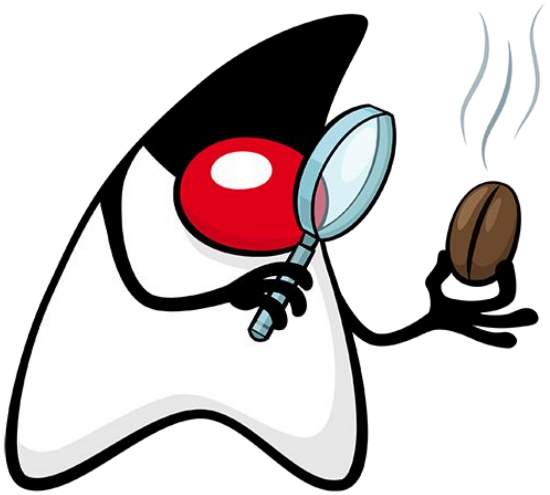
Java Community Process

JSR 380 Public Review Spec Lead Report

May 9 2017

Gunnar Morling, Red Hat



Agenda

- Overview on Bean Validation 2.0
- Working with the expert group and the community
- Questions & discussion

About Bean Validation

- Bean Validation 1.0/1.1
  - Annotation-based API and metamodel for applying constraints to Java objects
  - Java SE and EE
  - Integrated with JPA, CDI, JAX-RS, JSF etc.

- Bean Validation 2.0: Embrace Java SE 8
  - Support API additions
  - Take advantage of language additions

“Constrain once, validate everywhere!”

History

- JSR 303/Bean Validation 1.0: 2009 (Java EE 6)
- JSR 349/Bean Validation 1.1: 2012 (Java EE 7)
- JSR 380/Bean Validation 2.0
  - Submitted mid 2016
  - Early Draft: February 2017
  - Public Review: April 2017
  - Planned Final Submission: July 2017

JSR 380 - Technical scope and features

- Support Java 8 API additions
  - New date + time API
  - java.util.Optional
- Benefit from language additions
  - Type annotations
  - Repeatable annotations
  - Default methods

JSR 380 - Technical scope and features

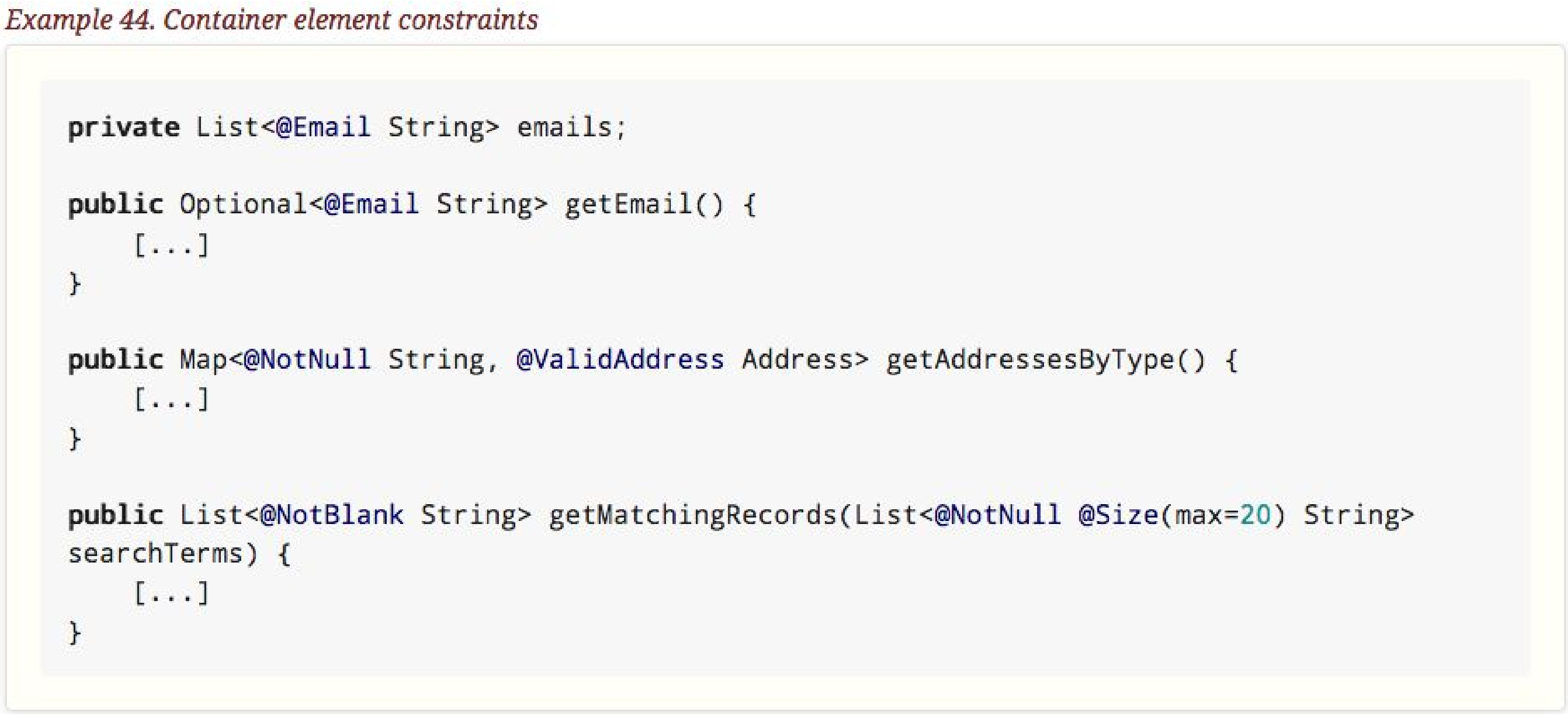
- Validation of container elements
  - `java.util.Iterable`, `Map`
  - `java.util.Optional`
  - Your custom container types

- JSR 310 types supported for `@Past/@Future`

- New built-in constraints
  - `@Email`, `@NotBlank`, `@NotEmpty`
  - `@Positive`, `@Negative`

- Constraints marked with `@Repeatable`

JSR 380 - Technical scope and features



Example 44. Container element constraints

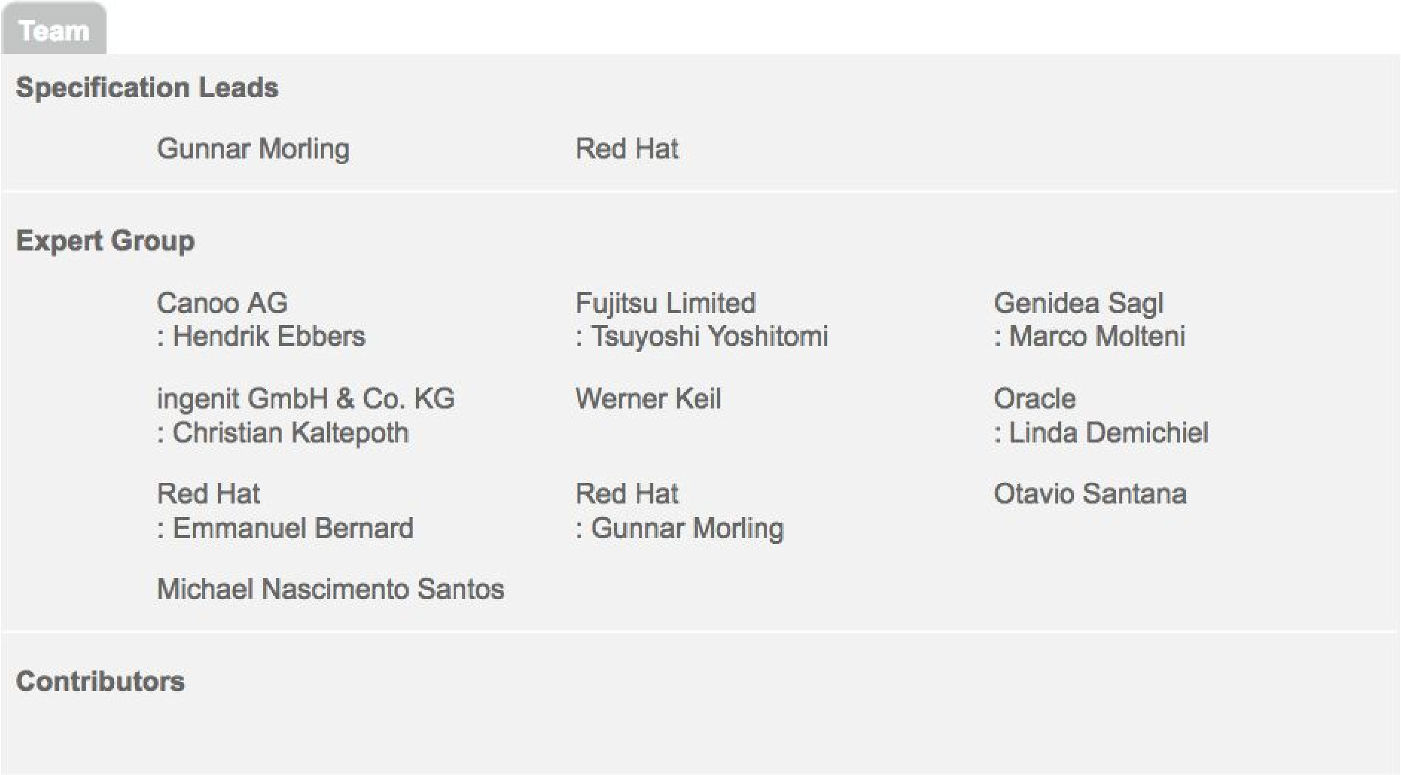
```
private List<@Email String> emails;

public Optional<@Email String> getEmail() {
    [...]
}

public Map<@NotNull String, @ValidAddress Address> getAddressesByType() {
    [...]
}

public List<@NotBlank String> getMatchingRecords(List<@NotNull @Size(max=20) String>
searchTerms) {
    [...]
}
```

