درس 14

در این درس ابتدا در کلاس program سیستم مسیریابی رو تنظیم می کنیم



قسمت هایی که با رنگ زرد مشخص شدن برای مسیر یابی ضروری هستند.

بعد از این کار یک کنترولر میسازیم و با دادن روت ان را اجرا می کنیم.



قسمتی که با رنگ زرد مشخص شده است روت است و با همین ادرس می توان به این کنترولر دست پیدا کرد.

درس 15 تنظیم مسیر یابی کنترولر ها

برای تنظیم مسیر یابی کنترولر ها در جلسته قبل روت رو اکشن بود که کارایی نداشت. در این جلسه روت رو کنترولر هست. به دو روش می توان گذاشت که با توجه به تجریه استاد مورد سبز بهتر است اما روش زرد رنگ هم درست است.



درس 16

معرفی و نصب Postman

درس 17 ساخت DTO

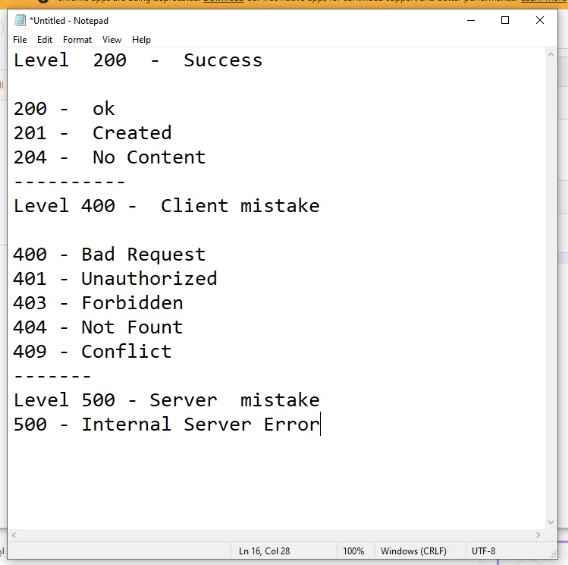
در این جلسه DTO ها رو ساختیم و مدل city بعد این کار موردی که جلسه پیش بود و اگر دو کنترلر با اندیس get داشتیم چطور این دو تا رو از هم متمایز کنیم را پاسخ دادیم.



باتوجه به قسمت زرد و سبز می توان بادادن اندیس ورودی روت این دو تایع Get ر ا از هم متمایز کنیم.

درس 18

کد های خطا در سمت سرور و کلاینت و موفقیت



درس 19

پیاده سازی کدها و کدها ی خطا و مشاهده و استفاده از IActionResult



درس 20

ایجاد مدل جدید و مقدار دهی به ان



و ایجاد ارتباط با مدل قبلی



قسمت رنگی یک پراپرتی تعریف شده که در همین سمت کلاینت مقدار دهی می شود و در دیتا بیس فیلد ندارد. فقط set دارد در سمت کلاینت محاسبه می شود و مقدار ان برگشت داده می شود. فقط readonly هست.

درس 21

در این قسمت برای pointofinterest کنترولر می نوسیم.



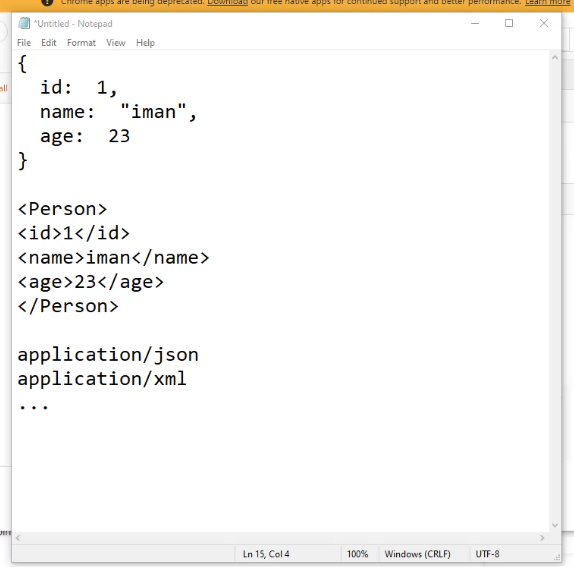
برای کل کنترولر از روت زرد رنگ استفاده می کنیم. با توجه به اینکه زیاد نمی خواهیم روت ایجاد کنیم روت را در ادامه Cities می نوسیم.

