

@agoncal

Understanding Bean Validation

Bean Validation 2.0

Antonio Goncalves

Foreword by Gunnar Morling

Understanding Bean Validation 2.0

Bean Validation

Antonio Goncalves

2021-05-03

Table of Contents

Foreword	3
About the Author	5
Acknowledgments	6
Introduction	8
Where Does This Fascicle Come From?	8
Who Is This Fascicle For?	9
How Is This Fascicle Structured?	9
Conventions	10
The Sample Application	11
Downloading and Running the Code	12
Getting Help	13
Contacting the Author	13
1. First Step with Bean Validation	14
2. Understanding Bean Validation	16
2.1. Understanding Constraints and Validation	16
2.1.1. Application	17
2.1.2. Database	18
2.1.3. Client	18
2.1.4. Interoperability	18
2.2. Bean Validation Overview	18
2.2.1. A Brief History of Bean Validation	19
2.2.2. JCP and Eclipse Foundation	19
2.2.3. Java EE and Jakarta EE	20
2.2.4. Expression Language	20
2.2.5. What's New in Bean Validation 2.0?	20
2.2.6. Implementations	21
3. Getting Started	23
3.1. Developing Your First Bean Validation Application	23
3.1.1. Setting up the Maven Dependencies	24
3.1.2. Applying Constraints	26
3.1.3. Validating Constraints	26
3.1.4. Running the Tests	29
3.2. A Closer Look at Bean Validation	29
3.2.1. Bean Validation Packages	30
3.2.2. Main Bean Validation APIs	30
3.2.3. Main Bean Validation Annotations	32
3.2.4. Deployment Descriptors	32
Validation	33

Constraint Descriptor	34
4. Applying Constraints	36
4.1. Built-in Constraints	36
4.2. Applying Built-in Constraints	37
4.2.1. Constraining Attributes	38
4.2.2. Constraining Containers	40
4.2.3. Constraining Optionals	41
4.2.4. Constraining Methods	42
4.3. Multiple Constraints for the Same Target	42
4.4. Messages	44
4.4.1. Templates	45
4.4.2. Overriding Error Messages	45
4.4.3. Resource Bundles	46
4.5. Deployment Descriptors	47
5. Validating Constraints	50
5.1. Validation APIs	50
5.1.1. Validator API	50
5.1.2. ConstraintViolation API	51
5.1.3. Obtaining a Validator	51
5.2. Validating Beans	52
5.3. Validating Properties	54
5.4. Validating Values	55
5.5. Validating Methods	55
5.5.1. ExecutableValidator API	55
5.5.2. Validating Method Parameters	56
5.6. Cascading Validation	57
6. Writing Constraints	60
6.1. Constraint Definition APIs	60
6.1.1. Constraint Annotation	61
Target	63
6.1.2. ConstraintValidator	63
6.1.3. Validator	65
6.2. Defining Your Own Constraints	66
6.2.1. Constraint Composition	66
6.2.2. Generic Constraint	68
6.3. ConstraintValidator Context	71
6.4. Messages	73
6.4.1. Templates	74
6.5. Class-level Constraints	74
7. Advanced Topics	76
7.1. Constraint Inheritance	76

7.2. Groups	77
7.2.1. Grouping Constraints	77
7.2.2. Validating Groups	79
8. Integrating Bean Validation with Other Technologies	82
8.1. Java Persistence API Integration	82
8.2. Java ServerFaces Integration	84
8.3. JAX-RS Integration	86
8.4. CDI Integration	88
8.5. Spring Integration	90
9. Putting It All Together	94
9.1. Writing the Constraints	95
9.1.1. Writing the Order and OrderLine Beans	96
9.1.2. Writing the Customer Bean	96
9.1.3. Writing the Address Bean	97
9.1.4. Writing the @ZipCode Constraint	97
9.2. Writing the OrderTest Integration Tests	99
9.3. Compiling and Testing with Maven	100
10. Summary	103
Appendix A: Setting up the Development Environment on macOS	104
A.1. Homebrew	104
A.1.1. A Brief History of Homebrew	104
A.1.2. Installing Homebrew on macOS	104
A.1.3. Checking for Homebrew Installation	104
A.1.4. Some Homebrew Commands	105
A.2. Java 11	105
A.2.1. Architecture	105
A.2.2. A Brief History of Java	106
A.2.3. Installing the JDK on macOS	106
A.2.4. Checking for Java Installation	108
A.3. Maven 3.6.x	109
A.3.1. A Brief History of Maven	109
A.3.2. Project Descriptor	109
A.3.3. Managing Artifacts	110
A.3.4. Installing Maven on macOS	111
A.3.5. Checking for Maven Installation	112
A.3.6. Some Maven Commands	112
A.4. Testing Frameworks	113
A.4.1. JUnit 5.x	113
A Brief History of JUnit	113
Writing Tests	113
Executing Tests	116

