November 7, 2020

Calin Doran

Software DEveloper

Personal Project

Commander

.NET Core, MVC, REST API

Table of Contents

[Introduction 2](#_Toc56795814)

[APIs & Interfaces 4](#_Toc56795815)

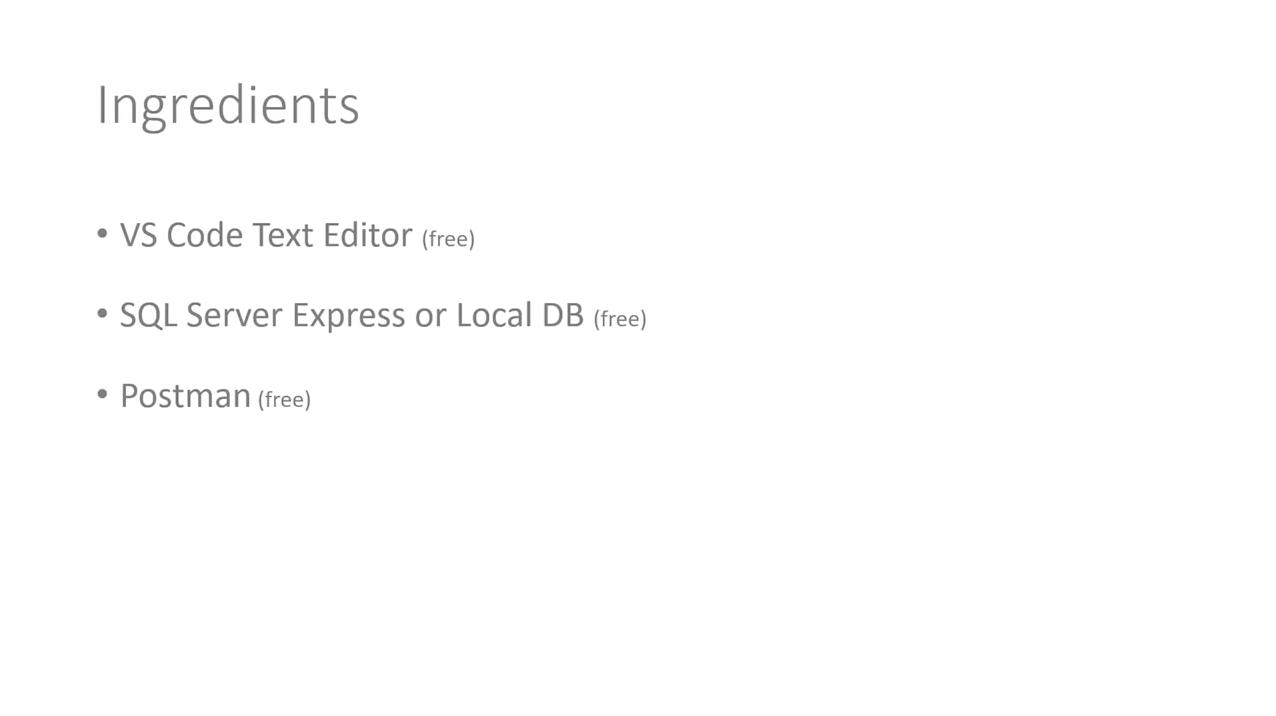
[Database set-up and Migrations 5](#_Toc56795816)

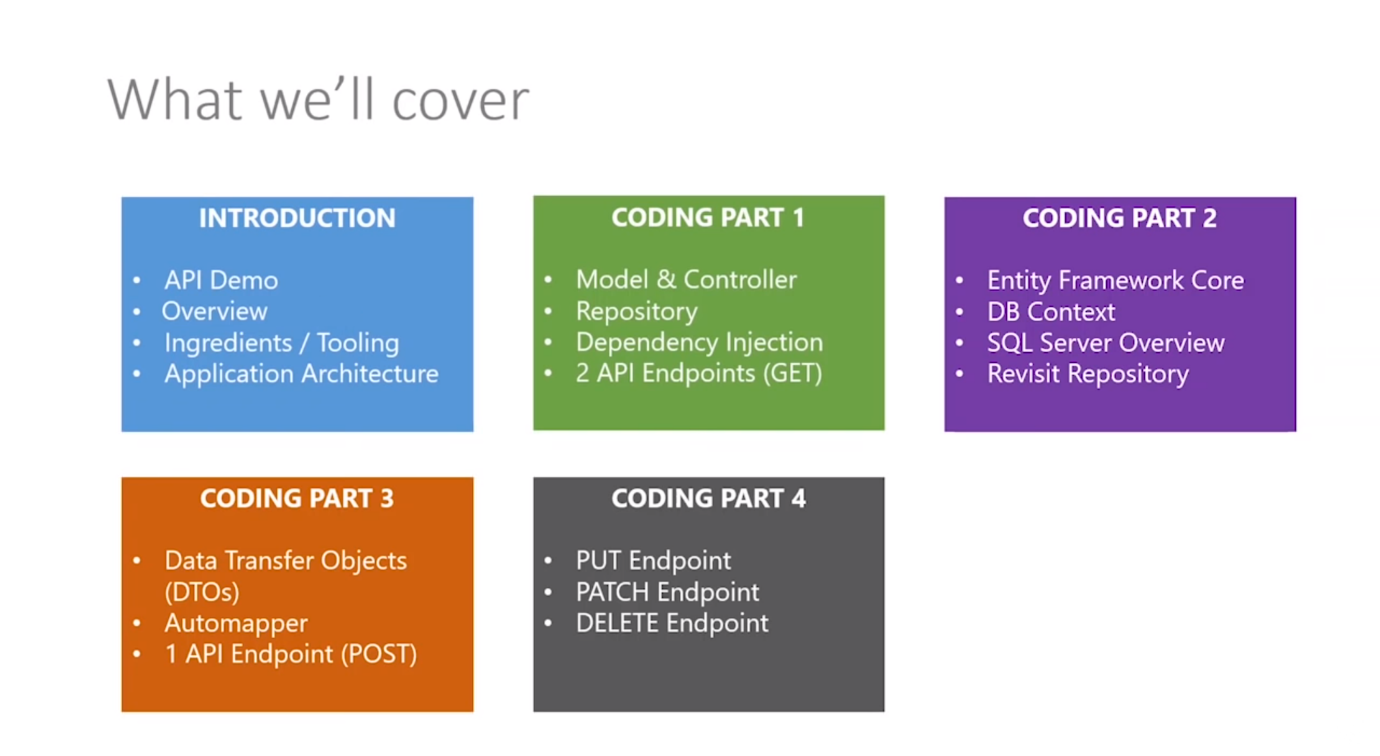
[Data Transfer & Client Requests 7](#_Toc56795817)

