# MVC TASK 4

[Document subtitle]

**LinkedIn Article:**

https://www.linkedin.com/pulse/simplifying-object-mapping-aspnet-core-automapper-mohamed-afify-131af/

https://www.linkedin.com/pulse/understanding-http-response-status-codes-aspnet-core-mohamed-afify-etbkf/

## Purpose of async / await

In C#, these keywords are used to:

- Run **asynchronous operations** (network, database, file I/O) without freezing the program.
- Keep the **UI or server responsive**.
- Write code that **looks simple and sequential** instead of using callbacks.

## How They Work

- async before a method tells the compiler "this method contains await".
- await says "wait for this operation to complete **but don't block the thread**; resume here when it's done".

## Simple Example

```
public async Task FetchDataAsync()
{
   var data = await GetDataFromServerAsync();
   Console.WriteLine(data);
}

public async Task<string> GetDataFromServerAsync()
{
   await Task.Delay(2000); // pretend it's slow I/O
   return "Data from server";
}
```

What happens:

1. FetchDataAsync calls GetDataFromServerAsync.
2. await releases the thread so it can do other work.
3. After 2 seconds the data comes back and the method continues after await.