Session 12

Implementing AutoMapper in ASP.NET Core

Session Overview

- Define AutoMapper
- List reasons for using AutoMapper
- Explain the installation and configuration process for AutoMapper
- Describe use of AutoMapper in an application

Introduction to AutoMapper

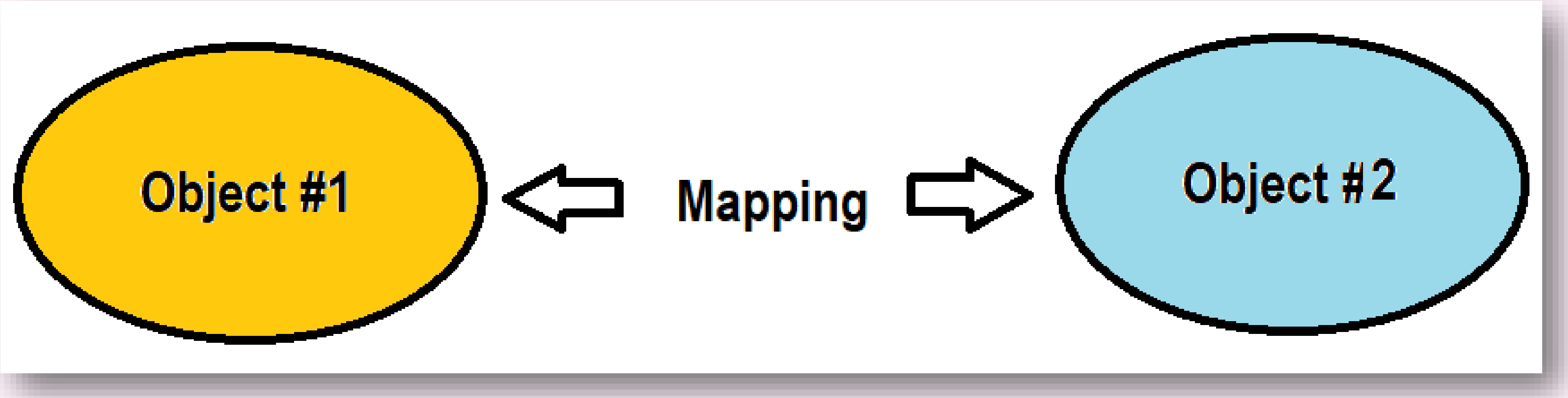
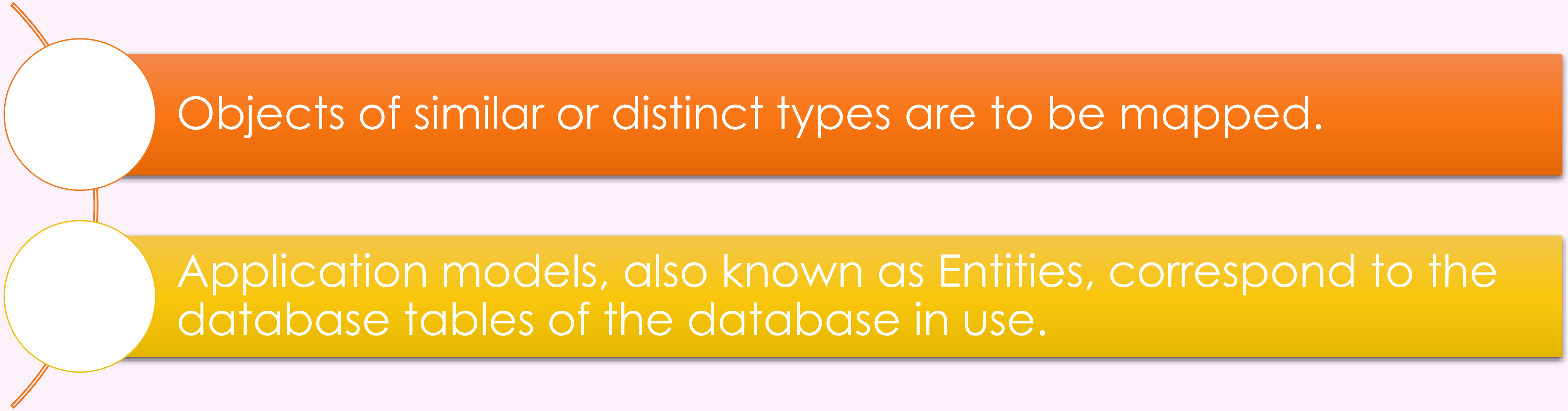


