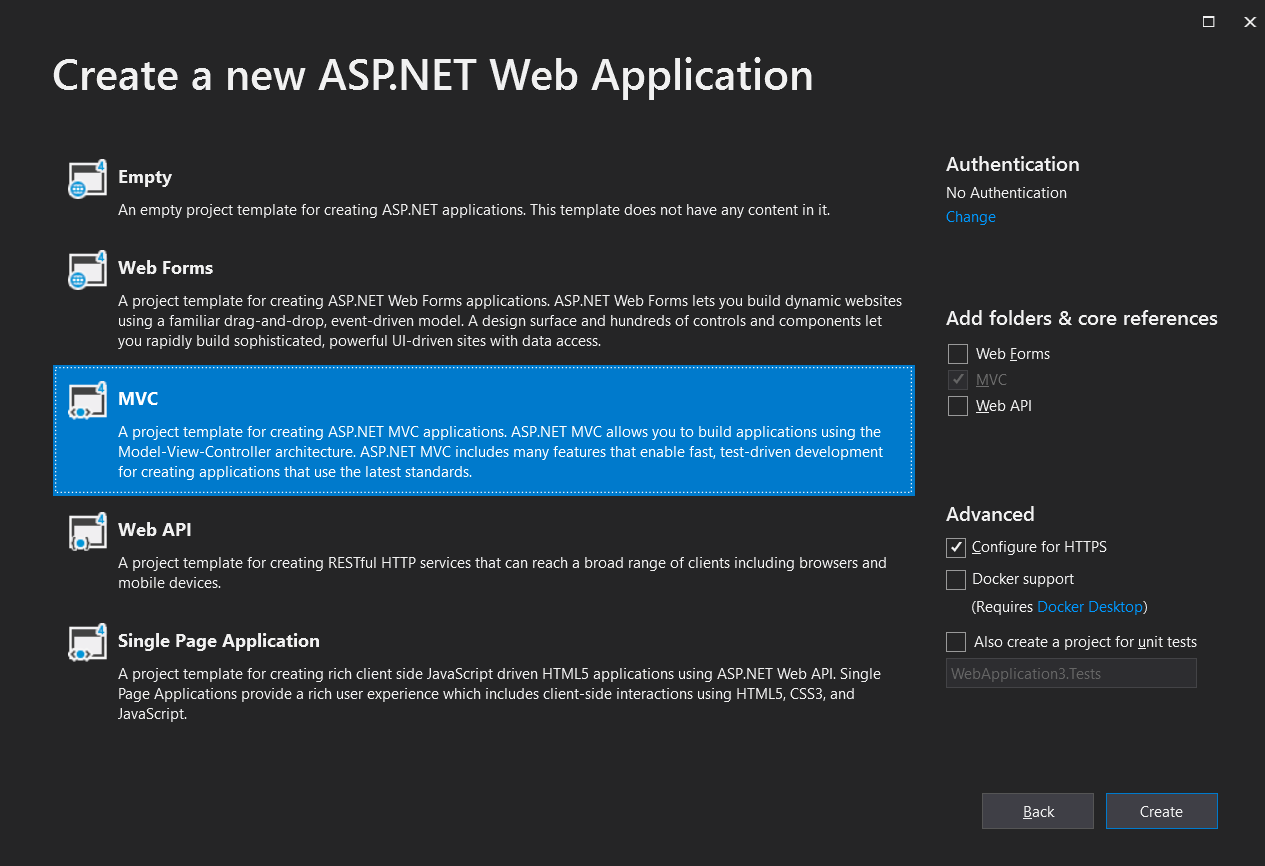
У Visual Studio остання версія версії ASP.NET MVC буде автоматично додана як пакет NuGet, коли ви створюєте новий проект ASP.NET MVC.

# Створення додатку ASP.NET MVC

Створимо нову веб-програму MVC за допомогою Visual Studio та розберемо основні будівельні блоки програми ASP.NET MVC.