در قسمتی که سبز است روت ادامه روت زرد می شود.



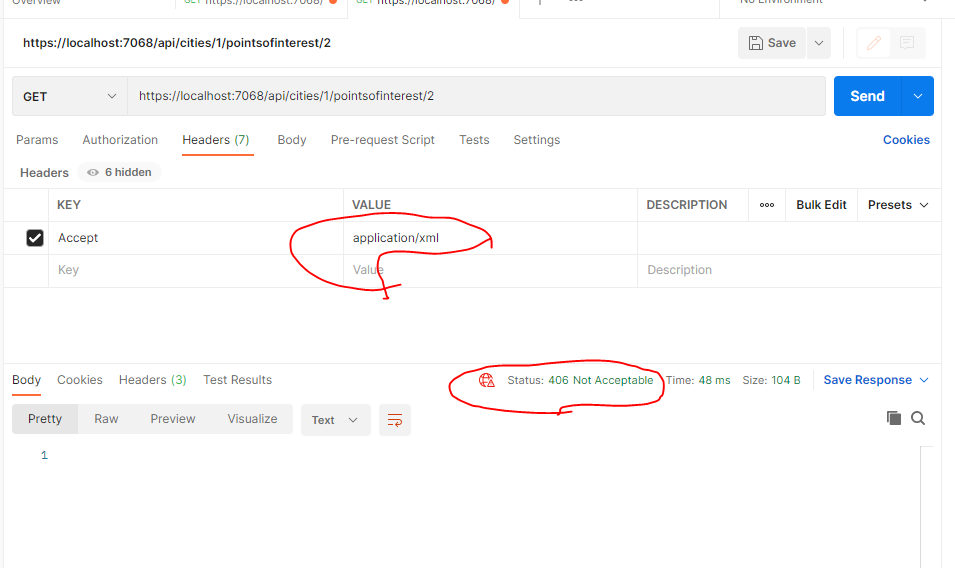
درس 22

در این جلسه در مورد فرمت وردی وخروجی صحبت شد. دو فرمت معروف JSON و XML بررسی شد. ورودی و خروجی باید فرمت مورد تایید خود را در هدر معرفی کنند.



درس 23

این جلسه ادامه جلسه قبل هست و تعیین فرمت وردی وخروجی بررسی می شود. این کار در قسمت Heder صورت می گیرد





با اضافه کردن کد بالا به Program زمانی که فرمت درخواست پشتیبانی نشود کد خطای 406 نمایش داده می شود.

برای پشتیبانی از فرمت XML از باید کد زیر را اضافه کنیم.



فرمت خروجی و وردی به صورت پیش فرض json است برای تغییر این پیش فرض از کدهای زیر استفاده می کنیم:



* 1. درس 24 دانلود فایل از طریق WebApi

برای دانلود فایل ابتدا یک کنترولر میسازیم :

اکشن دانلود را میسازیم از نوع GET هست. برای شناسایی فایل دانلودی به عنوان ورودی یک ایدی به فایل دانلود شده میدهیم.

1. ابتدا مسیر فایل را می دهیم. رنگ قرمز. ازانجای که الان فایل در روت است فقط اسم فایل وارد شده اگر داخل فولدر بود آدرس ان را می دادیم.
2. از قسمت زرد رنگ زرد برای چک وجود فایل استفاده می کنیم که در صورت نبودن فایل خطا برگرداند.
3. اگر فایل وجود داشت از خط سبز رنگ برای تبدیل به بایت کردن فایل استفاده می کنیم.
4. برای دانلود باید نوع فایل را به خروجی بدهیم در aspcore ما از کتابخانه فیروزه ای برای فهمیدن نوع فایل رنگ استفاده می کنیم. البته باید این کنابخانه را به سرویس های معرفی کنیم و اینجکت می کنیم رنگ خاکستری.



1. در قسمت صورتی رنگ هم فایل را برمی گردانیم. بایت شده فایل دانلود سپس نوع فایل و اسم فایل.

