

Summary of AssertJ Library

AssertJ is a powerful library for writing assertions in Java tests. Here is a quick summary of its features:

1. ****Fluent API****:

- Offers a readable and natural way of writing assertions.
- Example: `assertThat(actual).isEqualTo(expected);`

2. ****Support for Multiple Data Types****:

- Works with Strings, Collections, Maps, Files, Streams, Optionals, and more.
- Example: `assertThat(list).contains("value1", "value2").doesNotContain("value3");`

3. ****Clear Error Messages****:

- Provides detailed error messages to make debugging easier.
- Example: Expecting `<"Hello">` to be equal to `<"World">`.

4. ****Advanced Features****:

- File and Path Assertions: Check if files exist, are readable, or have specific content.
- Optional Assertions: Validate Optional objects seamlessly.
- Example:
`assertThat(optionalValue).isPresent().hasValue("expectedValue");`

5. ****Complex Assertions****:

- Allows combining multiple conditions in one assertion.
 - Example: `assertThat(person).isNotNull().extracting("name", "age").contains("John", 30);`

6. ****Extensibility****:

- Create custom assertions for your own types.
- Example: `assertThat(customObject).hasSpecificProperty("value");`

7. ****Integration****:

- Compatible with JUnit and TestNG.
- Easy to add to existing projects.

8. ****Readable and Scalable****:

- Makes test code cleaner and more maintainable.
- Ideal for both simple and complex testing scenarios.

In summary, AssertJ simplifies the testing process by providing a modern, feature-rich, and extensible assertion framework.