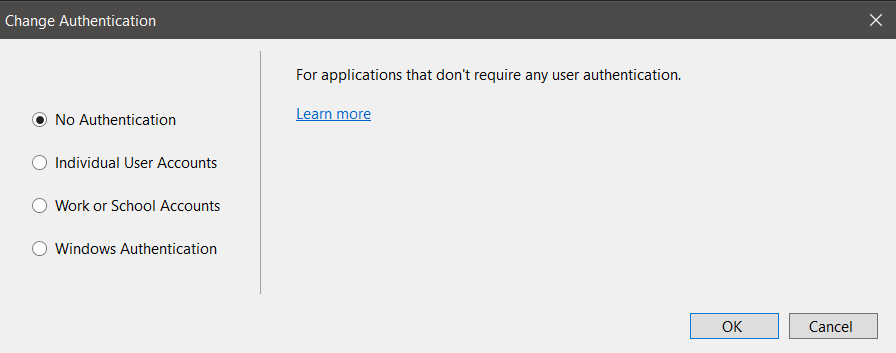
При створенні проекту виберемо ASP.NetWeb Application (.Net Framework)



У діалоговому вікні **Create** **New ASP.NET Web Application** виберіть MVC (якщо ще не вибрано), як показано нижче.

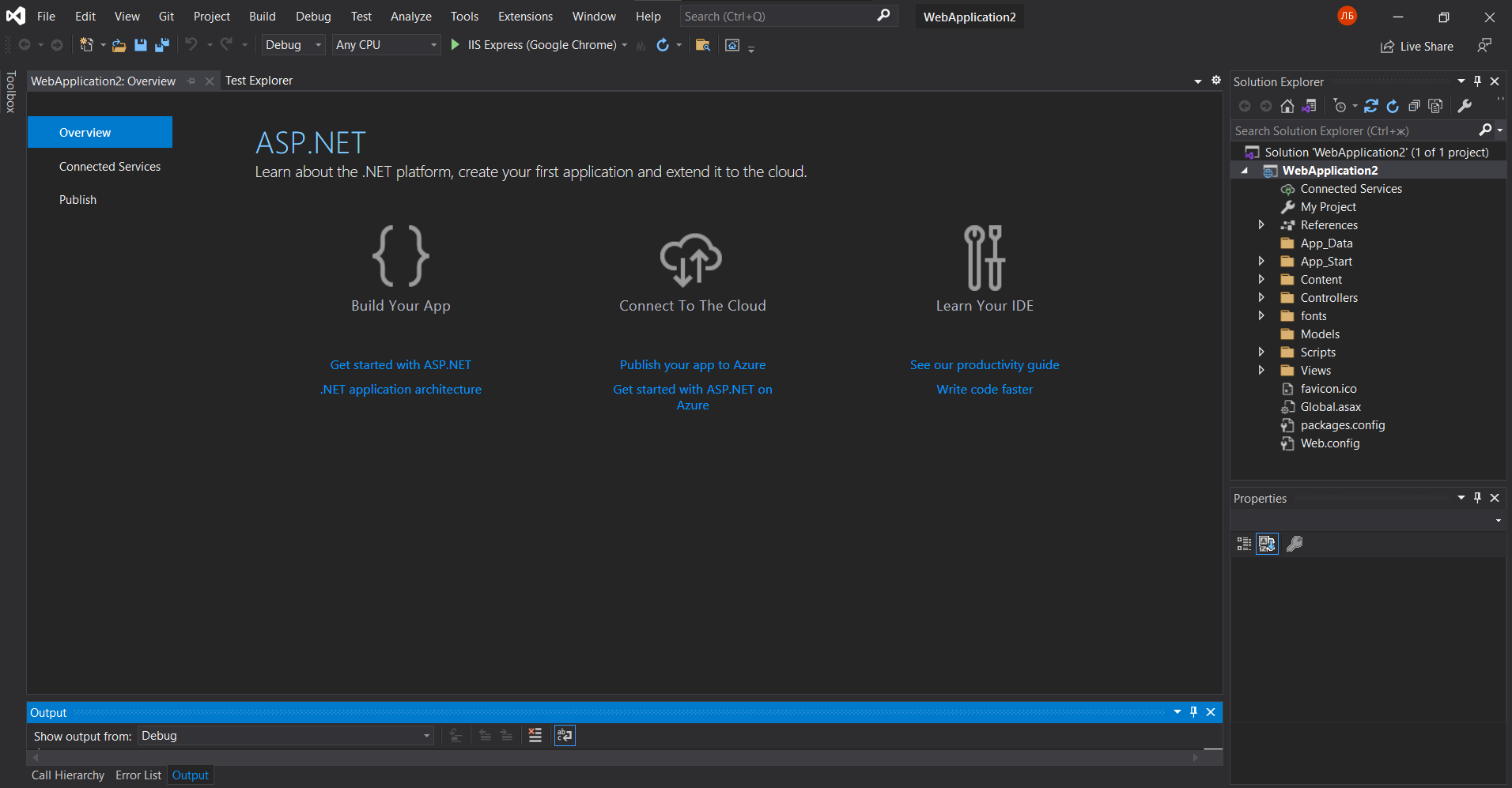