A.5. Git	118
A.5.1. A Brief History of Git	118
A.5.2. Installing Git on macOS	119
A.5.3. Checking for Git Installation	119
A.5.4. Cloning Repository	119
Appendix B: Bean Validation Specification Versions	120
B.1. Bean Validation 2.0	120
B.2. Bean Validation 1.1	120
B.3. Bean Validation 1.0	120
Appendix C: References	122
Appendix D: Revisions of the Fascicle	123
D.1. 2021-05-03	123
D.2. 2019-05-17	123
D.3. 2018-02-16	124
Appendix E: Resources by the Same Author	125
E.1. Fascicles	125
E.1.1. Understanding Bean Validation 2.0	125
E.1.2. Understanding JPA 2.2	125
E.1.3. Understanding Quarkus 2.x	126
E.1.4. Practising Quarkus 2.x	127
E.2. Online Courses	128
E.2.1. Starting With Quarkus	128
E.2.2. Building Microservices With Quarkus	128
E.2.3. Quarkus: Fundamentals (<i>PluralSight</i>)	129
E.2.4. Microservices: The Big Picture (<i>PluralSight</i>)	129
E.2.5. Java EE: The Big Picture (<i>PluralSight</i>)	130
E.2.6. Java EE: Getting Started (<i>PluralSight</i>)	130
E.2.7. Java EE 7 Fundamentals (<i>PluralSight</i>)	131
E.2.8. Java Persistence API 2.2 (<i>PluralSight</i>)	131
E.2.9. Context and Dependency Injection 1.1 (<i>PluralSight</i>)	132
E.2.10. Bean Validation 1.1 (<i>PluralSight</i>)	132
Appendix F: Printed Back Cover	134

Understanding Bean Validation 2.0

Copyright © 2018-2021 by Antonio Goncalves

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law. For permission requests, write to the publisher, addressed "*Attention: Permissions Coordinator*," at the email address below:

agoncal.fascicle@gmail.com

Trademarked names, logos, and images may appear in this fascicle. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image, we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The distribution of the book is made through Amazon KDP (Kindle Direct Publishing).^[1]

Any source code referenced by the author in this text is available to readers at <https://github.com/agoncal/agoncal-fascicle-bean-validation/tree/2.0>. This source code is available for reproduction and distribution as it uses an MIT licence.^[2]

- www.antonigoncalves.org
- agoncal.teachable.com
- www.amazon.com/author/agoncal

You can find two different formats of this fascicle:

- eBook (PDF/EPUB): <https://agoncal.teachable.com/p/ebook-understanding-bean-validation>
- Paper book: <http://amazon.com/Understanding-Bean-Validation-2-0-fascicle/dp/1980399026>
(ISBN: 9781980399025)

Version Date: 2021-05-03

To my wonderful kids, Eloise, Ligia and Ennio, who are the best thing life has given me.

Foreword

Exactly nine years ago, I received an email which would fundamentally change my professional life fundamentally.

The sender was Emmanuel Bernard, spec lead of Bean Validation at that time, asking me whether I'd be interested in writing the documentation for the Bean Validation reference implementation. I had published a few blog posts on Bean Validation and apparently Emmanuel liked them, so he offered me this job. I felt honoured ("Wow, they read my blog?!"), excited ("Yeah, I'll become a famous open source contributor!") but also a bit scared ("Hmm, can I even do that?"). Without thinking too long, I accepted the challenge and went off to write the first chapters of the Hibernate Validator reference documentation. After a while, I sent in a patch file which was eventually committed to the SVN repository. Boy, was I proud?!

