# Instructions

Make sure that all the sources you’ve used in your paper are listed on the reference page. Place the reference page on a new page, right after the body text, but before any appendices.

The Scribbr Citation Generator already applied the APA format to your reference list, meaning:

* The page starts with the section label “References” (bold and centered)
* References are ordered [alphabetically](https://www.scribbr.com/apa-style/ordering-references/?utm_source=citation-generator&utm_medium=word-export)
* The text is double-spaced
* A hanging indent of ½ inch is applied
* Page numbering in the top-right corner

Still have questions? Check out Scribbr’s [article on formatting the reference page](https://www.scribbr.com/apa-style/apa-reference-page/?utm_source=citation-generator&utm_medium=word-export).

Tip: don’t forget to check your [in-text citations](https://www.scribbr.com/apa-style/in-text-citation/?utm_source=citation-generator&utm_medium=word-export) for accuracy. Need a little help? The [Scribbr Citation Checker](https://www.scribbr.com/citation/checker/?utm_source=citation-generator&utm_medium=word-export) can automatically analyze your in-text citations for stylistic errors and inconsistencies, presenting you with simple instructions that help fix them.

# References

*A better domain events pattern · Los Techies*. (2014, May 13). https://lostechies.com/jimmybogard/2014/05/13/a-better-domain-events-pattern/

*Aggregate Pattern*. (n.d.). DevIQ. https://deviq.com/domain-driven-design/aggregate-pattern/

Ait-Aoudia, M. (2014, June 27). *Specification and Notification Patterns*. CodeProject. https://www.codeproject.com/Tips/790758/Specification-and-Notification-Patterns

Amendum, S. J., Vernon-Feagans, L., & Ginsberg, M. C. (2011). The Effectiveness of a Technologically Facilitated Classroom-Based Early Reading Intervention. *The Elementary School Journal*, *112*(1), 107–131. https://doi.org/10.1086/660684

Avram, A., & Marinescu, F. (2007). *Domain-Driven Design Quickly* (null). Lulu.com.

Bahrami, M. (2015). Cloud Computing for Emerging Mobile Cloud Apps. *2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*. https://doi.org/10.1109/mobilecloud.2015.40

Baptista, G., & Abbruzzese, F. (2022). *Software Architecture with C# 10 and .NET 6: Develop software solutions using microservices, DevOps, EF Core, and design patterns for Azure, 3rd Edition* (3rd ed.). Packt Publishing.

Bellemare, A. (2020). *Building Event-Driven Microservices: Leveraging Organizational Data at Scale* (1st ed.). O’Reilly Media.

Blane, E. (2022). *Microsoft Azure Basics for Beginners: Getting the Most Out of Microsoft Cloud Computing Software with Illustrative Images*. Independently published.

Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Conallen, J., & Houston, K. A. (2007). *Object-Oriented Analysis and Design with Applications* (3rd ed.). Addison-Wesley Professional.

*Clarified CQRS*. (2009, December 9). https://udidahan.com/2009/12/09/clarified-cqrs/

*CQRS with MediatR and AutoMapper · Los Techies*. (2015, May 5). https://lostechies.com/jimmybogard/2015/05/05/cqrs-with-mediatr-and-automapper/

Dashorst, M., & Hillenius, E. (2008). *Wicket in Action* (1st ed.). Manning.

*Domain Events – Take 2*. (2008, August 25). https://udidahan.com/2008/08/25/domain-events-take-2/

*Domain Model Validation*. (2023, January 7). https://colinjack.blogspot.com/2008/03/domain-model-validation.html

Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software* (1st ed.). Addison-Wesley Professional.

Fowler, M. (n.d.-a). *bliki: CQRS*. martinfowler.com. https://martinfowler.com/bliki/CQRS.html

Fowler, M. (n.d.-b). *bliki: ValueObject*. martinfowler.com. https://martinfowler.com/bliki/ValueObject.html

Fowler, M. (n.d.-c). *Replacing Throwing Exceptions with Notification in Validations*. martinfowler.com. https://martinfowler.com/articles/replaceThrowWithNotification.html

Fowler, M. (2012). *Fowler: Pattern Enterpr Applica Arch*. Pearson Education.

Gorodinski, L. (2012, May 19). *Validation in Domain-Driven Design (DDD) - Lev Gorodinski*. http://gorodinski.com/blog/2012/05/19/validation-in-domain-driven-design-ddd/

Grybniak, S. (2017, January 6). *Domain-Driven Design: Tactical Design Patterns. Part 2*. CodeProject. https://www.codeproject.com/Articles/1164363/Domain-Driven-Design-Tactical-Design-Patterns-Part

Hanson, C., & Sussman, G. J. (2021). *Software Design for Flexibility: How to Avoid Programming Yourself into a Corner*. The MIT Press.

Hoffman, K. (2017). *Building Microservices with ASP.NET Core: Develop, Test, and Deploy Cross-Platform Services in the Cloud* (1st ed.). O’Reilly Media.

*How to create fully encapsulated Domain Models*. (2008a, February 29). https://udidahan.com/2008/02/29/how-to-create-fully-encapsulated-domain-models/

*How to create fully encapsulated Domain Models*. (2008b, February 29). https://udidahan.com/2008/02/29/how-to-create-fully-encapsulated-domain-models/

Hunter, H. H. (2022, November 23). *Azure Application Insights Snapshot Debugger for .NET apps - Azure Monitor*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/azure-monitor/snapshot-debugger/snapshot-debugger

*Infrastructure Ignorance*. (n.d.). Ayende @ Rahien. https://ayende.com/blog/3137/infrastructure-ignorance

Kodali, R., Behara, G. K., & Govindarajulu, S. N. (2020). *Developing Cloud Native Applications in Azure using .NET Core: A Practitioner’s Guide to Design, Develop and Deploy Apps (English Edition)*. BPB Publications.

