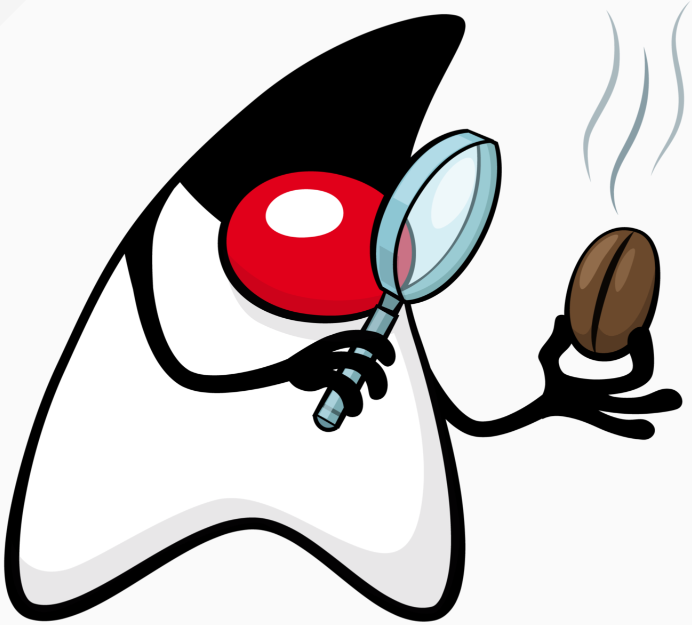


JSR 380 - BEAN VALIDATION 2.0

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

GUNNAR MORLING, RED HAT

BEAN VALIDATION

CONSTRAIN ONCE, VALIDATE EVERYWHERE!

```
public class User {

    @NotNull @Email
    private String email;

    @Min(12)
    private int age;
}
```

- Java SE + EE
- Integrated with JPA, JAX-RS, JSF
- Widely used: Spring MVC, GWT/Vaadin etc.

BEAN VALIDATION 1.1

METHOD VALIDATION

```java
public class UserService {

    @NotNull
    public User createUser(
        @NotNull @Email String email,
        @Min(12) int age) {
        ...
    }
}
```

- Automatically applied upon method invocation

BEAN VALIDATION 2.0

EMBRACE JAVA 8

- Support for java.time API (JSR 310)

```
@Past(orPresent=true)
private Year inceptionYear = Year.of( 2017 );
```

BEAN VALIDATION 2.0

EMBRACE JAVA 8

- Support for java.time API (JSR 310)

```
@Past(orPresent=true)
private Year inceptionYear = Year.of( 2017 );
```

- Repeatable annotations

```
@Size(min = 8, group = Default.class)
@Size(min = 12, group = Admin.class)
private String password = ...;
```

BEAN VALIDATION 2.0

CONSTRAINTS ON CONTAINER ELEMENTS

- java.util.Optional

```
private Optional<String @Email> optionalEmail = ...;
```

BEAN VALIDATION 2.0

CONSTRAINTS ON CONTAINER ELEMENTS

- java.util.Optional

```
private Optional<String @Email> optionalEmail = ...;
```

- Collections

```
@Size(min = 1)
private List<@NotNull @Size(min = 5) String> names;
```

BEAN VALIDATION 2.0

CONSTRAINTS ON CONTAINER ELEMENTS

- java.util.Optional

```
private Optional<String @Email> optionalEmail = ...;
```

- Collections

```
@Size(min = 1)
private List<@NotNull @Size(min = 5) String> names;


private Map<@Valid User, @Valid Address> addresses;
```

BEAN VALIDATION 2.0

CONSTRAINTS ON CONTAINER ELEMENTS

- java.util.Optional

```
private Optional<String @Email> optionalEmail = ...;
```

- Collections

```
@Size(min = 1)
private List<@NotNull @Size(min = 5) String> names;


private Map<@Valid User, @Valid Address> addresses;
```

- Arrays, JavaFX's Property<T>, your own containers!