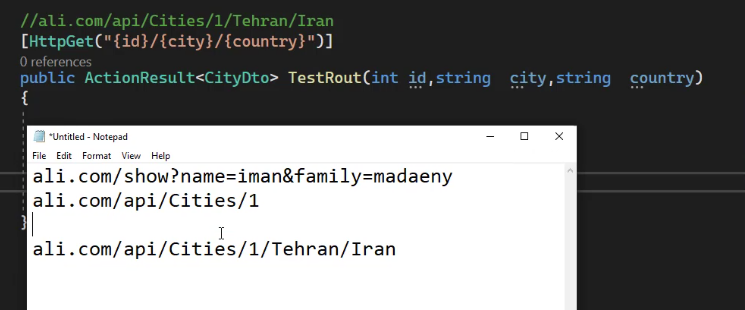


* 1. درس 25: روش های ارسا ل اطلاعات به API

1. استفاده از کوئری استرینگ ها: در این روش بعد از ? مطالب ارسالی به صورت json ارسال می شوند.



1. گرفتن اطلاعات از روت. در این مورد با قرار دادن اطلاعات در روت داده ها به api ارسال می شوند. دقت کنید روت در هنگام ران تایم قابل تغییر نیست. ولی کوئری استرینگ قابل تغییر است.



* 1. درس 26 ثبت اطلاعات از طریق frombody

در این قسمت Post بررسی می شودو اطلاعا ت FromBody به مرکز منتقل می شود.



* 1. جلسه 27 اعتبار سنجی مقادیر وردی API

در این قسمت با معرفی کتابخانه زیر اعتبار سنجی انجام شد.



* 1. درس 28 ویرایش اطلاعات از طریق HttpPut

در این جلسه از PUT برای ویرایش اطلاعات استفاده می کنیم.

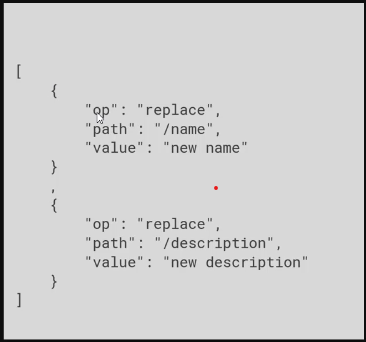


این اپ دیت کلی ابجکت هست و در ان تمام فیلدهای ابجکت به روز می شود.

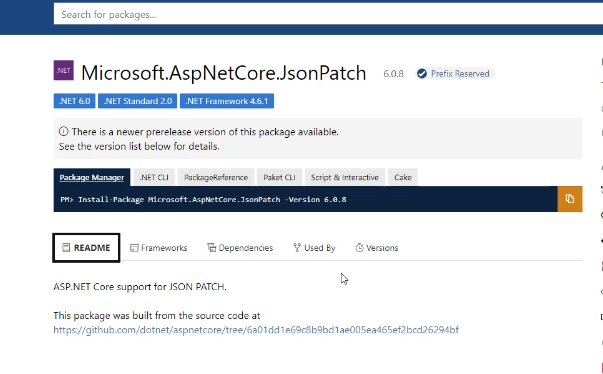
* 1. درس 29 معرفی HttpPatch برای ویرایش جزئی

برای ویرایش جزئی یک شی به جای استفاده از Put که تمام پراپرتی ها را اپدیت می کند از Patch استفاده می کنیم. در این روش بجای اینکه کل شی اپدیت شود فقط جزئی که می خواهیم را به روز می کنیم.

همانطور که از شکل زیر مشخص هست op عملیات را مشخص می کند که در اینجا جایگزینی هست path پراپرتی یا کلید را مشخص می کندو value مقدار جدید را مشخص می کند اگر قرار بود کلید دیگری تغییر بکند با کاما جدا می کنیم و json جدید را می نوسیم:



برای استفاده از Patch بایداز پکیج زیر استفاده کنیم:



* 1. درس 30 نصب بسته ها مورد نیاز Patch و تغییر فرمت از Json.Net

در این قسمت دو پکیج زیر را به لایه Api اضافه می کنیم:



سپس در فایل program.cs این سرویس را اضافه می کنیم: