

1. Record Nr.	UNINA9910254856503321
Autore	Japikse Philip
Titolo	Building Web Applications with Visual Studio 2017 : Using .NET Core and Modern JavaScript Frameworks // by Philip Japikse, Kevin Grossnicklaus, Ben Dewey
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2017
ISBN	9781484224786 1484224787
Edizione	[1st ed. 2017]
Descrizione fisica	1 online resource (418 pages) : illustrations
Disciplina	006.7882
Soggetti	Microsoft software Microsoft .NET Framework Programming languages (Electronic computers) Microsoft and .NET Programming Languages, Compilers, Interpreters
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di bibliografia	Includes bibliographical references.
Nota di contenuto	Part I - VS 2017 and .NET Core -- 1. Introducing Entity Framework Core -- 2. Building the Data Access Layer with Entity Framework Core -- 3. Building the RESTful Service with ASP .NET Core MVC Services -- 4. Introducing ASP .NET Core MVC Web Applications -- 5. Building the SpyStore Application with ASP .NET Core MVC -- Part II- JavaScript Frameworks -- 6. JavaScript Application Tools -- 7. Introduction to TypeScript -- 8. Angular 2 -- 9. React.
Sommario/riassunto	Learn how to build web applications from three Microsoft MVPs. After building the data application layer using Entity Framework Core and a RESTful service using ASP.NET Core, you will then build the client side web application three ways: first, using ASP.NET Core, then using Angular 2, and, finally, using React. You will be able to compare and contrast these UI frameworks and select the best one for your needs. .NET Core is a complete rewrite of the popular .NET and its related frameworks. While many concepts are similar between .NET Core and the .NET 4.6 framework, there are revolutionary changes as well, including updates to Entity Framework Core and ASP.NET Core. The

first section of this book covers the three main parts of building applications with C#: Entity Framework, ASP.NET Core Services, and ASP.NET Core Web Applications. There is also an explosion in popularity of JavaScript frameworks for client side development, and the authors cover two of the most popular UI frameworks. Start with TypeScript for developing clean JavaScript, along with a client side build tool such as Gulp, Grunt, and WebPack. Using the same data access layer and RESTful service from the .NET Core application, you can rebuild the UI using Angular 2. Then, repeat the process using React, for a true comparison of building client side applications using ASP.NET Core, Angular 2, and React. What You'll Learn: Understand the fundamentals of .NET Core and what that means to the traditional .NET developer Build: a data access layer with Entity Framework Core, a RESTful service with ASP.NET Core MVC, and a website with ASP.NET Core MVC and Bootstrap Automate many build tasks with client side build utilities.
