

Im Dutzend billiger

**Darf es eine Assertion mehr sein?
Mit AssertJ zum Testerfolg**

First some questions

AssertJ AssertWhat?

“AssertJ is a Java library that provides a rich set of assertions and truly helpful error messages, improves test code readability, and is designed to be super easy to use within your favorite IDE”

<https://assertj.github.io/doc>

“The idea is that disposal assertions should be specific to the **type** of the objects we are checking when writing unit tests. If you're checking the value of a **String**, you use **String-specific** assertions. Checking the value of a **Map**? Use **Map-specific** assertions to easily check the contents of the map.”


<https://github.com/assertj/assertj-core>

AssertJ Modules

- **Core (version 3.x requires Java 8+)**
- Guava — provides assertions for Guava types, version 3.x requires Java 8+
- Joda Time — provides assertions for Joda Time types, version 2.x requires Java 8+
- DB —to provide assertions for relational database types, requires Java8+
- (Neo4J) — does not seem to be active anymore
- Swing — API for functional testing of Swing user interfaces (less active), <https://github.com/assertj/assertj-swing>
- Vavr — Assertions for Vavr (<https://www.vavr.io/>)

Getting the library

Maven



```
<dependency>
  <groupId>org.assertj</groupId>
  <artifactId>assertj-core</artifactId>
  <version>3.22.0</version>
  <scope>test</scope>
</dependency>
```

Gradle



```
testImplementation("org.assertj:assertj-core:3.22.0")
```

Other

<https://search.maven.org/artifact/org.assertj/assertj-core/3.22.0/bundle>

It starts with *assertThat(<actual>)*

and then it is *fluent*

First example



```
assertThat("Kleine grüne Gummiente")  
    .isNotNull()  
    .isNotEmpty()  
    .isNotBlank()  
    .startsWith("Kleine")  
    .doesNotStartWith("kleine")  
    .endsWith("ente")  
    .hasSize(22);
```

Specialized Assertions for Types

File: isFile, is Directory, canRead, canWrite, hasExtension, hasParent, ...

URI/URL: hasProtocol, hasPath, hasHost, hasPort, hasQuery, hasAnchor, ...

Date/LocalDate: isEqualIgnoringMinutes, isBefore, isAfter, isBetween, ...


Map: containsKey, containsValue, hasEntrySatisfying

Future/CompletableFuture: succeedsWithin

... just to name a few

Configuring AssertJ

org.assertj.core.configuration.Configuration



```
private boolean comparingPrivateFields = true;
private boolean extractingPrivateFields = true;
private boolean bareNamePropertyExtraction = true;
private boolean removeAssertJRelatedElementsFromStackTrace = true;
private boolean lenientDateParsing = false;
private List<DateFormat> additionalDateFormats = Collections.emptyList();
private int maxLengthForSingleLineDescription = 80;
private int maxElementsForPrinting = 1000;
private boolean printAssertionsDescription = false;
private Consumer<Description> descriptionConsumer = null;
private int maxStackTraceElementsDisplayed = 3;
private PreferredAssumptionException preferredAssumptionException;
```

Demo

Descriptions and overriding error messages

Descriptions and overriding error messages



```
assertThat(person.getName( ))  
    .as( "Check person name" )  
    .withFailMessage( "Wrong person name" )  
    .isEqualTo( "Fred" );
```



```
java.lang.AssertionError: [Check person name] Wrong person name
```

Comparison, recursive
comparison

Exceptions

Assumptions

Extending assertions

Soft assertions

Migration

- Using migration scripts
 - JUnit3/4 to AssertJ
 - JUnit5 to AssertJ
 - TestNG to AssertJ
- Manually using regexes (description in AssertJ documentation)
- Using IntelliJ IDEA plugins (i.e. Assertion2AssertJ, Cajon)
- Cajon also helps finding wrong usages of AssertJ and places to refactor

Information, Documentation, License

Documentation: <https://assertj.github.io/doc>

Github: <https://github.com/assertj/assertj-core>

ExampleProject: <https://github.com/assertj/assertj-examples> (not very active)

License: Apache License 2.0

Interesting implementation Details

How to implement a fluent API:

Emulating “self types” using Java Generics to simplify fluent API implementation

<http://bit.ly/1IZIRcY>

Thank you

Beispielcode: <https://github.com/bkratz/AssertJTalk>

- Email: mail@birgitkratz.de
- Web: <https://www.birgitkratz.de>
- Twitter: @bikratz
- Github: <https://github.com/bkratz>