Building Your First API with ASP.NET Core (Kevin Dockx, Pluralsight)

# Overview

.NET Core versions can be installed side by side.

.NET Core runs on both .NET Framework and .NET Core

Program\Main() is the starting point of the application. Startup class is the entry point for a web application.

ConfigureServices() is used to add services to the container, and to configure those services.

Hosting environment can be configured for a .NET ASP Core application: Development, Staging, Production or Custom.

Middleware can be added to a .net Core application. Middleware are components that fit in the request/response pipeline.

# Creating the API and returning resources

ASP.NET Core MVC is used to create both http services and web applications.

The nuget package Microsoft.AspNetCore.All contains the MVC framework package (Microsoft.AspNetCore.Mvc) and other useful packages (entity framework, etc.).

RunTime store is a sort of GAC.

# HTTP Status Codes

100: Informational (not used by APIs)

200: Success

* 200 GET success
* 201 Resource created
* 204 Success, does not return anything (eg. DELETE)

300: Redirection (APIs don’t usually use these)

400: Client Error

* 400 Bad request
* 401 Unauthorized
* 403 User does not have access to the resource
* 404 Not found
* 409 Conflict (eg. Conflict between two simultaneous updates)

500: Server Error

* 500 Internal Server Error