The Expert Group



Team

Specification Leads

Gunnar Morling — Red Hat

Expert Group

Canoo AG
: Hendrik Ebbers

Fujitsu Limited
: Tsuyoshi Yoshitomi

Genidea Sagl
: Marco Molteni

ingenit GmbH & Co. KG
: Christian Kaltepoth

Werner Keil

Oracle
: Linda Demichiel

Red Hat
: Emmanuel Bernard

Red Hat
: Gunnar Morling

Otavio Santana

Michael Nascimento Santos

Contributors

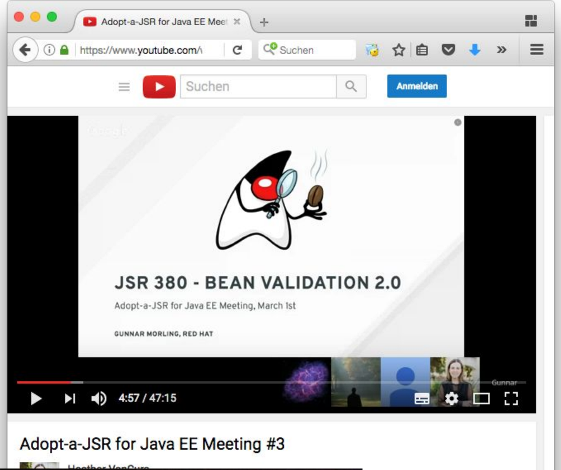
- Experts from JSR 310, JAX-RS, JavaFX, Java EE
- Open for contributions by everyone

The Expert Group - Working together

- Operating the open source way
- Discussing on the mailing list
- GitHub pull requests
- Proposals on the website
- F2F discussions at conferences
- EVERYTHING is open source and public

Publicity

- Sessions at conferences such as Devoxx US, JavaLand, RivieraDev, jdk.io
- “Adopt-a-JSR for Java EE”



