

مایکروسافت به تازگی ASP.NET Core 1.0 را منتشر کرد. ASP.NET Core یک طراحی جدید و قابل توجهی از ASP.NET است و در این مقاله می‌خواهیم یک برنامه ساده را با استفاده از آن پیاده سازی کنیم .

شروع کار با ASP.NET Core MVC

برنامه نویسی را از برنامه نویسان حرفه ای بیاموزید



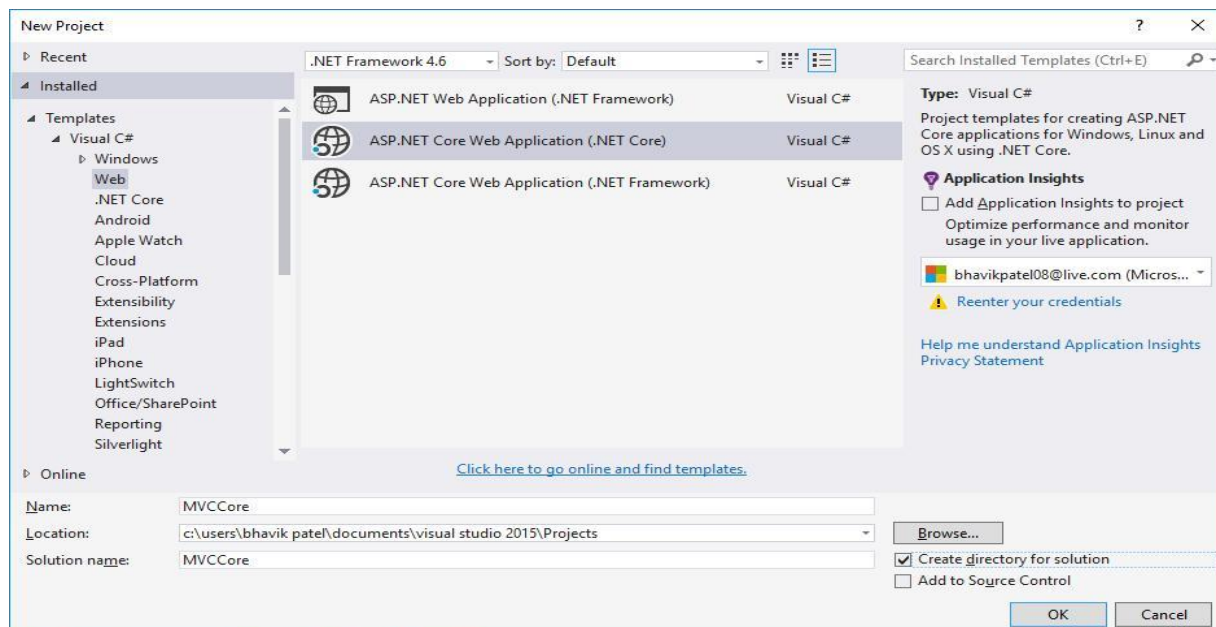
ASP.NET Core یک منبع باز (open source) است و قابل توسعه بر روی پلتفرم های دیگر است ASP.NET .
Core میتواند بر روی NET Core و .NET Framework اجرا شود.

در نسخه های قبلی WEB API , MVC تقریباً یکسان بودند. تا به حال assemblie های MVC 5 را مشاهده کرده اید ؟ اگر آنها را مشاهده کرده باشید متوجه می شوید که Web API متعلق به System.Net.Http است و MVC متعلق به System.Web.Mvc است. اما در ASP.NET Core هر دو آنها را در یک assemblie به نام Microsoft.AspNetCore.Mvc قرار داده است.