CURRENT STATUS

- In Early Draft 1
- Reference implementation: Hibernate Validator 6.0 Alpha1
  - Implements Early Draft 1
  - And more, e.g. support for lambda expressions

```
ConstraintMapping constraintMapping = ...;

constraintMapping
    .constraintDefinition( FileExists.class )
        .validateType( File.class )
            .with( f -> f.exists() );
```

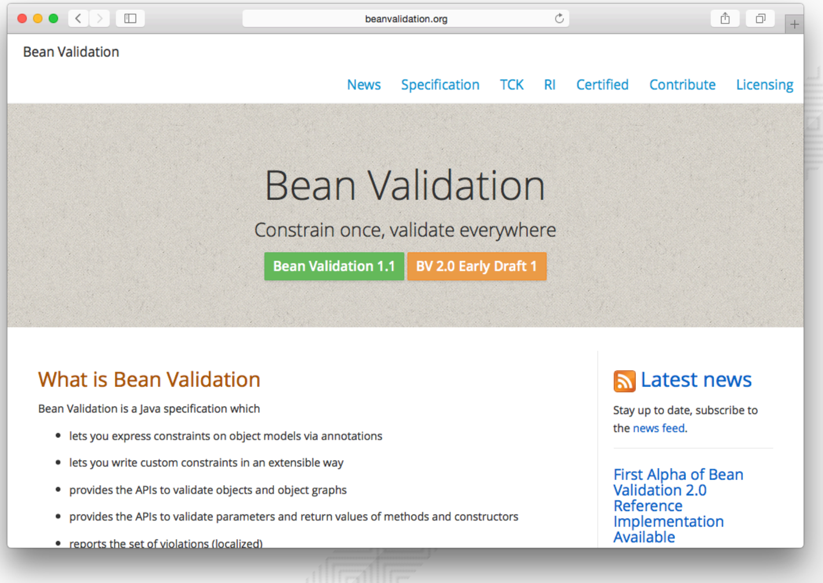
GETTING INVOLVED

- Feedback, feedback, feedback!
  - Try out container element constraints, value extractor API
  - Join our mailing list
  - Report bugs (and features)
- Spread the word
- Let's talk at conferences
  - Devoxx US - March 21 - 23, San Jose
  - JavaLand - March 28-30, Brühl, Germany
- Join the EG

ALL IS OPEN SOURCE

- API, specification, TCK, web site
- https://github.com/beanvalidation/
- Reference implementation
- https://github.com/hibernate/hibernate-validator/
- Apache License 2.0

beanvalidation.org/contribute/



Bean Validation

Constrain once, validate everywhere

What is Bean Validation

Bean Validation is a Java specification which

- lets you express constraints on object models via annotations
- lets you write custom constraints in an extensible way
- provides the APIs to validate objects and object graphs
- provides the APIs to validate parameters and return values of methods and constructors
- reports the set of violations (localized)

Latest news

Stay up to date, subscribe to the news feed.

First Alpha of Bean Validation 2.0 Reference Implementation Available

NEXT STEPS

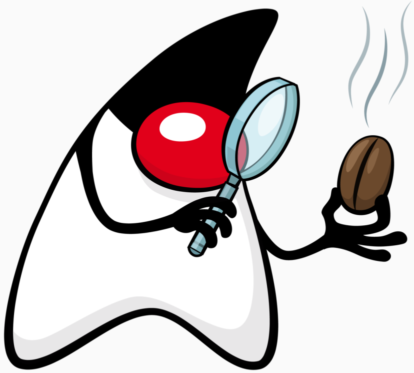
- Converge on container element open issues
- Some new constraints (@NotEmpty, @Email etc.)
- Make message interpolation more re-usable
- YOUR requests