Виберіть шаблон проекту MVC

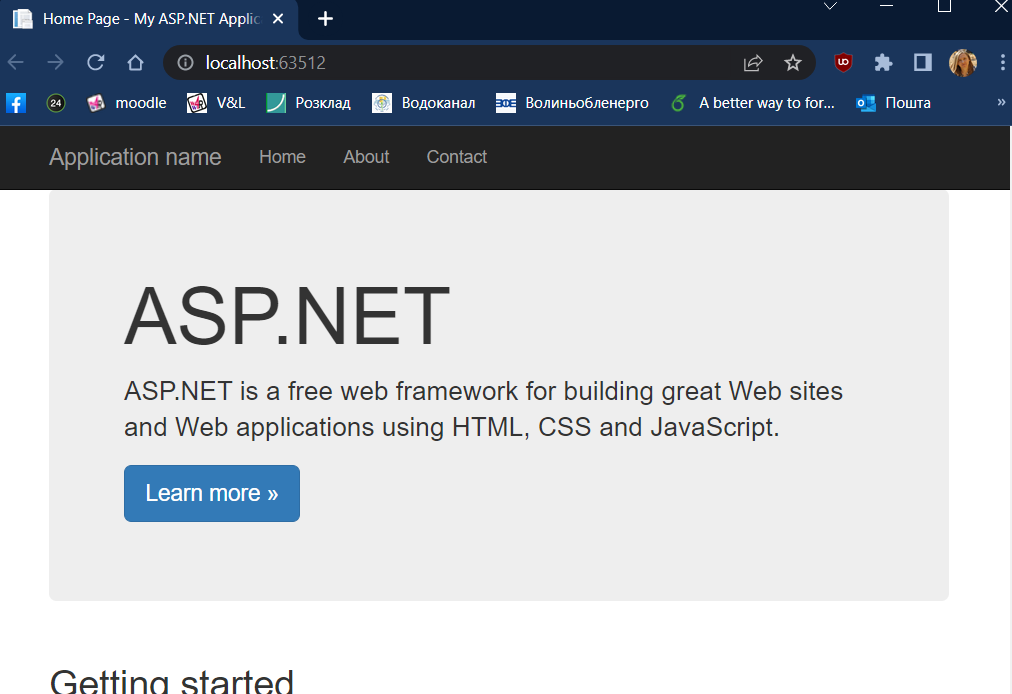
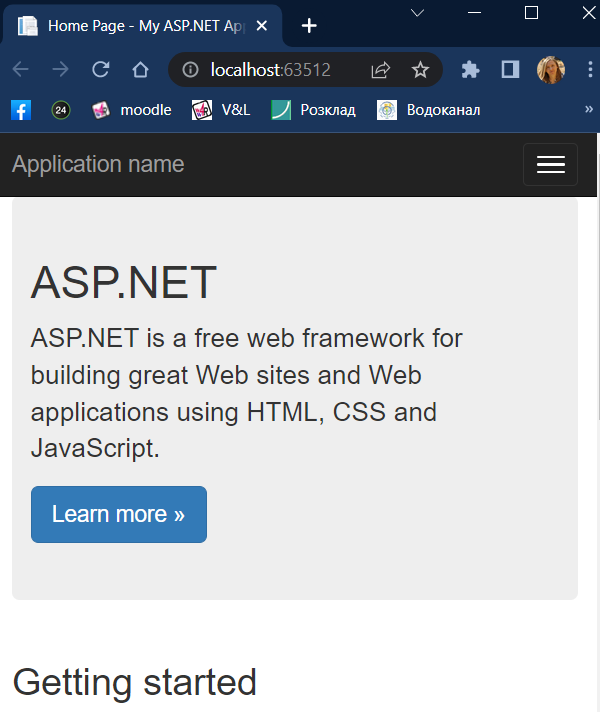
Ви також можете змінити автентифікацію, натиснувши кнопку **Authentifacation**. Ви можете вибрати відповідний режим автентифікації для своєї програми.