JSR 380 - BEAN VALIDATION 2.0

Adopt-a-JSR for Java EE Meeting, March 1st

GUNNAR MORLING, RED HAT

4:57 / 47:15

Adopt-a-JSR for Java EE Meeting #3

166 Aufrufe

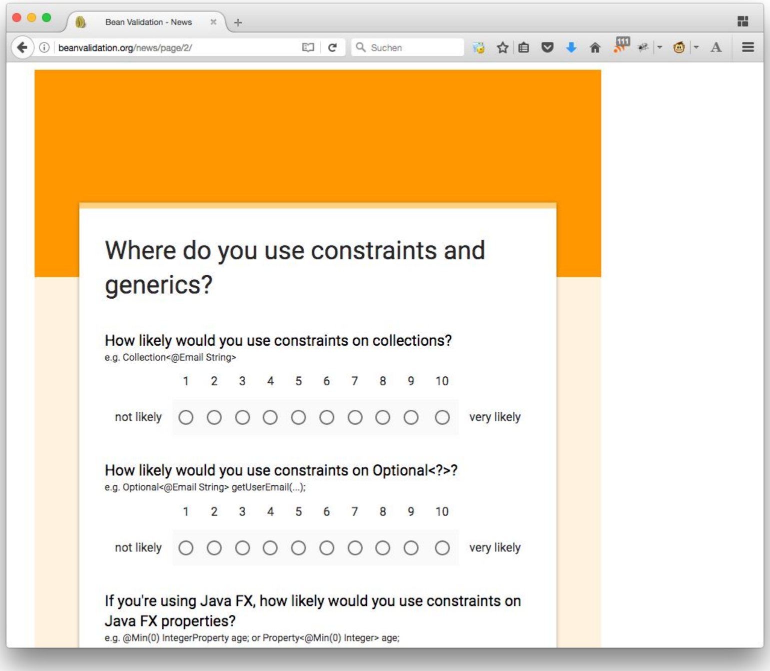
Rs being developed through the ar directly from Spec Leads the dation (JSR 380) JSRs. We will

Status

- Spec: Early Draft
- Referenzimplementierung: Hibernate Validator 6.0 Alpha2
  - Feedback erwünscht :-)
- Public Draft: April
- Bestandteil von Java EE 8 (Juli 2017)

Publicity

- Community surveys on http://beanvalidation.org, asking for feedback on Twitter
- http://javademo.io/bv-list-email



Where do you use constraints and generics?

How likely would you use constraints on collections?
e.g. Collection<@Email String>

1 2 3 4 5 6 7 8 9 10

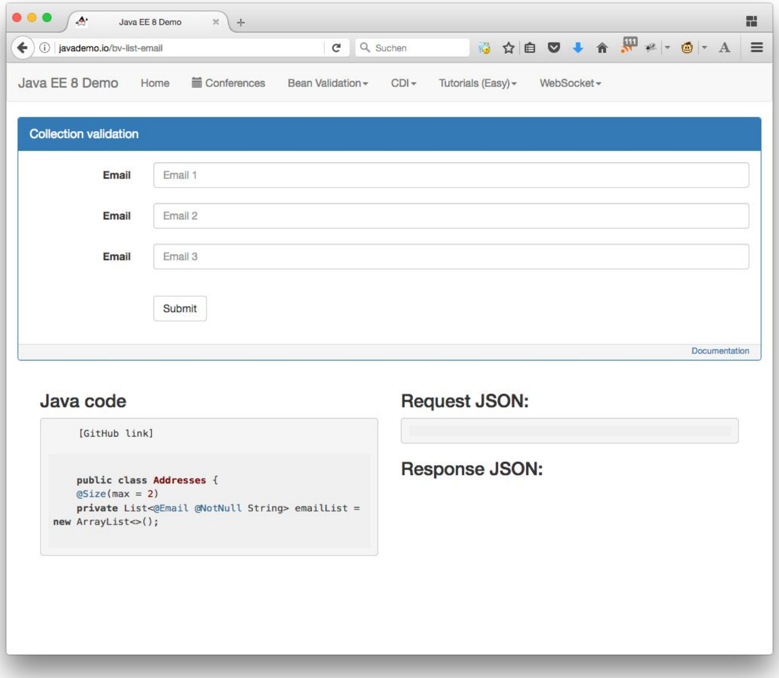
not likely ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ very likely