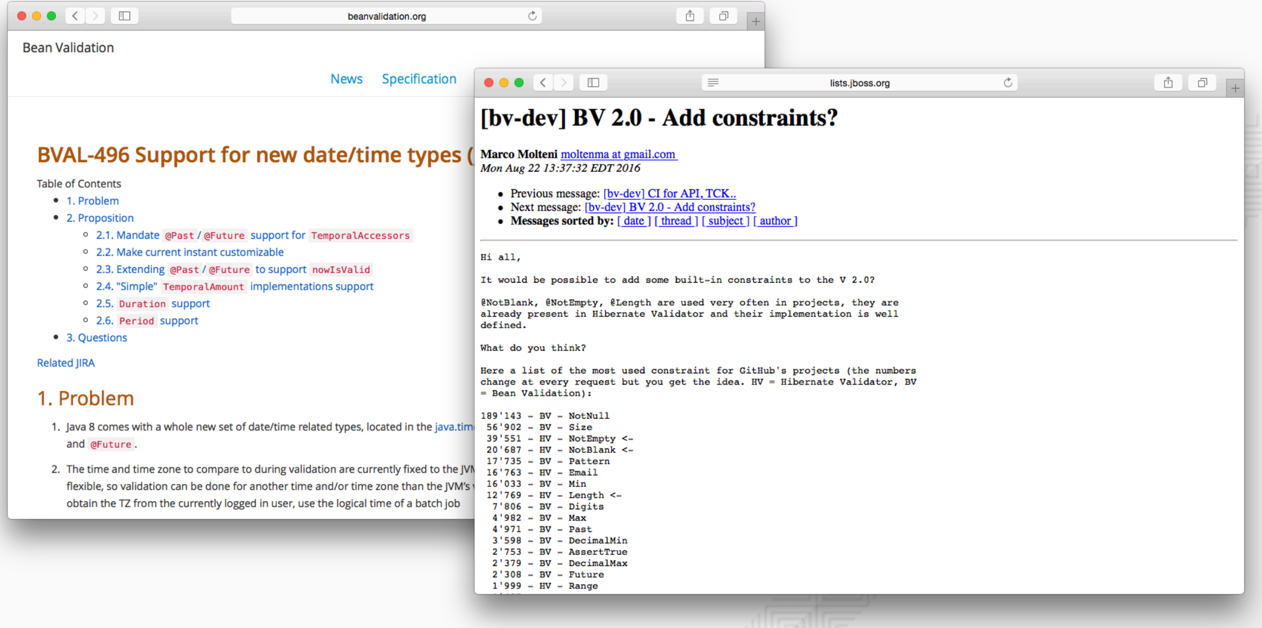
Time line

- (Early draft 2 in next few weeks)
- Public Review in April/May
- Be there for Java EE 8

THANKS TO FANTASTIC EG

- Worked out proposals
- Logo
- Survey
- Feedback
- Much more





Bean Validation

News Specification

BVAL-496 Support for new date/time types (

Table of Contents

- 1. Problem
- 2. Proposition
  - 2.1. Mandate `@Past` / `@Future` support for `TemporalAccessors`
  - 2.2. Make current instant customizable
  - 2.3. Extending `@Past` / `@Future` to support `nowIsValid`
  - 2.4. "Simple" `TemporalAmount` implementations support
  - 2.5. `Duration` support
  - 2.6. `Period` support
- 3. Questions

Related JIRA

1. Problem

1. Java 8 comes with a whole new set of date/time related types, located in the java.tim
   and `@Future`.
2. The time and time zone to compare to during validation are currently fixed to the JVI
   flexible, so validation can be done for another time and/or time zone than the JVM's
   obtain the TZ from the currently logged in user, use the logical time of a batch job

[bv-dev] BV 2.0 - Add constraints?

Marco Molteni moltenma at gmail.com
Mon Aug 22 13:37:32 EDT 2016

- Previous message: [bv-dev] CI for API, TCK..
- Next message: [bv-dev] BV 2.0 - Add constraints?
- **Messages sorted by:** [date] [thread] [subject] [author]

```
Hi all,

It would be possible to add some built-in constraints to the V 2.0?

@NotBlank, @NotEmpty, @Length are used very often in projects, they are
already present in Hibernate Validator and their implementation is well
defined.

What do you think?

Here a list of the most used constraint for GitHub's projects (the numbers
change at every request but you get the idea. HV = Hibernate Validator, BV
= Bean Validation):

189'143 - BV - NotNull
 56'902 - BV - Size
 39'551 - HV - NotEmpty <-
 20'687 - HV - NotBlank <-
 17'735 - BV - Pattern
 16'763 - HV - Email
 16'033 - BV - Min
 12'769 - HV - Length <-
  7'806 - BV - Digits
  4'982 - BV - Max
  4'971 - BV - Past
  3'598 - BV - DecimalMin
  2'753 - BV - AssertTrue
  2'379 - BV - DecimalMax
  2'308 - BV - Future
  1'999 - HV - Range
```

RESOURCES

- Website: beanvalidation.org
- RI: hibernate.org/validator
- Mailing list: http://lists.jboss.org/pipermail/beanvalidation-dev/
- Forum: https://forum.hibernate.org/viewforum.php?f=26
- Issue tracker: https://hibernate.atlassian.net/browse/BVAL

Get in touch:

✉ gunnar@hibernate.org

@gunnarmorling

THANK YOU!