Тут ми зберігаємо автентифікацію за замовчуванням для нашої програми, яка не автентифікована. Натисніть **OK** , щоб продовжити.

Зачекайте деякий час, поки Visual Studio створить простий проект MVC за допомогою шаблону за замовчуванням, як показано нижче.



Проект MVC 5 за замовчуванням включає файли JavaScript і CSS початкової версії 3.0. Тож ви можете створювати адаптивні веб-сторінки. Цей адаптивний інтерфейс користувача змінюватиме свій зовнішній вигляд залежно від розміру екрана різних пристроїв. Наприклад, верхня панель меню буде змінена на мобільних пристроях, як показано на рис

**Створення проекту**

Запустіть Visual Studio та виберіть Створити новий проект .

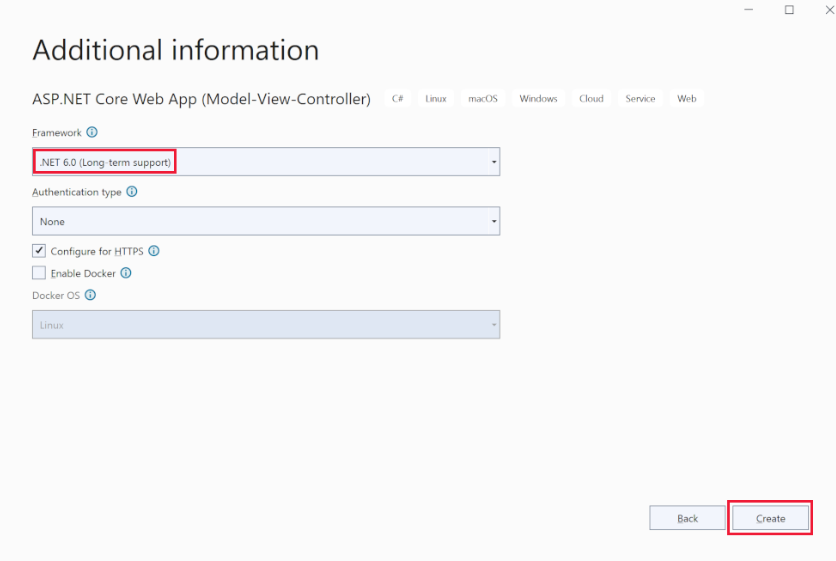
У діалоговому вікні « Створити новий проект » виберіть ASP.NET Core Web App (Model-View-Controller) > Next .

У діалоговому вікні « Налаштувати новий проект » введіть MvcMovieім’я проекту . Важливо назвати проект MvcMovie . Великі літери повинні відповідати кожному namespace, коли код копіюється.

Виберіть Далі .

У діалоговому вікні « Додаткова інформація » виберіть .NET 6.0 (довгострокова підтримка) .

Виберіть Створити.



Visual Studio використовує шаблон проекту за промовчанням для створеного проекту MVC. Створений проект це:

робочий додаток;

Простий початковий проект.