How likely would you use constraints on Optional<?>?
e.g. Optional<@Email String> getUserEmail(...);

1 2 3 4 5 6 7 8 9 10

not likely ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ very likely

If you're using Java FX, how likely would you use constraints on Java FX properties?
e.g. @Min(0) IntegerProperty age; or Property<@Min(0) Integer> age;



Collection validation

Email: Email 1

Email: Email 2

Email: Email 3

Java code

```
    [GitHub link]

    public class Addresses {
    @Size(max = 2)
    private List<@Email @NotNull String> emailList =
new ArrayList<>();
```

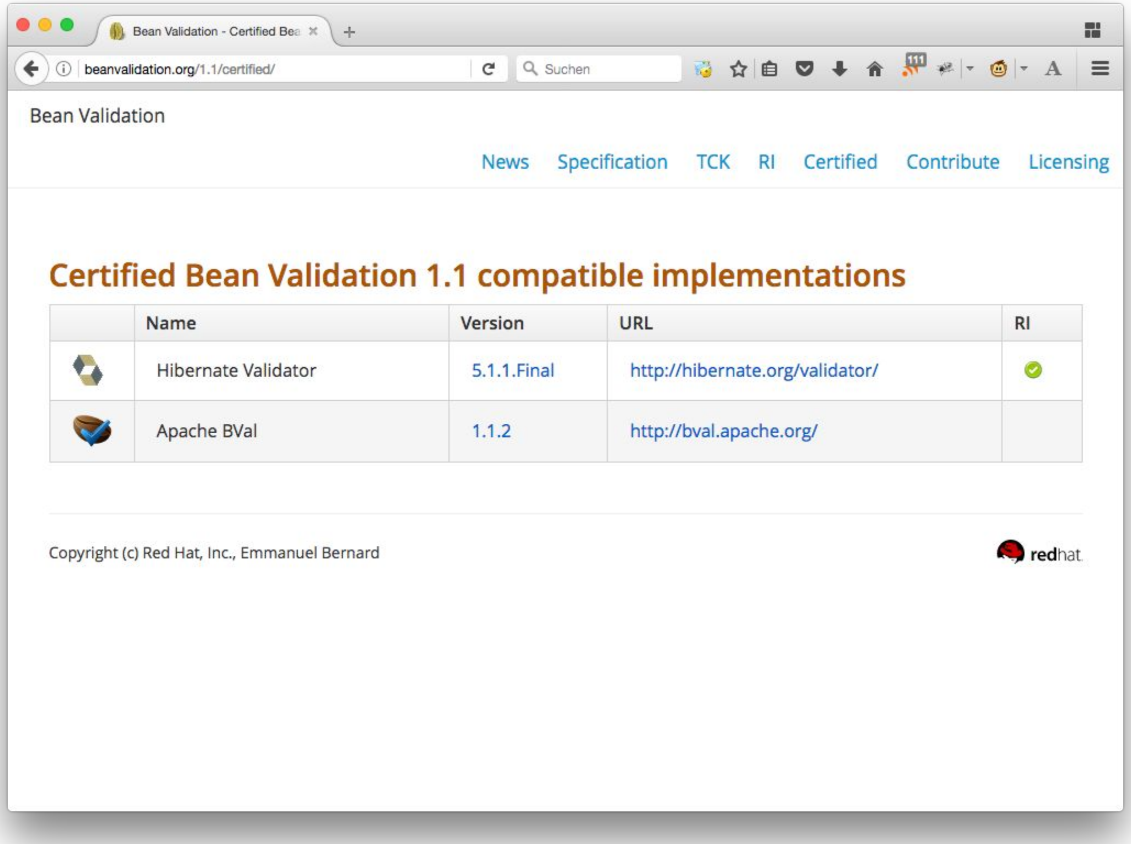
Request JSON:

Response JSON:

Collaboration with other community groups

- JSRs
  - One of JSR 310 (Java Time) leads is part of EG
  - Collaborating with JSR 370 lead (JAX-RS 2.1)
- Reached out to Ceylon team
- Fixed an integration issue with the Java EE platform spec
- JavaFX

