

AssertJ Overview

By Ramesh Fadatare (Java Guides)

AssertJ Overview

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.

When we want to write assertions with AssertJ, we have to use the static **assertThat()** method of the **org.assertj.core.api.Assertions** class.

Spring boot starter test dependency internally provides assertj-core dependency so we don't have to add assertj-core dependency manually in our Spring boot project

Steps to use AssertJ

`import static org.assertj.core.api.Assertions.assertThat;`

```
// basic assertions
assertThat(frodo.getName()).isEqualTo("Frodo");
assertThat(frodo).isNotEqualTo(sauron);

// chaining string specific assertions
assertThat(frodo.getName()).startsWith("Fro")
    .endsWith("do")
    .isEqualToIgnoringCase("frodo");

// collection specific assertions (there are plenty more)
// in the examples below fellowshipOfTheRing is a List<TolkienCharacter>
assertThat(fellowshipOfTheRing).hasSize(9)
    .contains(frodo, sam)
    .doesNotContain(sauron);

// as() is used to describe the test and will be shown before the error message
assertThat(frodo.getAge()).as("check %s's age", frodo.getName()).isEqualTo(33);
```


Why I use AssertJ?

- Easy to learn: Quick start
- Easy to use: you just need to add a dependency and static import in your test class to start using AssertJ.
- Fluent APIs: AssertJ helps you to diversify your assertions.
- More readable code auto-completion: AssertJ provides auto-completion in IDEs. So you don't need to remember all method names.

You can use JUnit 5 assertions

JUnit 5 `assertAll`

JUnit 5 `assertArrayEquals`

JUnit 5 `assertEquals`

JUnit 5 `assertNotEquals`

JUnit 5 `assertNotSame`

JUnit 5 `assertNull` and `assertNotNull`

JUnit 5 `assertSame`

JUnit 5 `assertThrows`

JUnit 5 `assertTimeout`

JUnit 5 `assertFalse`

JUnit 5 `assertTrue`