В адресному рядку вказується localhost:<port#>, а не будь-який тип example.com. Стандартне ім'я для локального комп'ютера — localhost. Коли Visual Studio створює веб-проект, для веб-сервера використовується випадковий порт.

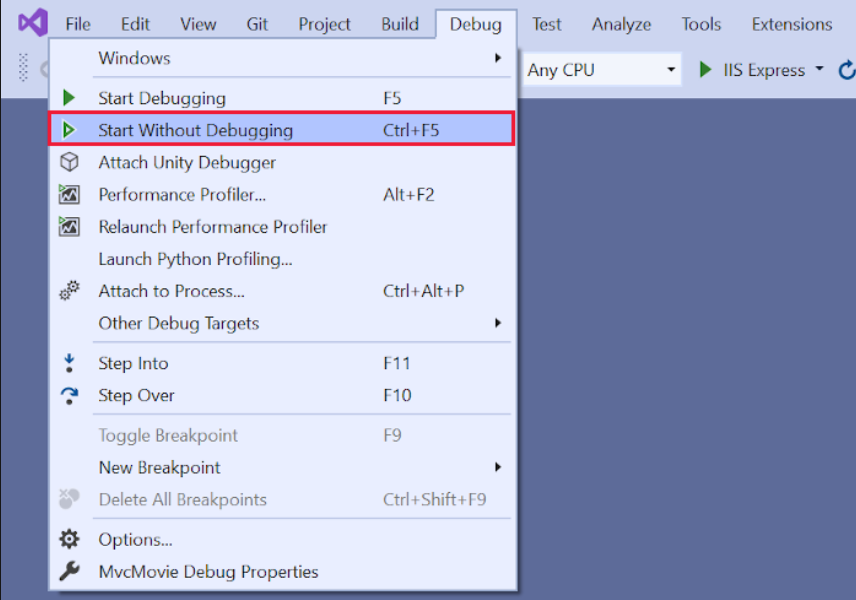
Запуск програми без налагодження шляхом натискання клавіш CTRL+F5 дозволяє:

Внесіть зміни до коду.

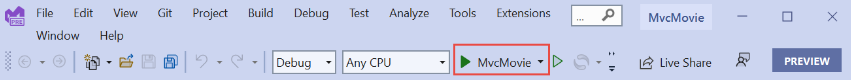
Збережіть файл.

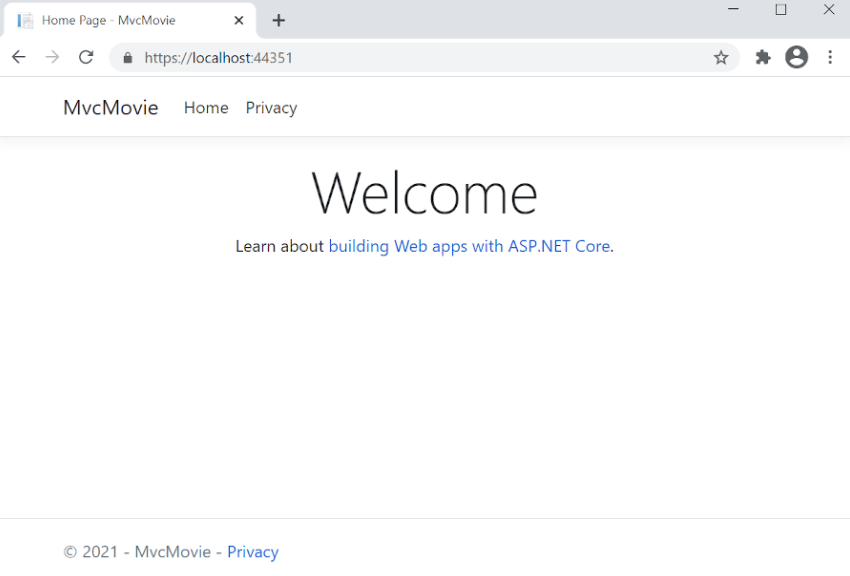
Швидко оновити браузер і переглянути зміни коду.