Implementations


Bean Validation - Certified Bea
beanvalidation.org/1.1/certified/
Suchen

Bean Validation

News Specification TCK RI Certified Contribute Licensing

Certified Bean Validation 1.1 compatible implementations

	Name	Version	URL	RI
	Hibernate Validator	5.1.1.Final	http://hibernate.org/validator/	✓
	Apache BVal	1.1.2	http://bval.apache.org/	

Copyright (c) Red Hat, Inc., Emmanuel Bernard

redhat

- Hibernate Validator 6: RI for Bean Validation 2.0

Schedule

- Public Review until May 27th
- Public Review Ballot May 30th - June 12th
- Proposed Final Draft: Second half of June
- Submission for Final Approval Ballot July
  - Ship with Java EE 8

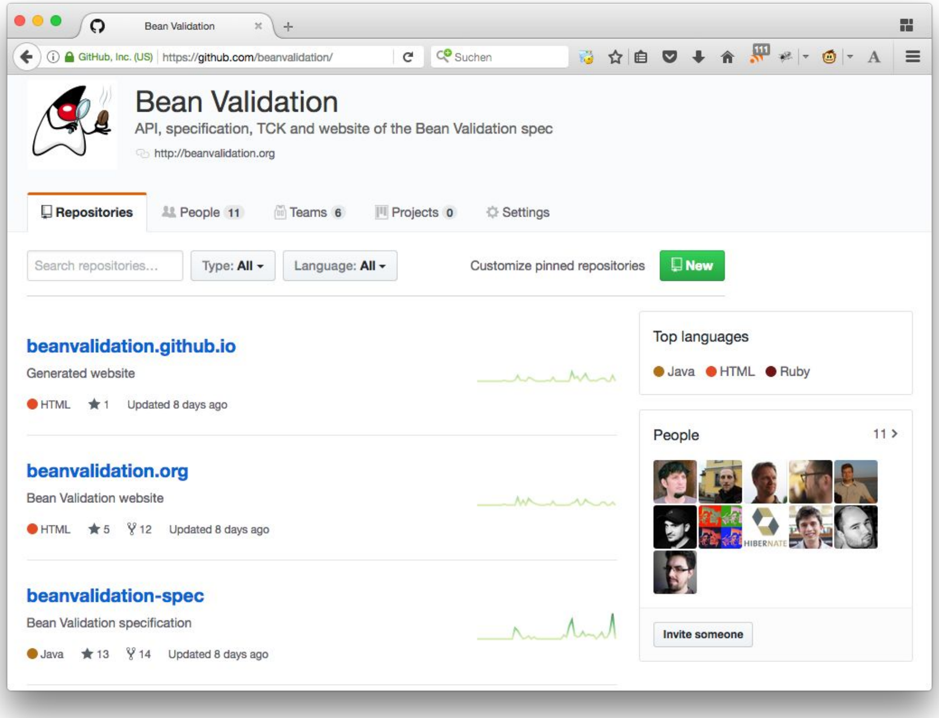
IP flow

- Bean Validation uses Dual Licensing since 1.0
  - Specification, API, TCK, RI provided under the Apache Software License 2.0
  - Specification, API, JavaDoc additionally available under the JCP click through license

- Open for contributions by everyone
  - Applying CLA

RI and TCK development

- Development fully based on open source principles
- Source code on GitHub
  - http://github.com/beanvalidation/
  - https://github.com/hibernate/hibernate-validator
- Binaries on Maven Central