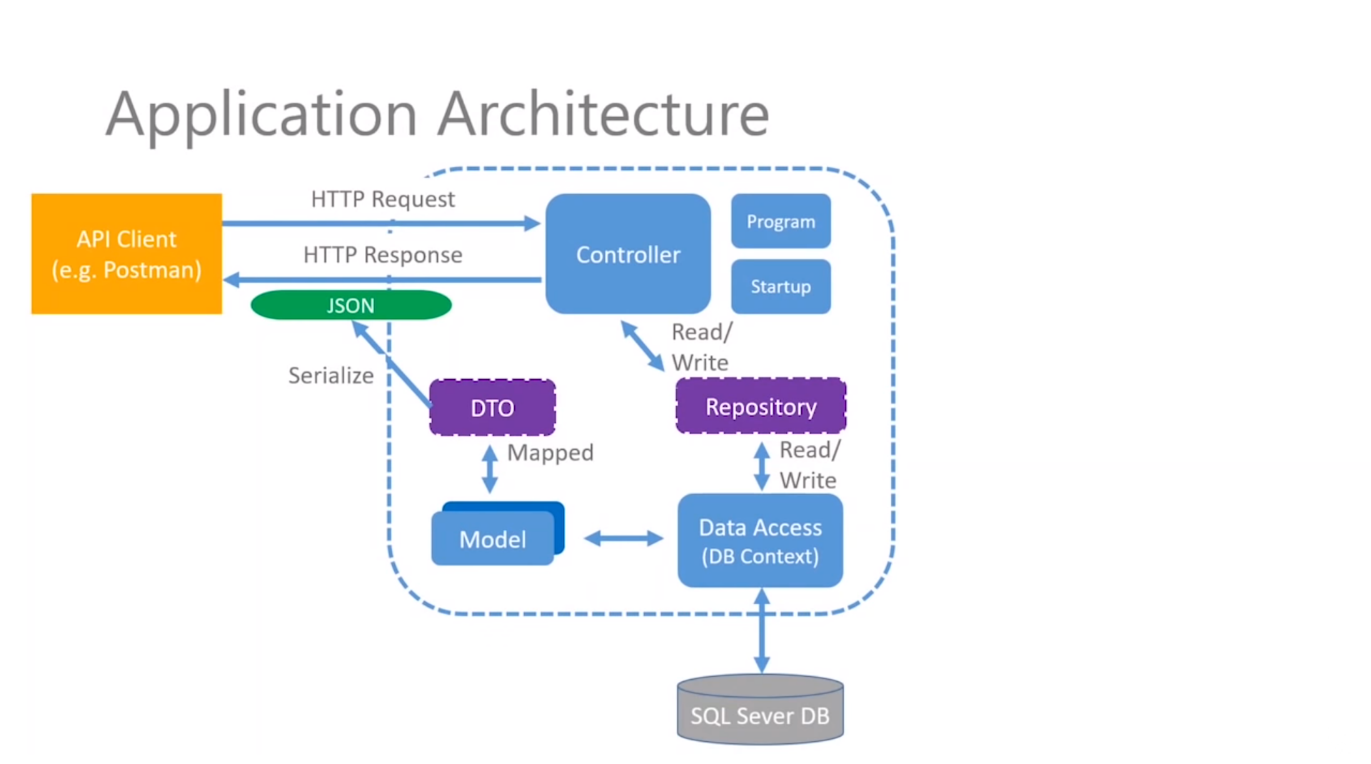
[Endpoint Update Patch and Delete 9](#_Toc56795818)

# Introduction

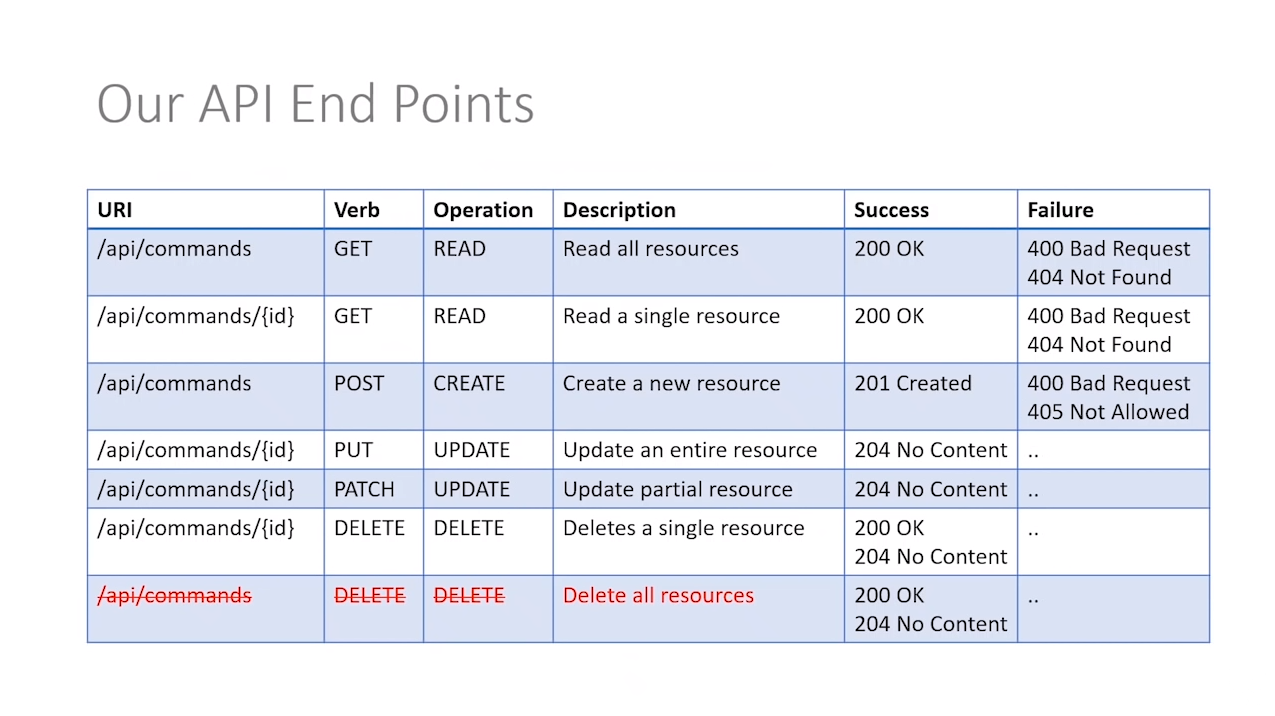
In this project the developer aims to create an application that will provide commands for different platforms such as EF Core, .NET Core and so on. It will keep up commands for said platforms and allow the user to easily choose from which platform they need commands for. It can be difficult as a developer to try and remember every different command used on a daily basis, this helps to solve this problem.