З меню Налагодження можна запустити програму з налагодженням або без:



Ви можете виконати налагодження програми, натиснувши кнопку **MvcMovie** на панелі інструментів:





**Додавання контролера до програми MVC ASP.NET Core**

Модель архітектури MVC поділяє додаток на три основні компоненти: **M** odel (модель), **V** iew (подання) та **C** ontroller (контролер). За допомогою моделі MVC можна створювати програми, які зручніше тестувати та оновлювати порівняно з традиційними монолітними програмами.

Програми на основі моделі MVC містять таке:

* **Моделі** : класи, що представляють дані у додатку. Класи моделі використовують логіку перевірки, що дозволяє застосовувати бізнес-правила до цих даних. Як правило, об'єкти моделі витягують та зберігають стан моделі у базі даних. У цьому посібнику модель Movie отримує відомості про фільми з бази даних і передає їх у подання або оновлює. Оновлені дані записуються до бази даних.
* **Уявлення** : компоненти, які формують інтерфейс програми користувача. Як правило, в інтерфейсі користувача відображаються дані моделі.
* **Контролери** : класи, які:
  + обробляють запити браузера;
  + одержують дані моделі;
  + викликають шаблони представлення дзвінків, які повертають відповідь.

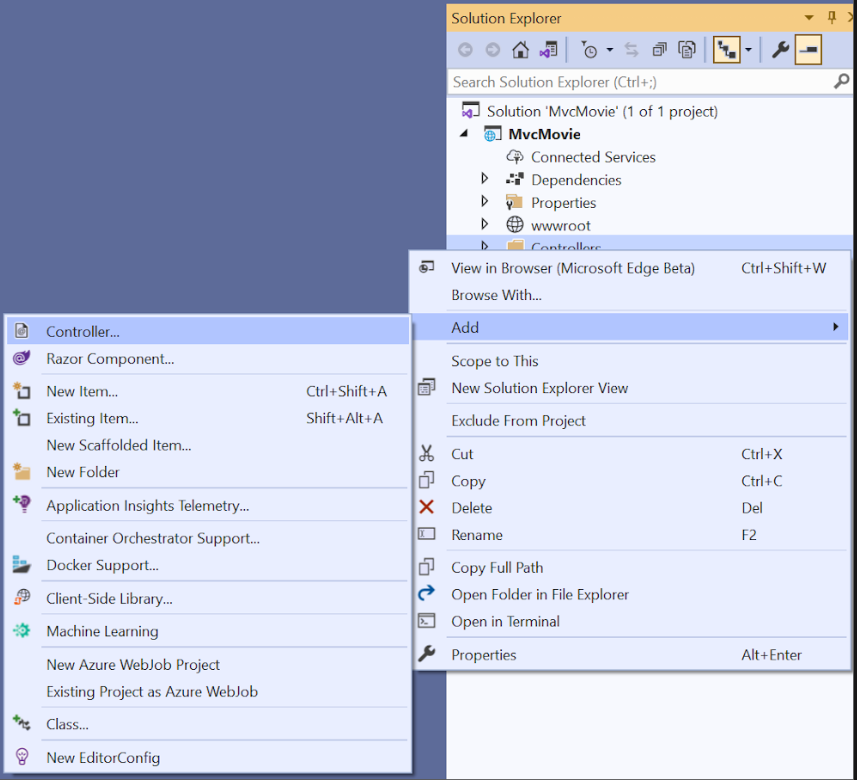
У програмі MVC представлення використовується лише для відображення інформації. Контролер обробляє та реагує на введення та взаємодію користувача. Наприклад, контролер обробляє сегменти URL-адреси та значення рядка запиту і передає ці значення моделі. Модель може використовувати ці значення для виконання запитів до бази даних. приклад.

* https://localhost:5001/Home/Privacy: задає контролер Home та дію Privacy.
* https://localhost:5001/Movies/Edit/5: це запит на зміну фільму ID=5 за допомогою контролера Movies та дії Edit, які докладно описані далі в цьому посібнику.