Bean Validation

API, specification, TCK and website of the Bean Validation spec

http://beanvalidation.org

beanvalidation.github.io
Generated website
● HTML ★ 1 Updated 8 days ago

beanvalidation.org
Bean Validation website
● HTML ★ 5 ⑂ 12 Updated 8 days ago

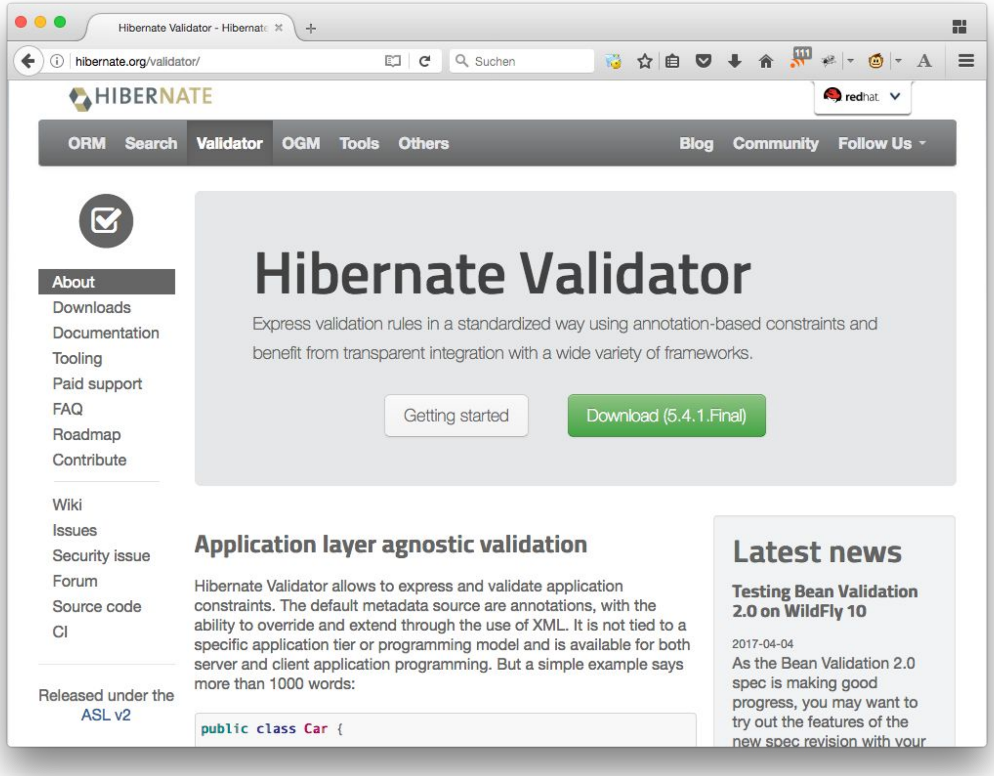
beanvalidation-spec
Bean Validation specification
● Java ★ 13 ⑂ 14 Updated 8 days ago

Top languages
● Java ● HTML ● Ruby

People 11 >

Other deliverables

- Provided by the RI (hibernate.org/validator):
  - Extensive reference guide
  - Example projects
  - FAQ



Released under the ASL v2

Hibernate Validator

Express validation rules in a standardized way using annotation-based constraints and benefit from transparent integration with a wide variety of frameworks.

Application layer agnostic validation

Hibernate Validator allows to express and validate application constraints. The default metadata source are annotations, with the ability to override and extend through the use of XML. It is not tied to a specific application tier or programming model and is available for both server and client application programming. But a simple example says more than 1000 words:

```
public class Car {
```

Latest news

Testing Bean Validation 2.0 on WildFly 10

2017-04-04

As the Bean Validation 2.0 spec is making good progress, you may want to try out the features of the new spec revision with your

Java Community Process

Adopt-a-JSR

- Discussed some questions with Medellin JUG
- Java EE demo built by Marco Molteni

Mailing lists or forums

- Mailing list
  - http://lists.jboss.org/pipermail/beanvalidation-dev/
  - Primary communication channel
  - Open to everyone
  - ~65 threads for BV 2.0
  - ~35 messages per month

- Forum
  - https://forum.hibernate.org/viewforum.php?f=26
  - Not used as much

- Also discussions on GitHub pull requests

Issue tracker

- https://hibernate.atlassian.net/projects/BVAL

- 84 issues for 2.0.x
  - Open: 11
  - In Progress: 0
  - Reopened: 0
  - Resolved: 5
  - Closed: 73

- Website issues handled separately

Document archive

- All spec revisions at http://beanvalidation.org/specification/
  - Drafts
  - Diff to previous versions
  - API diffs

- Original proposals at http://beanvalidation.org/proposals/

Questions & Discussion

Thank you!