I had no idea that this would be the first step on my path to working full-time on open source and even becoming the Bean Validation spec lead myself one day. But I had learned an important thing: thorough documentation, written in an easy-to-follow style is a vital factor for software to become successful. The challenge lies in hitting the sweet spot of completeness (all the relevant features should be covered) and conciseness (the reader should be able to quickly find the information they're after). The best functionality isn't worth much if potential users cannot easily find out about it.

That's why I was immediately convinced of Antonio's idea of ripping apart his tremendously successful book on Java EE and extracting multiple, smaller fascicles out of it. Each one focuses on one specific API, providing a gentle introduction to it as well as discussing more advanced topics at the same time. It's with great joy that I see that the first of these fascicles is dedicated to Bean Validation.

Based on his extensive experience of using Bean Validation in many Java EE and Spring based projects, Antonio did an outstanding job writing this fascicle. Starting with the basics of putting data validation into the wider context of application development and of setting up your first Bean Validation application, the fascicle touches all the important aspects of Bean Validation such as using built-in and custom constraints, message interpolation, validation groups, method validation and much more. A wide range of examples shows, in depth, how to use the API, and there's even detailed instructions for setting up your development environment, making it very easy to get started.

Since I received that email from Emmanuel in February 2009, Bean Validation has come a long way. Closely integrated with many other specifications and technologies such as JPA, JAX-RS, CDI, JavaFX or Spring, it's successfully used in countless projects. Bean Validation 1.1 added the notion of method validation, making it trivial to automatically validate parameters and return values upon method invocation. Bean Validation 2.0, released in 2017 and part of Java EE 8, brought closer integration with Java 8 and the long-awaited support for validating the elements of any generic container type. How this is done? Find out about this and much more by turning over and diving into this excellent fascicle!

Gunnar Morling

Spec Lead of Bean Validation 2.0 (JSR 380)

Hamburg, February 2018

[1] KDP <https://kdp.amazon.com>

[2] MIT licence <https://opensource.org/licenses/MIT>

About the Author



Antonio Goncalves is a senior software architect living in Paris. Having been focused on Java development since the late 1990s, his career has taken him to many different countries and companies where he now works as a recognised consultant. As a former employee of BEA Systems (acquired by Oracle), he developed a very early expertise on distributed systems. He is particularly fond of open source and is a member of the OSSGTP (Open Source Solution Get Together Paris). Antonio loves to create bonds with the community. So, he created the Paris Java User Group in 2008 and co-created Devovx France in 2012 and Voxxed Microservices in 2018.^[3]

Antonio wrote his first book on Java EE 5, in French, in 2007. He then joined the JCP to become an Expert Member of various JSRs (Java EE 8, Java EE 7, Java EE 6, CDI 2.0, JPA 2.0, and EJB 3.1) and wrote *Beginning Java EE 7* and *Beginning Java EE 8* with Apress.^[4] Still hooked on sharing his knowledge, Antonio Goncalves decided to then self-publish his later fascicles.

For the last few years, Antonio has given talks at international conferences, mainly on Java, distributed systems and microservices, including JavaOne, Devovx, GeeCon, The Server Side Symposium, Jazoon, and many Java User Groups. He has also written numerous technical papers and articles for IT websites (DevX) and IT magazines (Java Magazine, Programmez, Linux Magazine). Since 2009, he has been part of the French Java podcast called Les Cast Codeurs.^[5]

In recognition of his expertise and all of his work for the Java community, Antonio has been elected **Java Champion**.^[6]

Antonio is a graduate of the Conservatoire National des Arts et Métiers in Paris (with an engineering degree in IT), Brighton University (with an MSc in object-oriented design), Universidad del Pais Vasco in Spain, and UFSCar University in Brazil (MPhil in Distributed Systems). He also taught for more than 10 years at the Conservatoire National des Arts et Métiers where he previously studied.

Follow Antonio on Twitter ([@agoncal](https://twitter.com/agoncal)) and on his blog (www.antoniogoncalves.org).