Kronquist, J. (2013, June 20). *Post Author: Jan Kronquist*. Blog. https://blog.jayway.com/2013/06/20/dont-publish-domain-events-return-them/

Kumar, V., & Agnihotri, K. (2021). *Serverless Computing Using Azure Functions: Build, Deploy, Automate, and Secure Serverless Application Development with Azure Functions (English Edition)*. BPB Publications.

Martin, R. (2017). *Clean Architecture: A Craftsman’s Guide to Software Structure and Design (Robert C. Martin Series)* (1st ed.). Pearson.

Millett, S., & Tune, N. (2015a). *Patterns, Principles, and Practices of Domain-Driven Design* (1st ed.). Wrox.

Millett, S., & Tune, N. (2015b). *Patterns, Principles, and Practices of Domain-Driven Design* (1st ed.). Wrox.

Newman, S. (2021). *Building Microservices: Designing Fine-Grained Systems* (2nd ed.). O’Reilly Media.

*P of EAA: Repository*. (n.d.). https://martinfowler.com/eaaCatalog/repository.html

*P of EAA: Separated Interface*. (n.d.). https://www.martinfowler.com/eaaCatalog/separatedInterface.html

*P of EAA: Unit of Work*. (n.d.). https://martinfowler.com/eaaCatalog/unitOfWork.html

*Persistence Ignorance*. (n.d.). DevIQ. https://deviq.com/principles/persistence-ignorance/

*Read Versioning in an Event Sourced System*. (n.d.). Leanpub. https://leanpub.com/esversioning/read

Richardson, C. (2006a). *POJOs in Action: Developing Enterprise Applications with Lightweight Frameworks*. Manning.

Richardson, C. (2006b). *POJOs in Action: Developing Enterprise Applications with Lightweight Frameworks* (1st ed.). Manning.

Richardson, C. (2016, October 3). *Developing Transactional Microservices Using Aggregates, Event Sourcing and CQRS - Part 1*. InfoQ. https://www.infoq.com/articles/microservices-aggregates-events-cqrs-part-1-richardson/

Roberts, M. (n.d.). *Serverless Architectures*. martinfowler.com. https://martinfowler.com/articles/serverless.html

Shah, M., & Shah, C. (2018). *Virtual Machine in Cloud Computing*. LAP LAMBERT Academic Publishing.

*SOLID*. (n.d.). DevIQ. https://deviq.com/principles/solid/

Stenberg, J. (2015, September 27). *Domain Events and Eventual Consistency*. InfoQ. https://www.infoq.com/news/2015/09/domain-events-consistency/

*Strengthening your domain: Domain Events · Los Techies*. (2010, April 8). https://lostechies.com/jimmybogard/2010/04/08/strengthening-your-domain-domain-events/

Torre, C. de la. (2017, February 7). *Domain Events vs. Integration Events in Domain-Driven Design and microservices architectures*. Cesar De La Torre. https://devblogs.microsoft.com/cesardelatorre/domain-events-vs-integration-events-in-domain-driven-design-and-microservices-architectures/

*Validation in a DDD world · Los Techies*. (2009, February 15). https://lostechies.com/jimmybogard/2009/02/15/validation-in-a-ddd-world/

Vasconcellos, P. R., Bezerra, V. M., & Bianchini, C. P. (2018). Applying Event Sourcing in a ERP System: A Case Study. *2018 XLIV Latin American Computer Conference (CLEI)*. https://doi.org/10.1109/clei.2018.00019

Vermeir, N. (2022). *Introducing .NET 6: Getting Started with Blazor, MAUI, Windows App SDK, Desktop Development, and Containers* (1st ed.). Apress.

Vernon, V. (2013). *Implementing Domain-Driven Design* (1st ed.). Addison-Wesley Professional.

Vernon, V. (2014a, October 13). *Modeling Aggregates with DDD and Entity Framework*. Kalele. https://kalele.io/modeling-aggregates-with-ddd-and-entity-framework/

Vernon, V. (2014b, December 8). *The Ideal Domain-Driven Design Aggregate Store?* Kalele. https://kalele.io/the-ideal-domain-driven-design-aggregate-store/

Vernon, V., & Tomasz, J. (2022). *Strategic Monoliths and Microservices: Driving Innovation Using Purposeful Architecture (Addison-Wesley Signature Series (Vernon))* (1st ed.). Addison-Wesley Publishing.

Vernon-Feagans, L., Bratsch-Hines, M., Varghese, C., Cutrer, E. A., & Garwood, J. D. (2018). Improving Struggling Readers’ Early Literacy Skills through a Tier 2 Professional Development Program for Rural Classroom Teachers: The Targeted Reading Intervention. *The Elementary School Journal*, *118*(4), 525–548. https://doi.org/10.1086/697491

Whitesell, S., Richardson, R., & Groves, M. D. (2022). *Pro Microservices in .NET 6: With Examples Using ASP.NET Core 6, MassTransit, and Kubernetes* (1st ed.). Apress.

Wolff, E. (2016). Microservices: Flexible Software Architecture. *Microservices: Flexible Software Architecture. Addison-Wesley Professional*.

Young, A. (2019). *Microservices Patterns: Your Complete Handbook on Building Testable, Scalable, and Maintainable Microservices*. Independently published.

Zimarev, A. (2019). *Hands-On Domain-Driven Design with .NET Core: Tackling complexity in the heart of software by putting DDD principles into practice*. Packt Publishing.