

# Migration from JUnit

## FEST Assertions Module has moved to Github !

Check this page :<https://github.com/alexruiz/fest-assert-2.x/wiki>

The documentation below is for Fest 1.x which is no more maintained, we are focusing our effort to the 2.x version !

....

Here's a list of find/replace based on regexp that allow to change JUnit assertion into FEST assertion (don't forget to check regexp mode in your editor replace window).

A similar idea would be to use the [Structured Search and Replace \(SSR\)](#) feature of IntelliJ IDEA.

Note that FEST & JUnit assertions can coexist together, you don't have to migrate all in one time.

### Changing

**assertEquals(0, myList.size())**

to

**assertThat(myList).isEmpty()**

```
assertEquals(0,(.*).size()); -> assertThat(\1).isEmpty();
```

It's important to run this one before the assertEquals -> isEqualTo, to avoid ending with : `assertThat(myList.size()).isEqualTo(0)`

### Changing

**assertEquals(expectedSize, myList.size())**

to

**assertThat(myList).hasSize(expectedSize)**

```
assertEquals((.*),(.*).size()); -> assertThat(\2).hasSize(\1);
```

It's important to run this one before the assertEquals -> isEqualTo, to avoid ending with : `assertThat(myList.size()).isEqualTo(expectedSize)`

### Changing

**assertEquals(value, valueUnderTest)**

to

**assertThat(valueUnderTest).isEqualTo(value)**

```
assertEquals((.),(.); -> assertThat(\2).isEqualTo(\1);
```

### Changing

**assertNotEquals(value, valueUnderTest)**

to

**assertThat(valueUnderTest).isNotEqualTo(value)**

```
assertNotEquals((.),(.); -> assertThat(\2).isNotEqualTo(\1);
```

### Changing

**assertNull(objectUnderTest)**

to

**assertThat(objectUnderTest).isNull()**

```
assertNull((.*)); -> assertThat(\1).isNull();
```

Changing  
**assertNotNull(objectUnderTest)**

to  
**assertThat(objectUnderTest).isNotNull()**

```
assertNotNull((.*)); -> assertThat(\1).isNotNull();
```

Changing  
**assertTrue(logicalCondition)**

to  
**assertThat(logicalCondition).isTrue()**

```
assertTrue((.*)); -> assertThat(\1).isTrue();
```

Changing  
**assertFalse(logicalCondition)**

to  
**assertThat(logicalCondition).isFalse()**

```
assertFalse((.*)); -> assertThat(\1).isFalse();
```