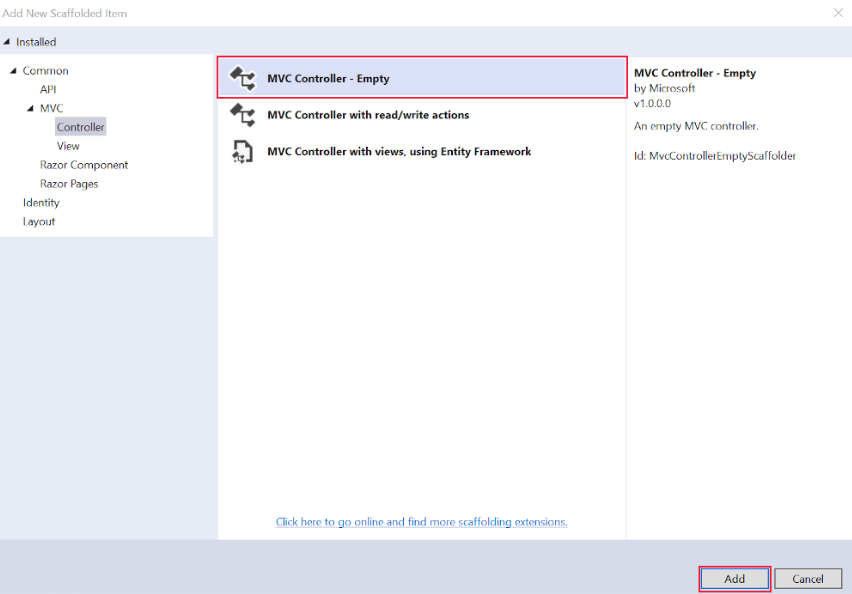
Дані маршруту описані далі у цьому посібнику.

Структура архітектури MVC поділяє додаток на три основні групи компонентів: моделі, уявлення та контролери. Цей шаблон допомагає реалізувати принципи поділу завдань: логіка інтерфейсу користувача відноситься до уявлення. Логіка введення відноситься до контролера. Бізнес-логіка розміщується у моделі. Подібний поділ дозволяє ефективно справлятися з труднощами при побудові програми, оскільки він дає можливість працювати одночасно з одним аспектом реалізації, не торкаючись при цьому коду інших. Наприклад, ви можете змінювати код подання незалежно від бізнес-логіки.

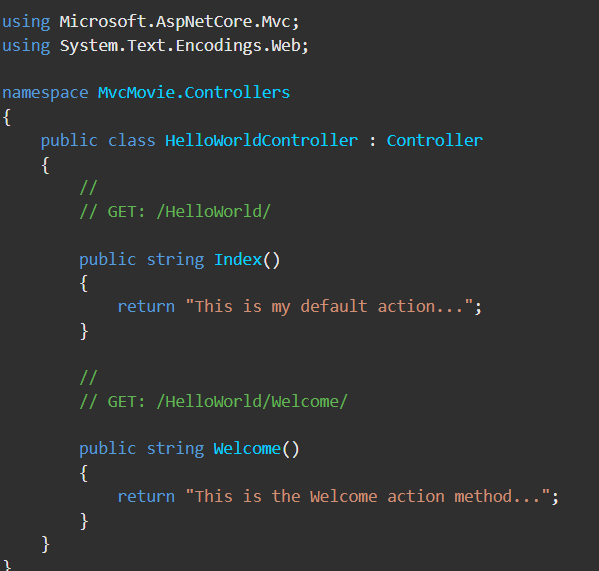
Ці принципи продемонстровані в цій серії посібників при створенні програми для роботи з фільмами. Проект MVC містить папки для *контролерів* та *уявлень*.



У діалоговому вікні **Додати новий шаблонний елемент** виберіть **Контролер MVC — порожній** > **Додати** .



Замініть весь вміст Controllers/HelloWorldController.csнаступним кодом:



Кожен метод public у контролері викликається як кінцева точка HTTP. У наведеному вище прикладі обидва методи повертають рядок. Зверніть увагу на коментарі, що передують кожному методу.