Figure 12.1: Object-to-Object Mapping

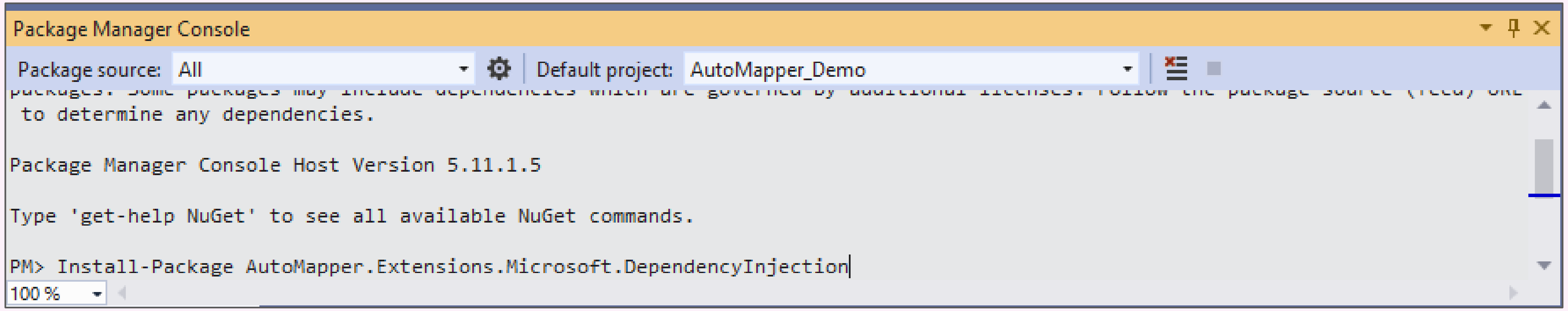
Uses of AutoMapper

AutoMapper can be useful in following scenarios:



- Objects of similar or distinct types are to be mapped.
- Application models, also known as Entities, correspond to the database tables of the database in use.

Working with AutoMapper



Package source: All

Default project: AutoMapper_Demo

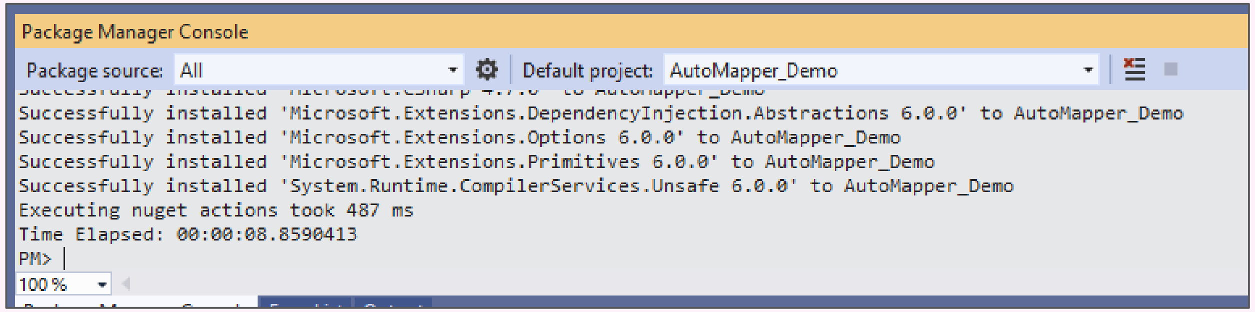
```
packages. Some packages may include dependencies which are governed by additional licenses. Follow the package source (feed) URL
 to determine any dependencies.

Package Manager Console Host Version 5.11.1.5

Type 'get-help NuGet' to see all available NuGet commands.

PM> Install-Package AutoMapper.Extensions.Microsoft.DependencyInjection
```

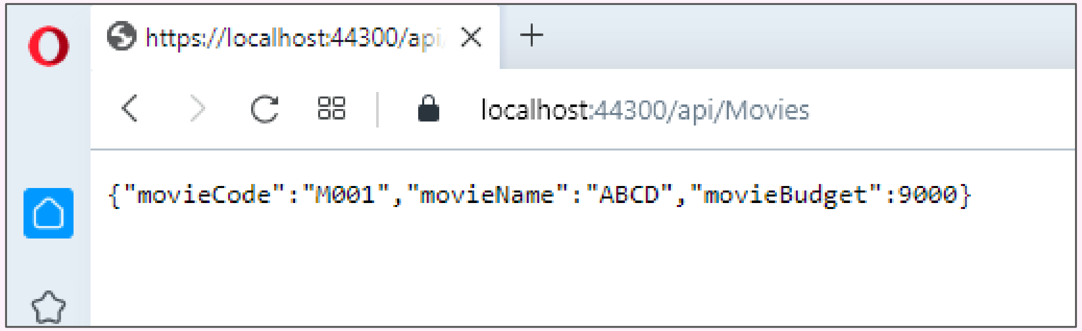
Figure 12.2: Installing AutoMapper



Package source: All

Default project: AutoMapper_Demo

```
Successfully installed 'Microsoft.CSharp 4.7.0' to AutoMapper_Demo
Successfully installed 'Microsoft.Extensions.DependencyInjection.Abstractions 6.0.0' to AutoMapper_Demo
Successfully installed 'Microsoft.Extensions.Options 6.0.0' to AutoMapper_Demo
Successfully installed 'Microsoft.Extensions.Primitives 6.0.0' to AutoMapper_Demo
Successfully installed 'System.Runtime.CompilerServices.Unsafe 6.0.0' to AutoMapper_Demo
Executing nuget actions took 487 ms
Time Elapsed: 00:00:08.8590413
PM>
```



localhost:44300/api/Movies

```
{"movieCode":"M001","movieName":"ABCD","movieBudget":9000}
```

Figure 12.3: Installed AutoMapper

Figure 12.4: Output of Using AutoMapper

Summary

- ✓ AutoMapper is a convention-based object-to-object mapper for .NET, which can be used when there are a large number of members in objects.
- ✓ AutoMapper changes an input object to an output object of a different type.
- ✓ AutoMapper saves time and effort while manually mapping the properties of incompatible types in the application.
- ✓ AutoMapper can be used not only in mapping similar objects, but also dissimilar objects.
- ✓ One can be install AutoMapper in a project by adding AutoMapper.Extensions.Microsoft.DependencyInjection via Package Manager.
- ✓ Developers can dynamically retrieve a type of existing object and invoke its methods or access its fields and attributes using AutoMapper.