[3] Devovx France <https://devovx.fr>

[4] Amazon <https://www.amazon.com/author/agoncal>

[5] Les Cast Codeurs <https://lescastcodeurs.com>

[6] Java Champions <https://developer.oracle.com/javachampions>

Acknowledgments

In your hands, you have a technical fascicle that comes from my history of writing, learning and sharing. When writing, you need a dose of curiosity, a glimpse of discipline, an inch of concentration, and a huge amount of craziness. And of course, you need to be surrounded by people who help you in any possible way (so you don't get totally crazy). And this is the space to thank them.

First of all, I really want to thank my proofreading team. After the process of writing, I was constantly in contact with Gunnar, Youness, and Guillaume who reviewed the book and gave me precious advice. I have to say, it was a real pleasure to work with such knowledgeable developers.

It is a great honour to have **Gunnar Morling** write the foreword for this fascicle. Gunnar is the Specification Lead for Bean Validation 2.0 (JSR 380) and works as a Principal Software Engineer for Red Hat. An open source enthusiast by heart, he contributes to projects such as Hibernate Validator, Search, ORM and OGM and is the lead of Debezium, a platform for change data capture. He is also the founder of MapStruct, a code generator for bean mappings. Gunnar has spoken at conferences such as DevOxx, JavaOne, JavaZone and many others. He tweets as @gunnarmorling and occasionally blogs at <http://in.relation.to/gunnar-morling>. Gunnar lives and works in Hamburg, Germany.^[7]

Youness Teimouri is a Senior Software Java Developer with over a decade of experience in Java development. He has utilised Java stack to grow numerous companies in a variety of industries such as Telecoms, ERP systems and Mobile Banking. He has co-authored and contributed to some papers on Cloud-Computing and some of my previous books. Youness is fascinated by the endless possibilities of Java in different industries and enjoys mentoring junior developers, inspiring them to develop their own Java skill-set. He lives in Canada.^[8]

Guillaume Smet is a Senior Software Engineer at Red Hat. He works primarily on Hibernate Validator, Hibernate Search and Hibernate OGM. An Open Source contributor since 2001, he has contributed to a lot of Open Source software (PostgreSQL, pgFouine, GForge etc.). In his spare time, he plays volleyball, does yoga and enjoys a good book - preferably from John Irving. Guillaume blogs about Hibernate at <http://in.relation.to/guillaume-smet>. He lives in Lyon, France.^[9]

Thanks to my proofreader, **Gary Branigan**, who added a Shakespearean touch to the fascicle.

I could not have written this fascicle without the help and support of the Java community: blogs, articles, mailing lists, forums, tweets etc.

The fascicle you have in your hands uses a rich AsciiDoctor 2.0.14 toolchain, making it possible to create PDF, EPUB and MOBI files. I am really grateful to the entire AsciiDoctor community, and to Dan Allen and Marat Radchenko in particular, who helped me in sorting out a few things so that the end result looks so great.^[10] PlantUML is an amazing tool with a very rich syntax for drawing diagrams... and sometimes, you need a bit of help. So, thanks to the PlantUML community.^[11] As for the text editor used to write this fascicle, you might have guessed: it's an IDE! Thank you JetBrains for providing me with a free licence for your excellent IntelliJ IDEA.^[12]

Living in Paris, I also have to thank all the bars who have given me shelter so that I could write while drinking coffee and talking to people: La Fontaine, Le Chat Bossu, La Grille, La Liberté and

Bottle Shop.

As you might have guessed, I have a passion for IT. But I have other passions such as science, art, philosophy, cooking... and music (I even play jazz guitar). I cannot work without listening to music, so while I was writing this fascicle, I spent most of my time listening to the best radio ever: FIP.^[13] Thank you FIP.

And a big kiss to my wonderful kids, Eloise, Ligia and Ennio. They are the best present life has given me.

Thank you all!

[7] Gunnar Morling <http://in.relation.to/gunnar-morling>

[8] Youness Teimouri <http://www.youness-teimouri.com>

[9] Guillaume Smet <http://in.relation.to/guillaume-smet>

[10] Ascidoctor <http://ascidoctor.org>

[11] PlantUML <http://plantuml.com>

[12] IntelliJ IDEA <https://www.jetbrains.com/idea>

[13] FIP <https://www.fip.fr>