Кінцева точка HTTP:

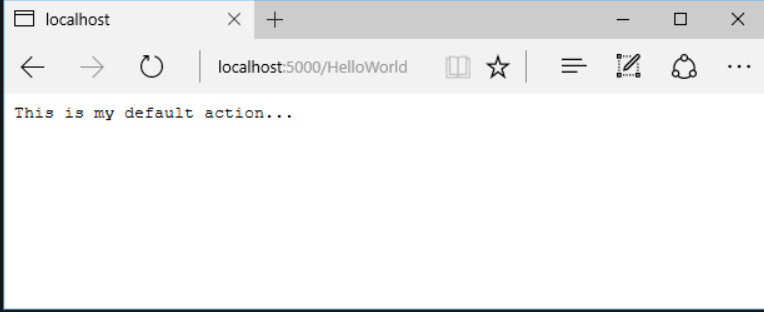
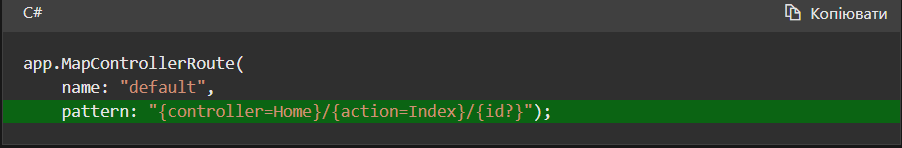
* Це URL-адреса, яка є цільовою у веб-програмі, наприклад https://localhost:5001/HelloWorld.
* Вона об'єднує:
  + використовуваний протокол: HTTPS;
  + мережеве розташування веб-сервера, включаючи порт TCP: localhost:5001;
  + цільовий універсальний код ресурсу (URI): HelloWorld.

У першому коментарі зазначено, що цей метод [HTTP GET](https://developer.mozilla.org/docs/Web/HTTP/Methods/GET) викликається шляхом додавання /HelloWorld/до базової URL-адреси.

У другому коментарі зазначено, що цей метод [HTTP GET](https://developer.mozilla.org/docs/Web/HTTP/Methods) викликається шляхом додавання /HelloWorld/Welcome/URL-адреси. Далі у цьому посібнику використовується механізм формування шаблонів створення методів HTTP POST, які оновлюють дані.

Запустіть програму без відладчика.

Додайте HelloWorld до шляху в адресному рядку. Метод Index повертає рядок.



MVC викликає класи контролера (і методи дії в них) залежно від вхідної URL-адреси. [Логіка маршрутизації URL-адрес](https://learn.microsoft.com/ru-ru/aspnet/core/mvc/controllers/routing?view=aspnetcore-6.0) , яка використовується моделлю MVC за умовчанням, визначає код, що викликається, на основі формату наступного вигляду:

/[Controller]/[ActionName]/[Parameters]

Формат маршрутизації задається у файлі Program.cs. Якщо при переході до програми не вказати сегменти URL-адреси, за замовчуванням встановлюються контролер Home та метод Index, які задані у виділеному вище рядку шаблону. У попередніх сегментах URL-адрес:

* Перший сегмент URL-адреси визначає клас контролера, який буде виконуватись. Тому localhost:5001/HelloWorld зіставляється із класом Controller **HelloWorld** .
* Друга частина сегмента URL визначає метод дії для класу. Отже, localhost:5001/HelloWorld/Indexвиконує метод Indexкласу  HelloWorldController. Зверніть увагу, що в цьому випадку достатньо перейти на адресу localhost:5001/HelloWorld, а метод Indexвикликається за замовчуванням. Якщо ім'я методу, що викликається, не вказано явно, для контролера викликається метод за замовчуванням Index.
* У третій частині сегмента URL-адреси ( id) вказуються дані маршруту. Дані маршруту описані далі у цьому посібнику.

Перейдіть на адресу https://localhost:{PORT}/HelloWorld/Welcome. Замініть {PORT}порт порту.

Метод Welcomeзапускається і повертає рядок This is the Welcome action method.... Для цього URL-адреси задані контролер HelloWorldта метод дії Welcome. Частина URL-адреси [Parameters]ще не використовувалася.