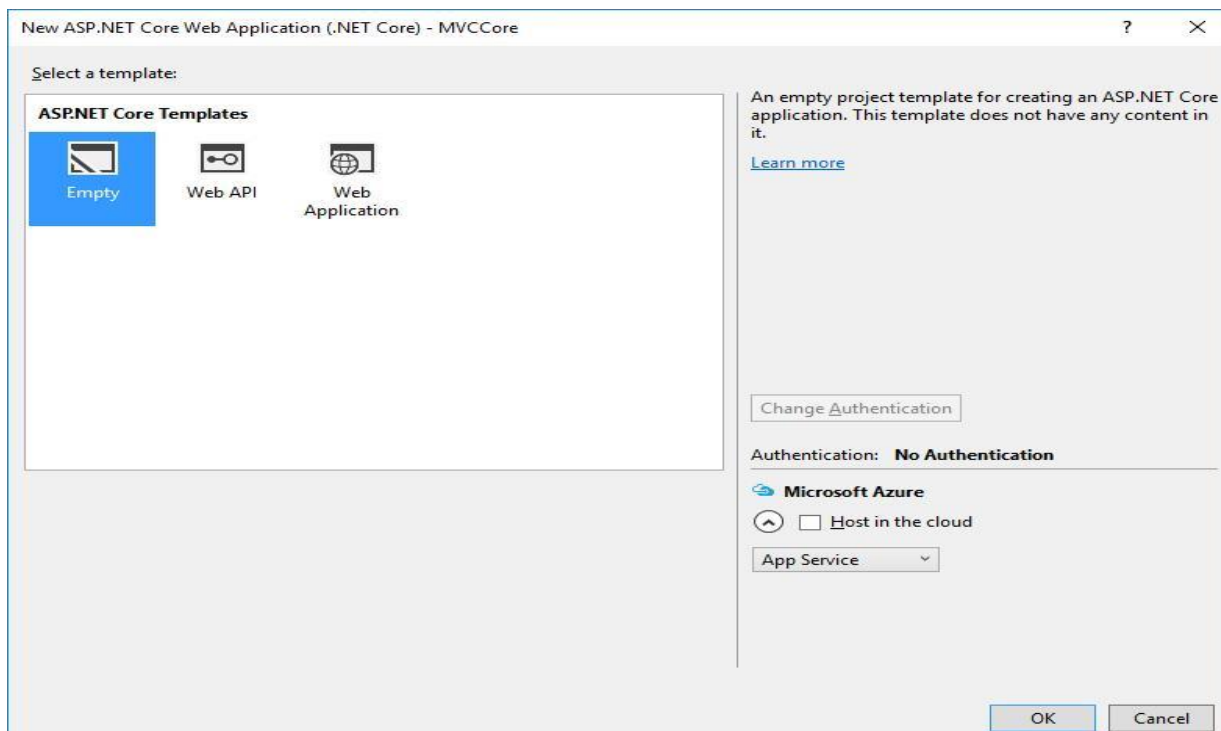
مراحل تکمیل پروژه

ویژوال استودیو را اجرا می کنیم.