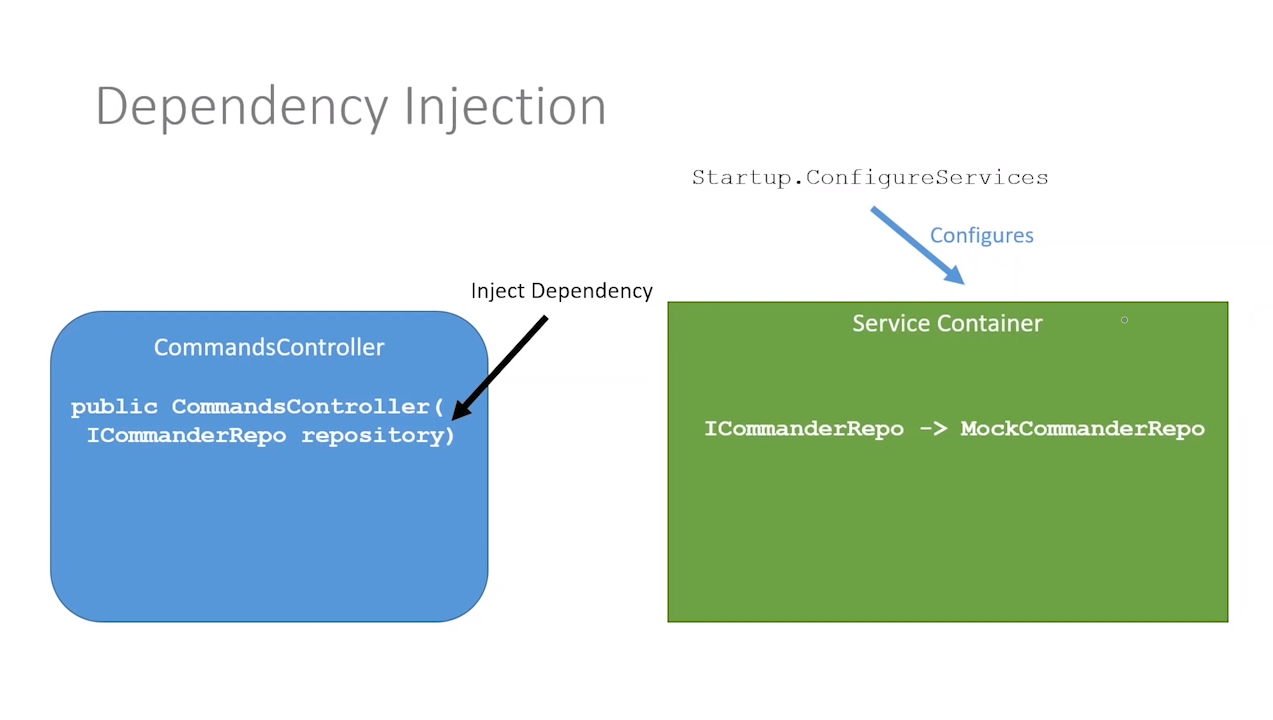


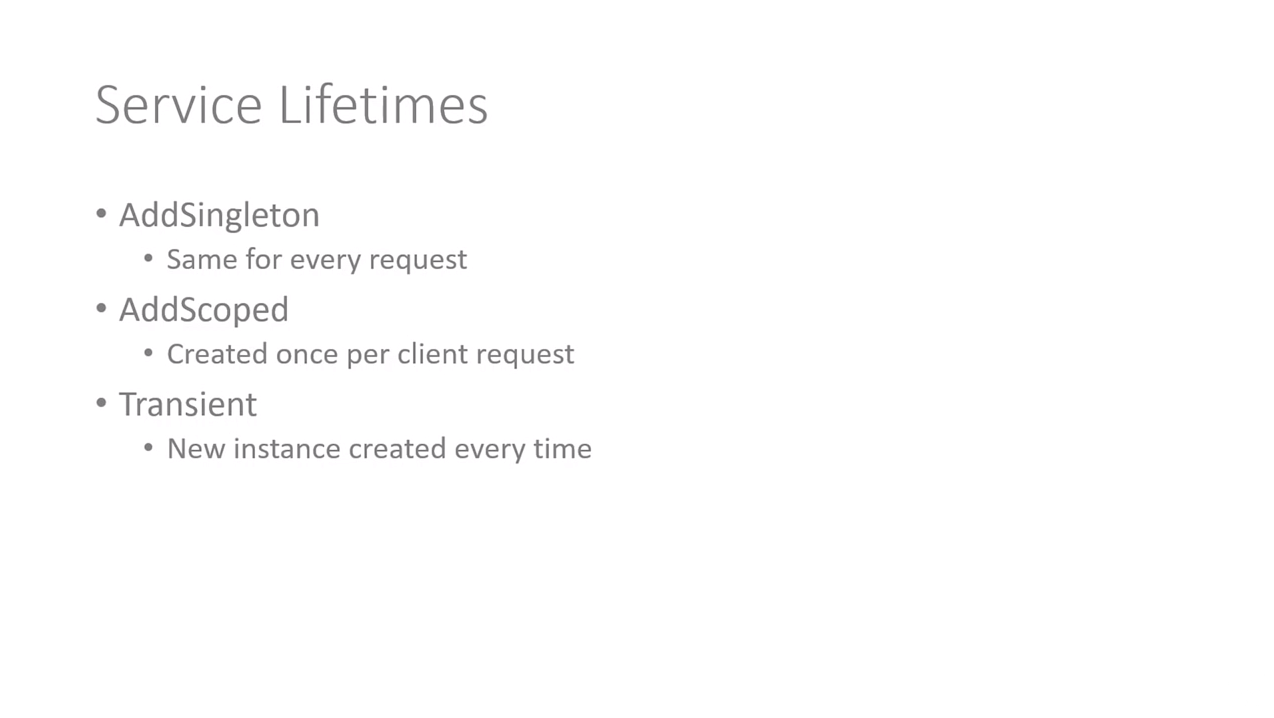




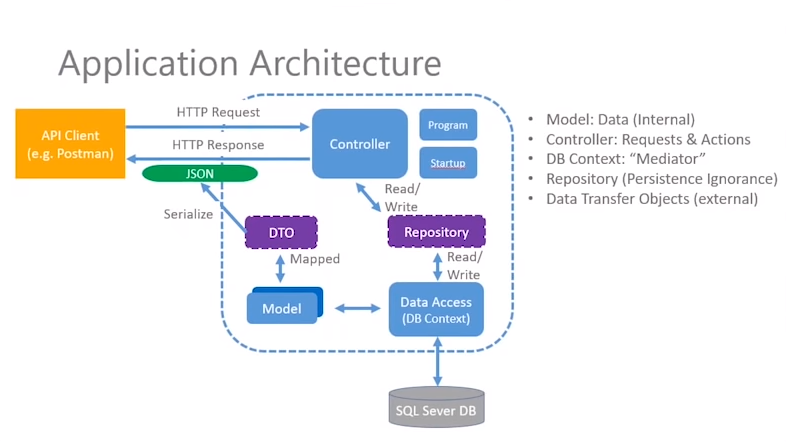
# APIs & Interfaces







# Database set-up and Migrations



Nuget.org

Install via command line with dotnet add package.

