

1. Record Nr.	UNINA9910300759703321
Autore	Reynders Fanie
Titolo	Modern API Design with ASP.NET Core 2 [[electronic resource]] : Building Cross-Platform Back-End Systems // by Fanie Reynders
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2018
ISBN	1-4842-3519-3
Edizione	[1st ed. 2018.]
Descrizione fisica	1 online resource (245 pages)
Disciplina	005.2768
Soggetti	Microsoft software Microsoft .NET Framework Microsoft and .NET
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	1. API Design Demystified -- 2. Introduction to ASP.NET Core -- 3. Your First API Application -- 4. Extensibility Architecture -- 5. MVC -- 6. The Configuration Model -- 7. Logging and Error Handling -- 8. Securing APIs -- 9. Bells & Whistles -- 10. Testing & Debugging -- 11. Hosting & Deployment.
Sommario/riassunto	Use ASP.NET Core 2 to create durable and cross-platform web APIs through a series of applied, practical scenarios. Examples in this book help you build APIs that are fast and scalable. You'll progress from the basics of the framework through to solving the complex problems encountered in implementing secure RESTful services. The book is packed full of examples showing how Microsoft's ground-up rewrite of ASP.NET Core 2 enables native cross-platform applications that are fast and modular, allowing your cloud-ready server applications to scale as your business grows. Major topics covered in the book include the fundamentals and core concepts of ASP.NET Core 2. You'll learn about building RESTful APIs with the MVC pattern using proven best practices and following the six principles of REST. Examples in the book help in learning to develop world-class web APIs and applications that can run on any platform, including Windows, Linux, and MacOS. You can even deploy to Microsoft Azure and automate your delivery by implementing Continuous Integration and Continuous Deployment pipelines. What You Will Learn: Incorporate automated API tooling such as Swagger

from the OpenAPI specification Standardize query and response formats using Facebook's GraphQL query language Implement security by applying authentication and authorization using ASP.NET Identity Ensure the safe storage of sensitive data using the data protection stack Create unit and integration tests to guarantee code quality.