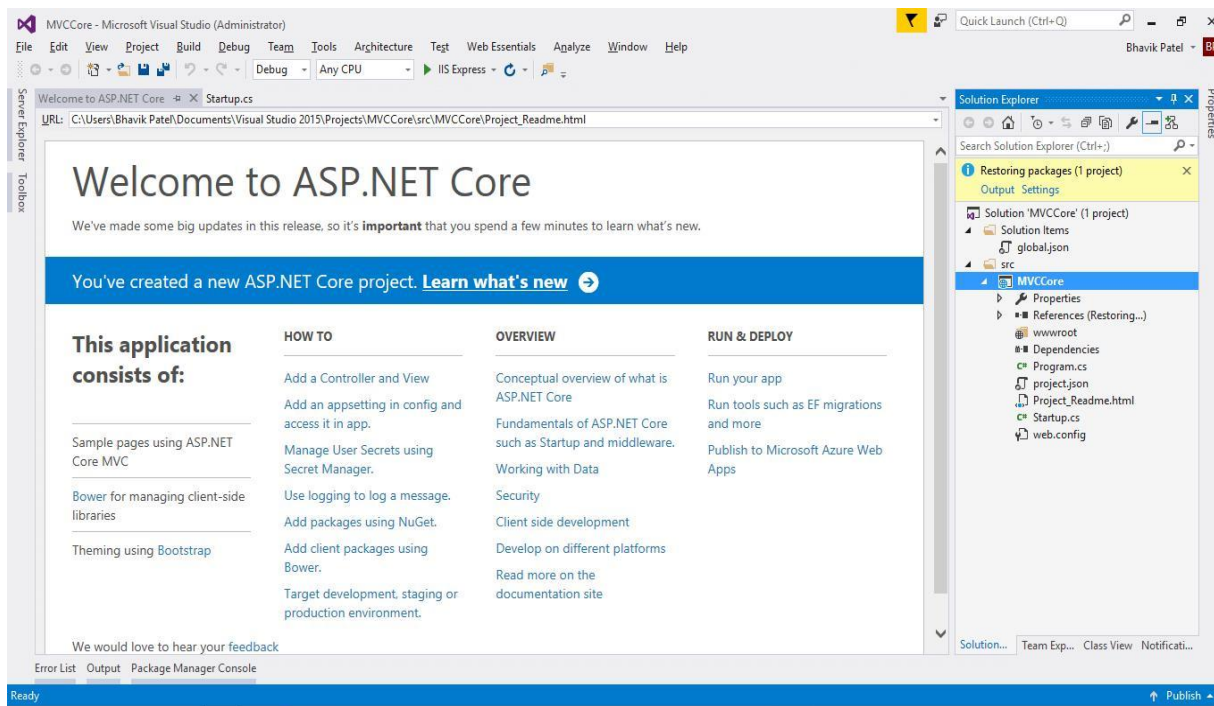
یک پروژه از نوع ASP.NET Core ایجاد می کنیم.



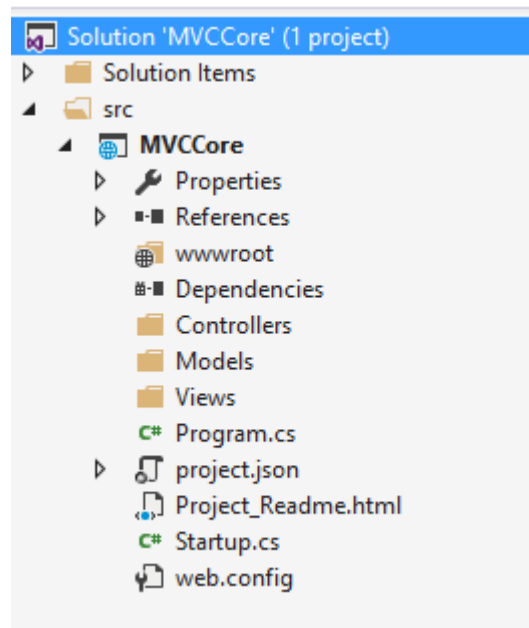
در این قسمت Empty را انتخاب می کنیم .



پس از چند دقیقه به پروژه خالی به صورت خودکار ایجاد می شود.



به طور پیش فرض ، پروژه خالی نیاز به اضافه کردن چند Component دارد . ساختار پروژه جدید تشکیل شده از `wwwroot` , `project.json` , `Global.json` , `Program.cs` , `Startup.cs` است .

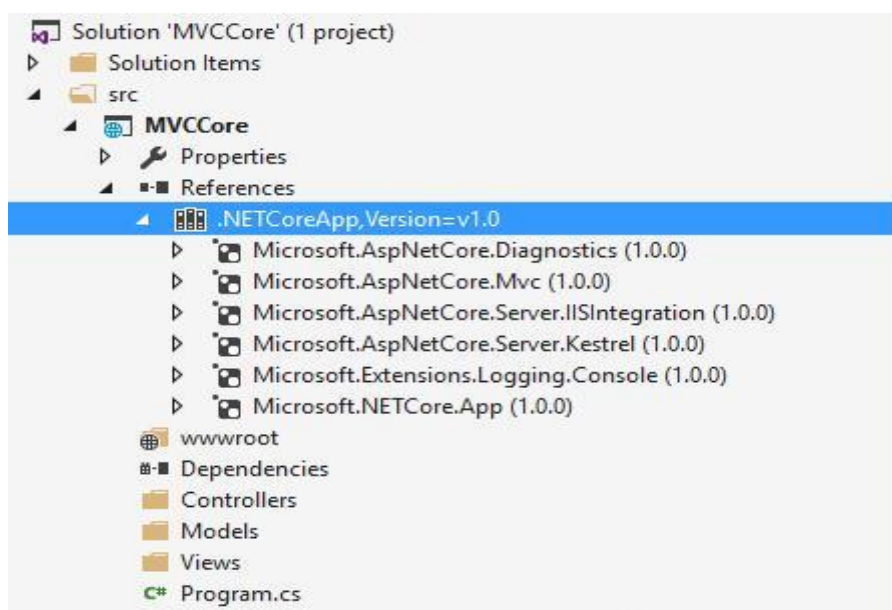


ساختار پروژه جدید تشکیل شده از , Program.cs , Global.json , project.json , wwwroot , Startup.cs است .

برای پیاده سازی MVC، باید پوشه های Views , Models , Controllers را ایجاد کنیم اما این کافی نیست باید MVC middleware نیز به ASP.NET Core اضافه کنیم . برای این کار ابتدا Microsoft.AspNetCore.Mvc را از طریق Nuget بر روی پروژه نصب می کنیم . اگر به فایل project.json را باز کنید ، مشاهده می کنید که به پروژه اضافه شده است .

```
"dependencies": {
  "Microsoft.NETCore.App": {
    "version": "1.0.0",
    "type": "platform"
  },
  "Microsoft.AspNetCore.Diagnostics": "1.0.0",
  "Microsoft.AspNetCore.Mvc": "1.0.0",
  "Microsoft.AspNetCore.Server.IISIntegration": "1.0.0",
  "Microsoft.AspNetCore.Server.Kestrel": "1.0.0",
  "Microsoft.Extensions.Logging.Console": "1.0.0"
}
```

بعد از افزودن MVC به Project.Json، ویژوال استودیو به صورت خودکار ، آن را Restore می کند .



پی‌کریبندی ، pipeline برنامه

در ASP.NET Core میتوان، درخواست pipeline خود را با استفاده از Middleware ایجاد کنیم ASP.NET Core Middleware منطق ناهمزمان را بر روی HttpContext اجرا می‌کند و سپس Middleware بعدی یا درخواست پایانی را فراخوانی می‌کند .

در کلاس Startup.cs متد ConfigureServices با استفاده از Service فریم ورک MVC را اضافه می‌کنیم .

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc();
}
```

Middleware Configure را در درخواست تعریف میکند. به طور پیش فرض ASP.Net Core، هیچ Route ای برای برنامه در نظر نمی‌گیرد. با استفاده از UseMvc میتوان یک Route برای برنامه تنظیم کنید.

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory
loggerFactory)
{
    loggerFactory.AddConsole();

    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }

    app.UseMvc(m =>
    {
        m.MapRoute(
            name: "default",
            template: "{controller}/{action}/{id?}",
            defaults: new { controller = "Home", action = "Index" });
    });
}
```

افزودن Controller

برای این کار بر روی پوشه Controllers کلیک راست کرده و یک Controller اضافه می کنیم .

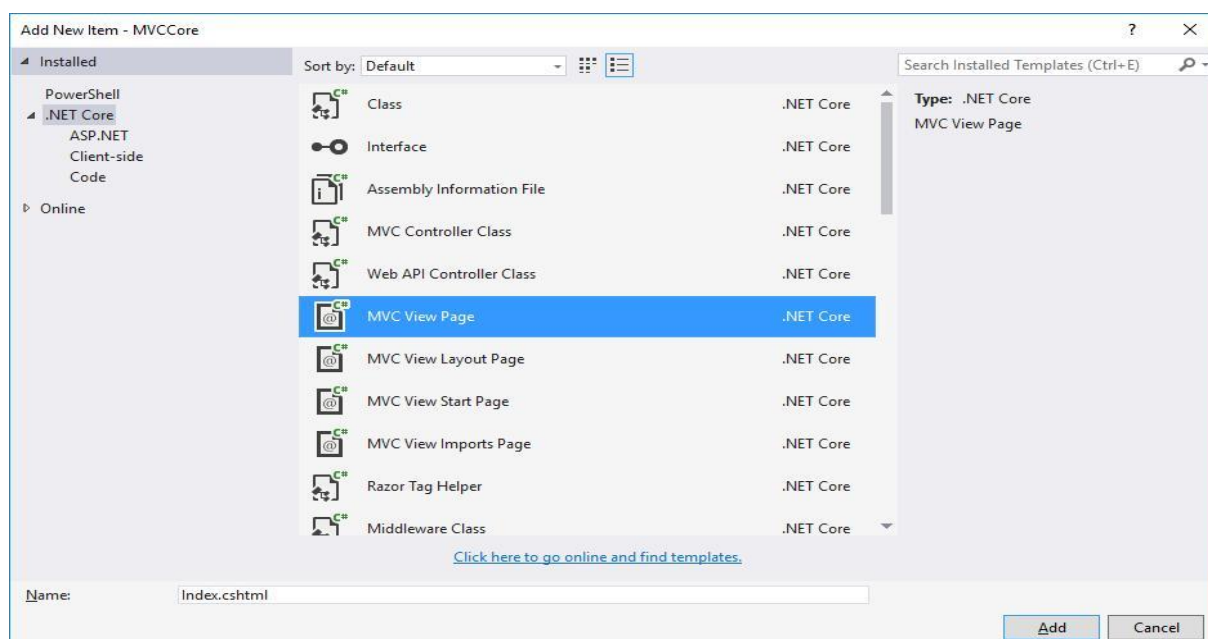
ویژوال استودیو به طور پیش فرض یک action به نام Index ایجاد می کند . بعد از آن یک نمونه از `List<string>`

تعریف می کنیم و چند نام به آن اضافه می کنیم . در آخر آن را با استفاده از Return ارسال می کنیم .

```
public IActionResult Index()
{
    List<string> nameList = new List<string>()
    {
        "Iman",
        "Sajad",
        "reza"
    };
    return View(nameList);
}
```

افزودن View

برای افزودن View بر روی پوشه Views کلیک راست کرده و یک پوشه اضافه می کنیم و نام آن را Home قرار می دهیم و داخل آن یک View ایجاد می کنیم.



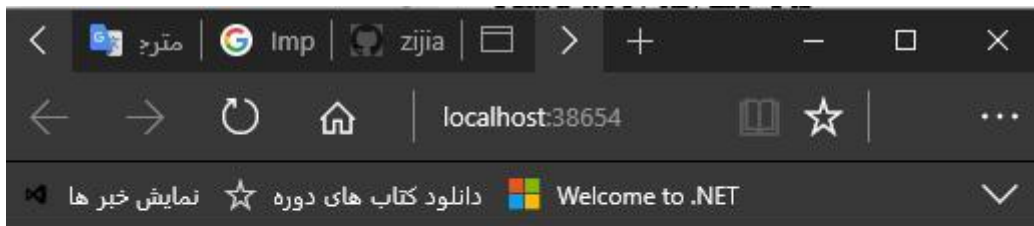
همان طور که ما در حال Return کردن `List<string>` از Controller هستیم ، `List<string>` را به عنوان `@model` دریافت می کنیم و آنها را به صورت یک لیست نمایش می دهیم .

```
@model List<string>
```

```
<h1>کاربران</h1>
```

```
@foreach (var name in Model)
{
    <li>
        @(name)
    </li>
}
```

خروجی برنامه



کاربران

- Iman
- Sajad
- reza