Packages needed:

1. Microsoft.EntityFrameworkCore
2. Microsoft.EntityFrameworkCore.Design
3. Microsoft.EntityFrameworkCore.SqlServer

No need for the –version as it will install the latest without that part of the command.

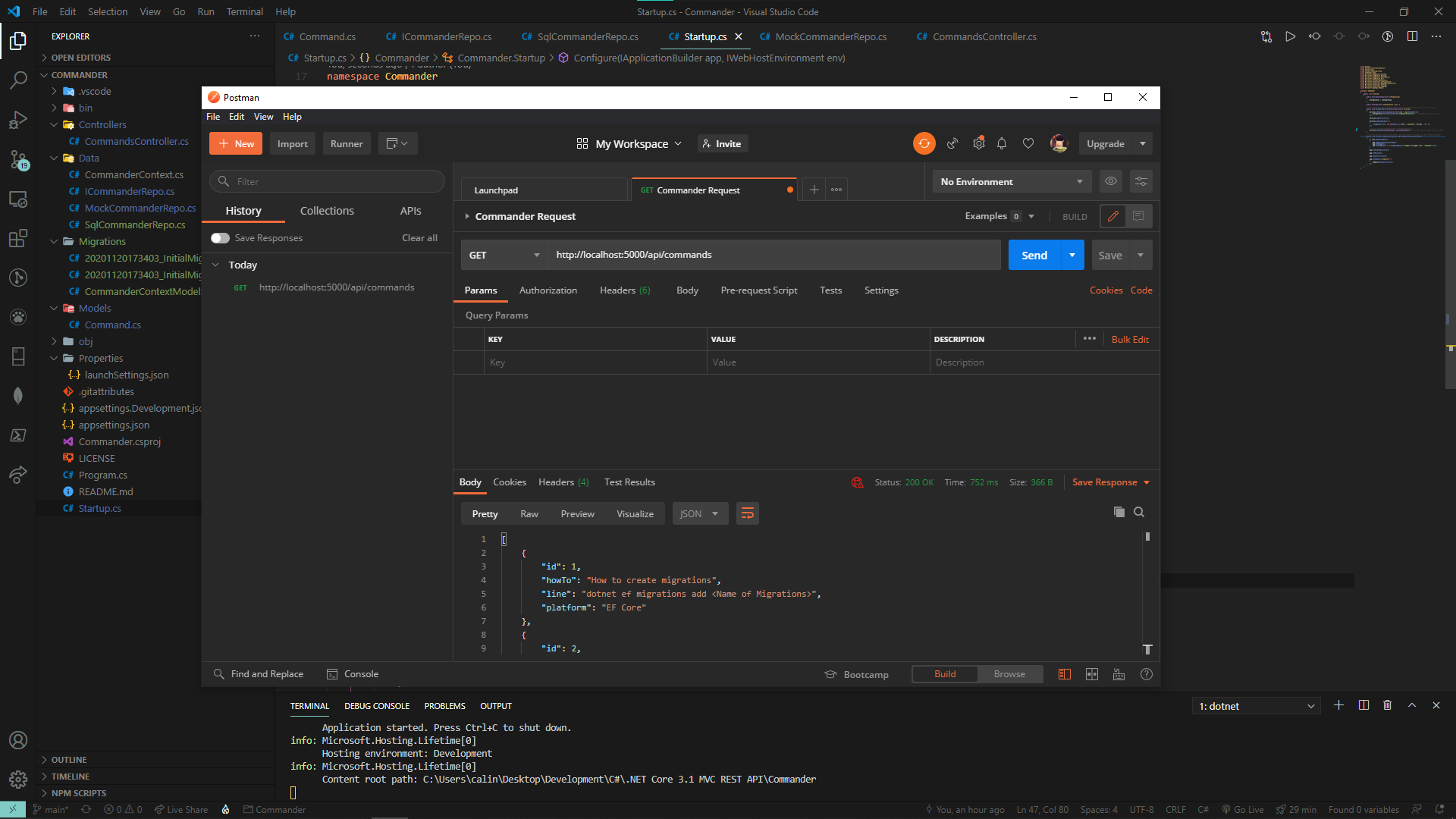
If there is an error with dotnet ef then we will need this command:

“dotnet tool install --global dotnet-ef”

“dotnet ef migrations add InitialMigration” to create a migration file.

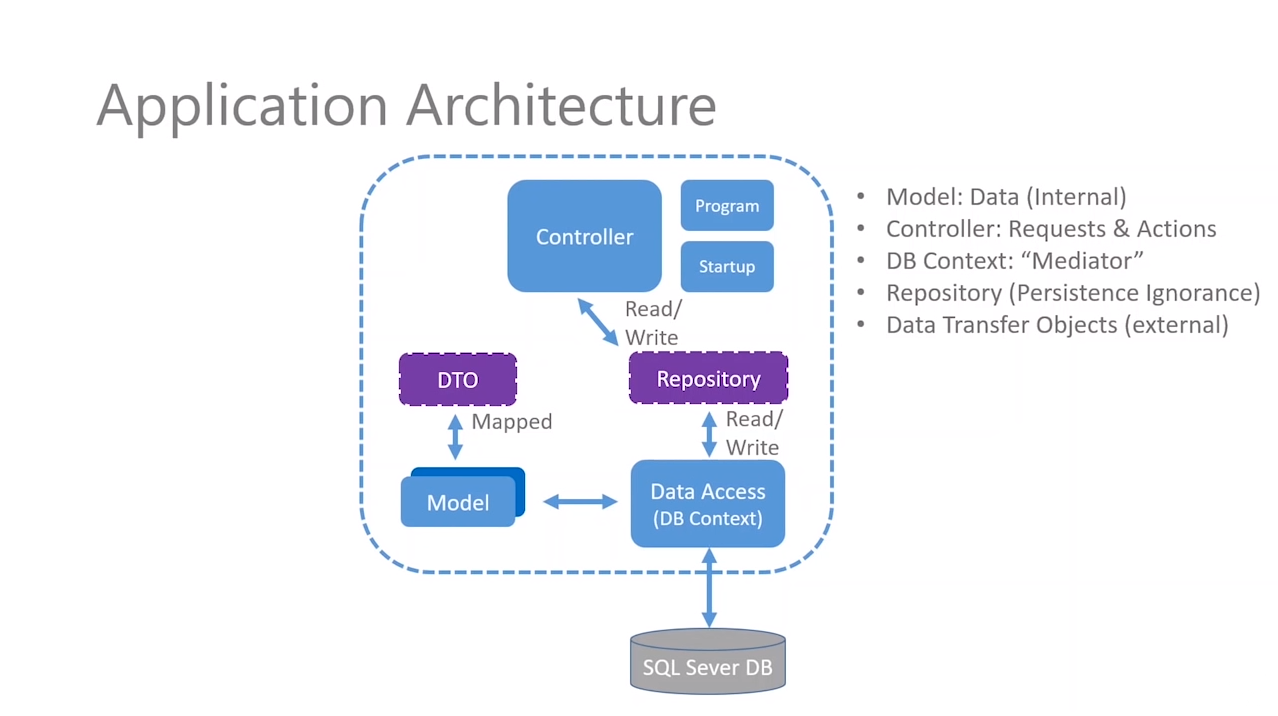
Modified the constraints of our model so that its not null and maxlength is 250. Once this is done, we can run the command “dotnet ef database update” and it should create the database CommanderDB.

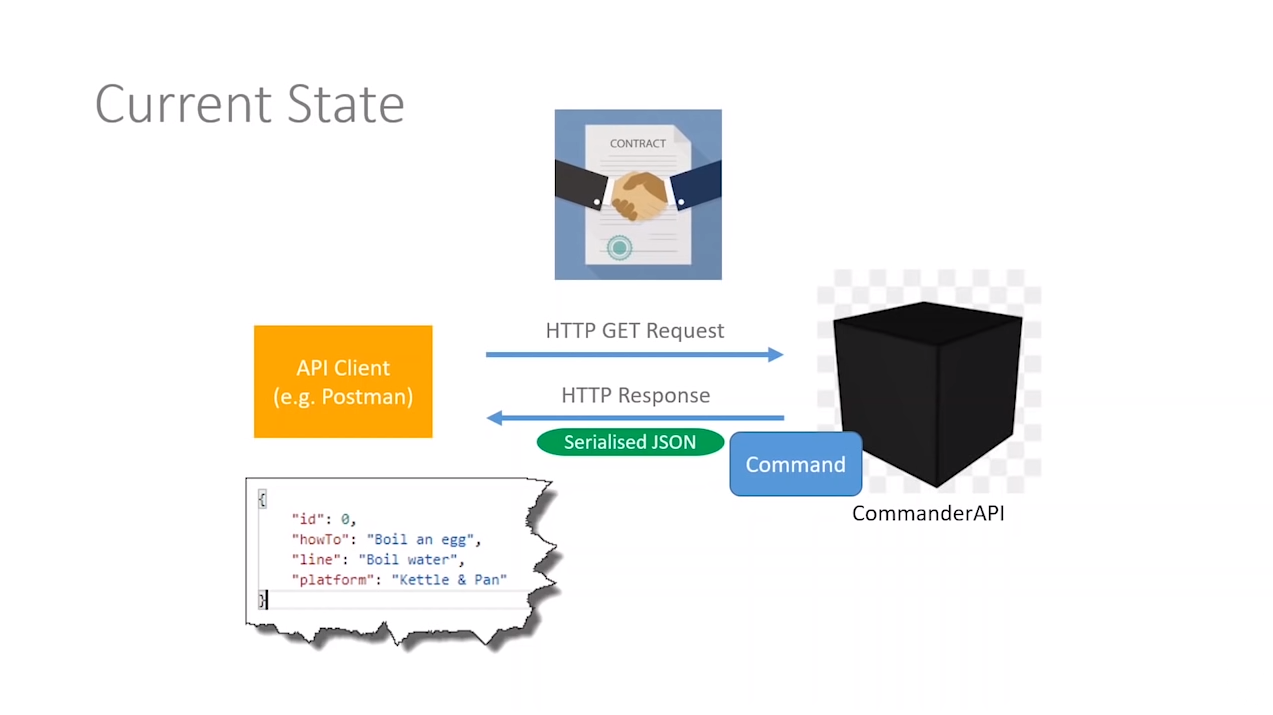
From here we can look at the database and edit/update fields with data for our commands table. And we are done with the database, model and migration setup.

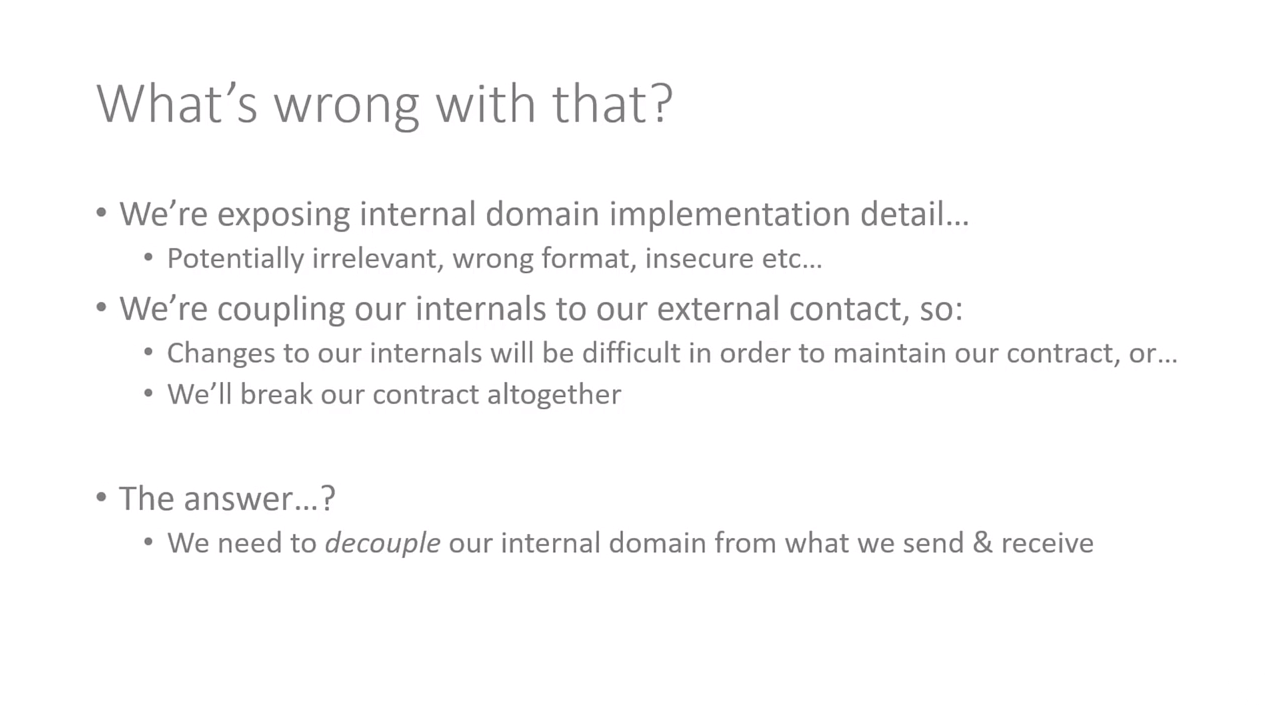


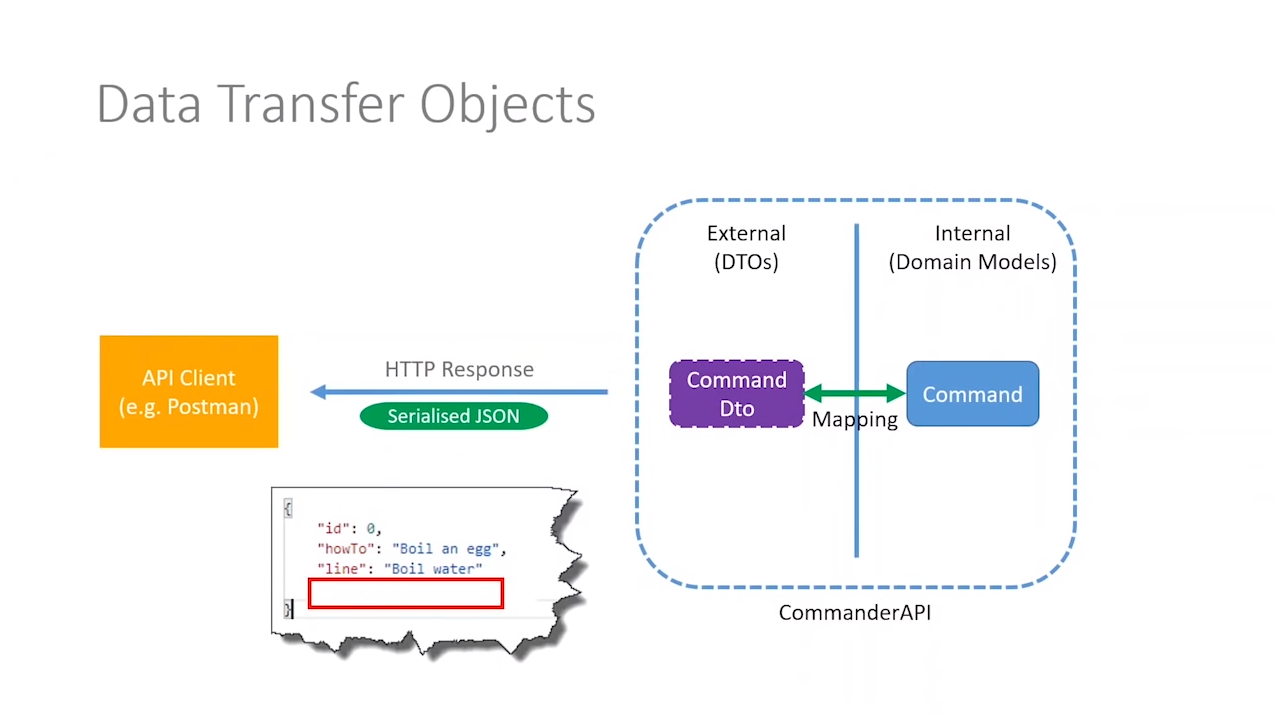
We are getting actual information from our sqldb CommanderDB using the right context, this is great.

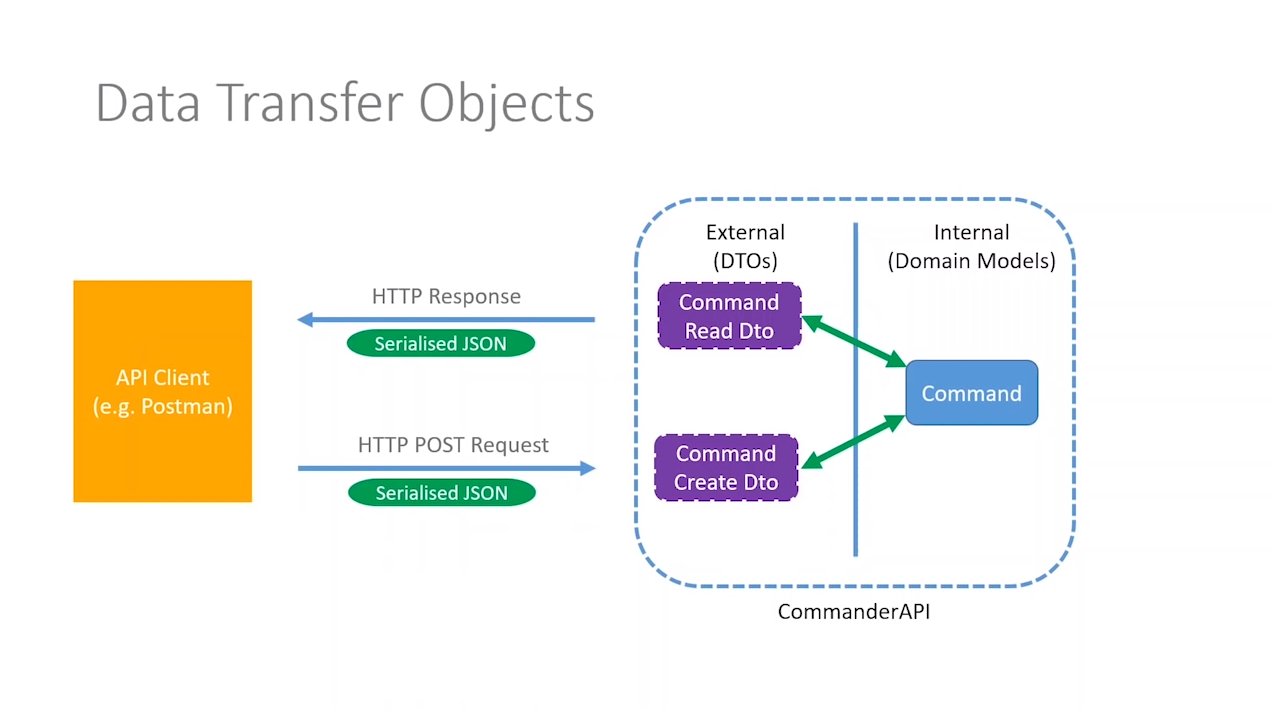
# Data Transfer & Client Requests









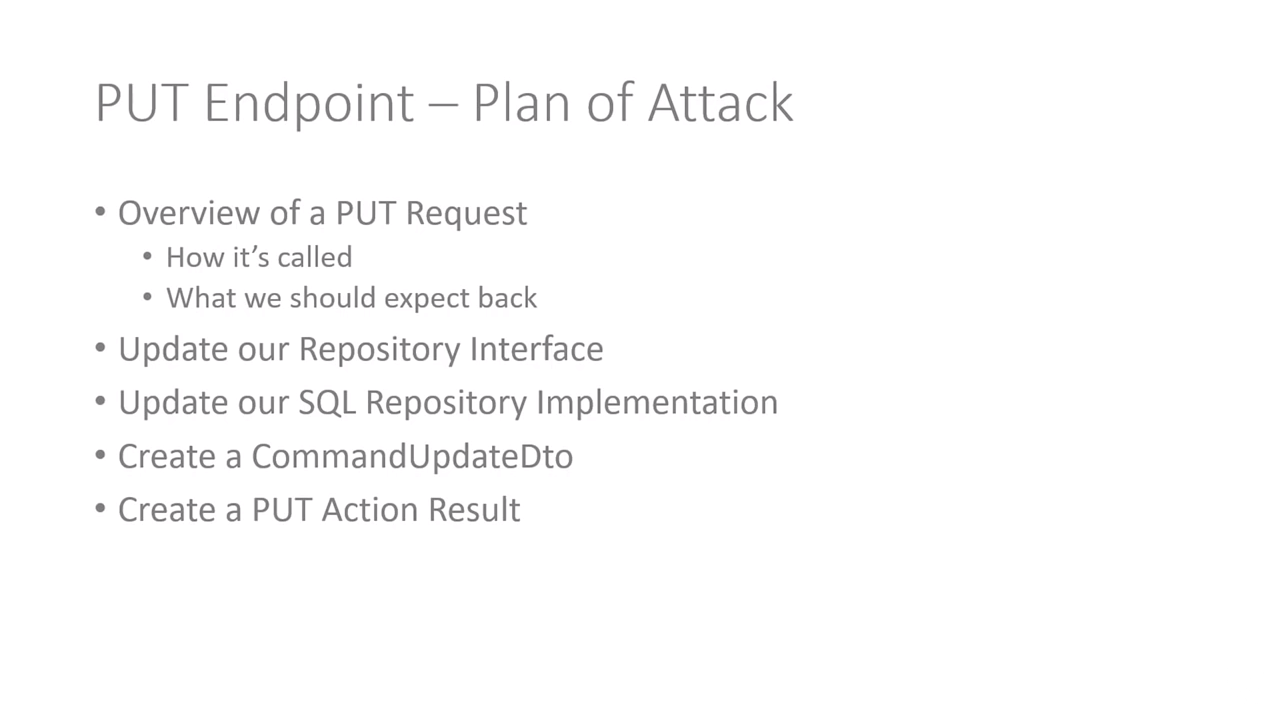


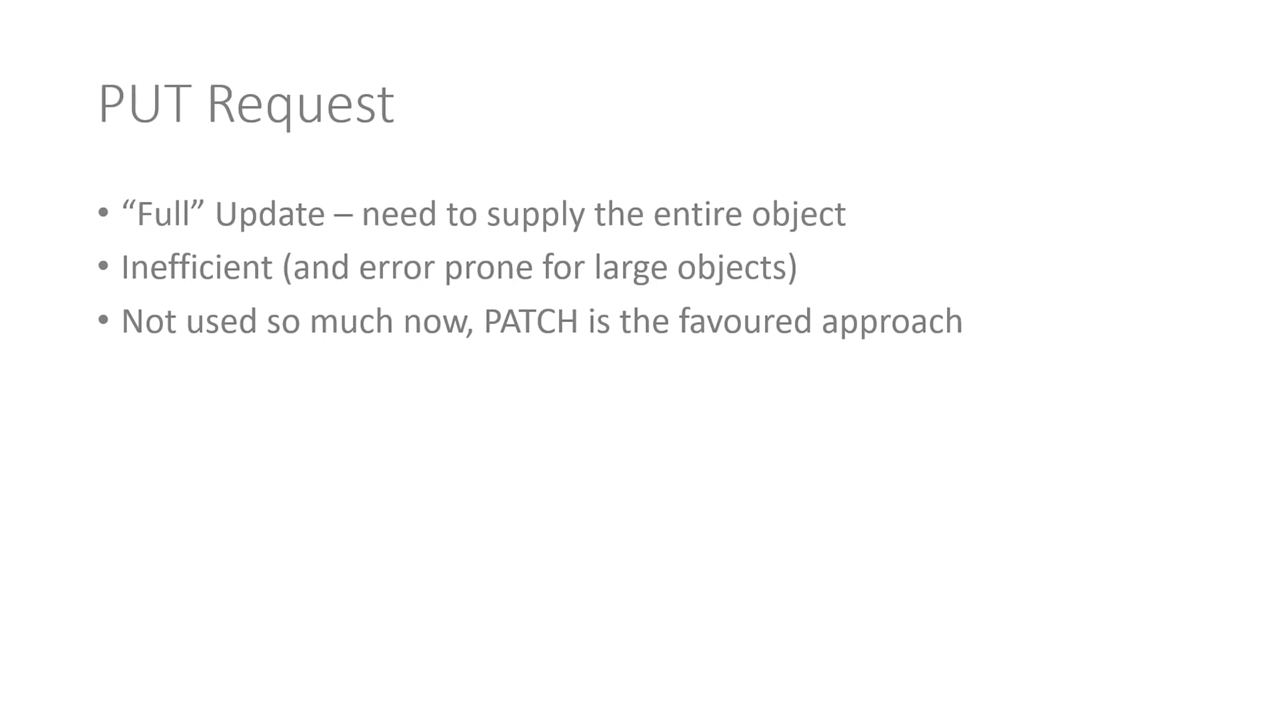
Adding additional packages from nuget.org,

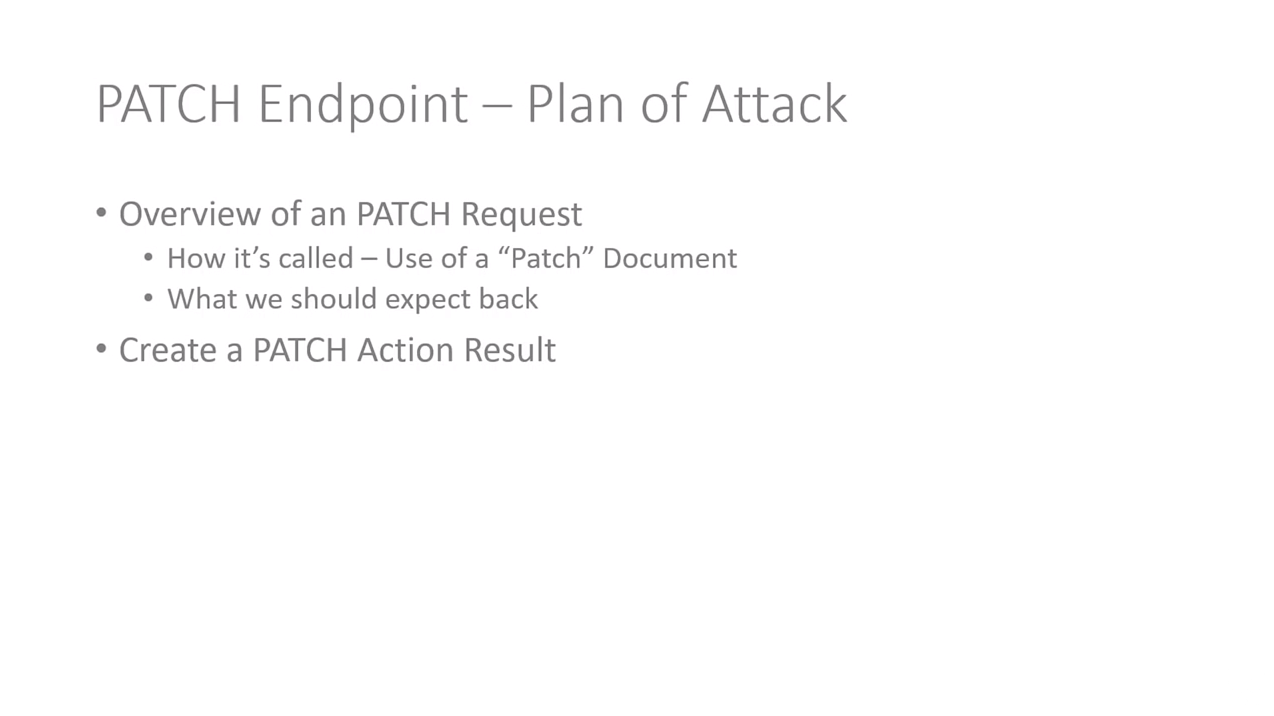
Packages:

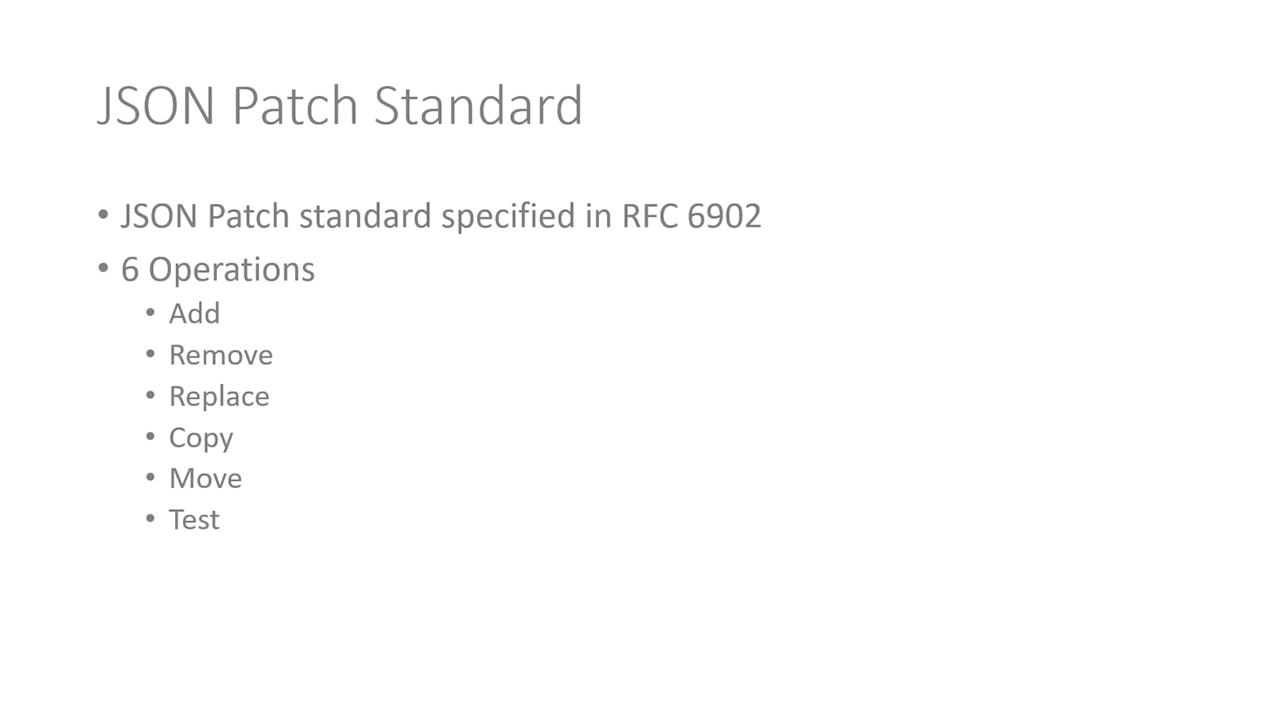
1. dotnet add package AutoMapper.Extensions.Microsoft.DependencyInjection

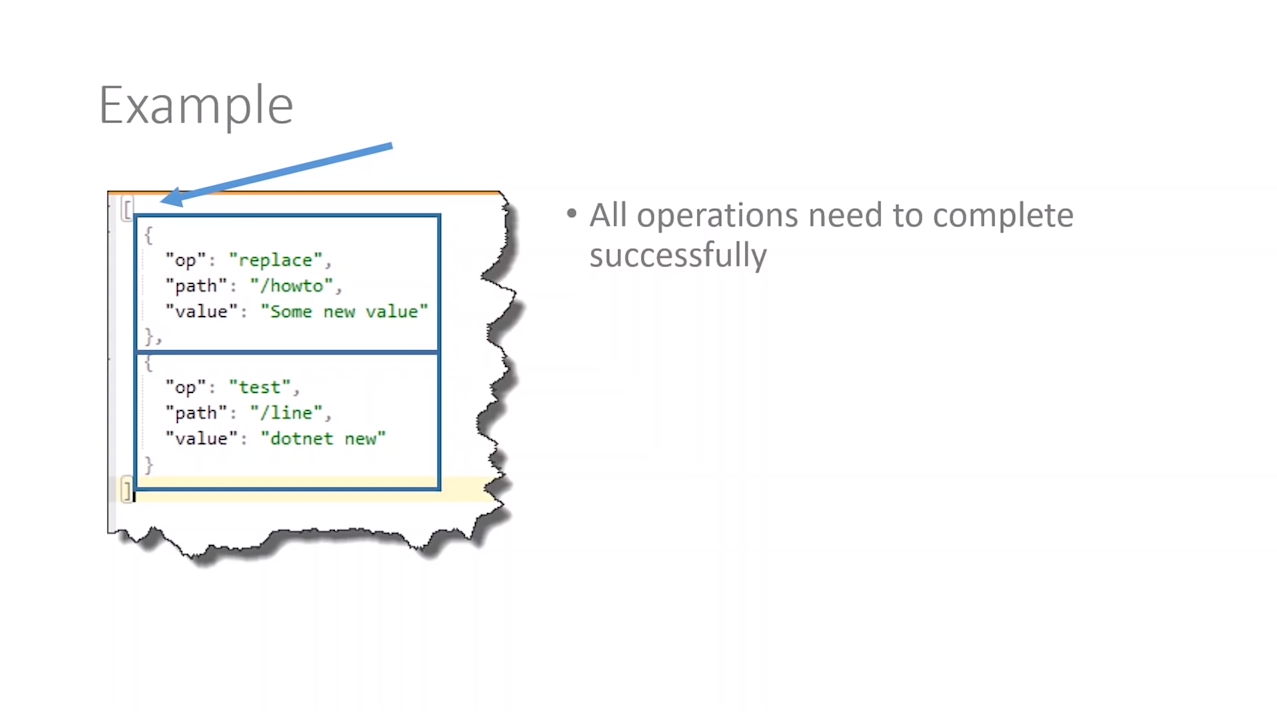
# Endpoint Update Patch and Delete











Adding additional packages from nuget.org,

Packages:

1. Microsoft.AspNetCore.JsonPatch
2. Microsoft.AspNetCore.Mvc.NewtonsoftJson