



KotlinConf

Copenhagen

2024

Kotlin
+ Power-Assert



Brian Norman

Kotlin Compiler Developer



bnorm@kotlin.social

Crafting an Assertion



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertTrue(members.size == 9)  
}
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertTrue(members.size == 9)  
}
```



```
java.lang.AssertionError: Expected value to be true.  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertTrue(members.size == 9)  
}
```

java.lang.AssertionError: Expected value to be true.
at [...]



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertTrue(members.size == 9)  
}
```

✓ Clear assertion condition

```
java.lang.AssertionError: Expected value to be true.  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertTrue(members.size == 9)  
}
```

✓ Clear assertion condition

✗ Useless failure message

```
java.lang.AssertionError: Expected value to be true.  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertEquals(9, members.size)  
}
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertEquals(9, members.size)  
}
```

java.lang.AssertionError: expected:<9> but was:<8>
 at [...]



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertEquals(9, members.size)  
}
```

```
java.lang.AssertionError: expected:<9> but was:<8>  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertEquals(9, members.size)  
}
```

✓ Improved failure message

```
java.lang.AssertionError: expected:<9> but was:<8>  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertEquals(9, members.size)  
}
```

✓ Improved failure message

✗ No intermediate values

```
java.lang.AssertionError: expected:<9> but was:<8>  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertEquals(9, members.size, "Members: $members")  
}
```

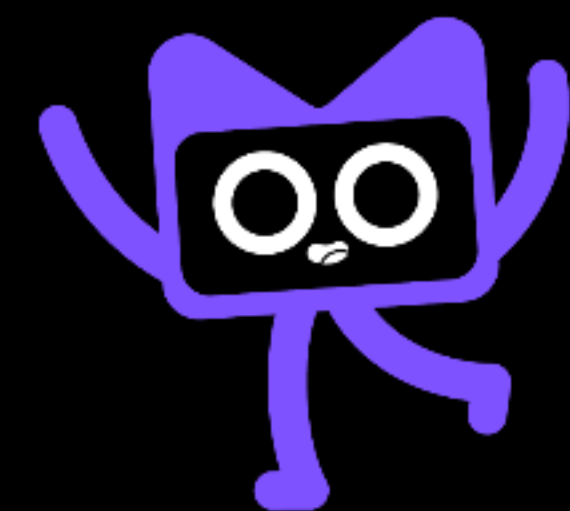


Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertEquals(9, members.size, "Members: $members")  
}
```



```
java.lang.AssertionError: Members: [Frodo, Sam, Merry, Pippin, Gandalf, Aragorn, Legolas,  
Gimli] expected:<9> but was:<8>  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertEquals(9, members.size, "Members: $members")  
}
```

```
java.lang.AssertionError: Members: [Frodo, Sam, Merry, Pippin, Gandalf, Aragorn, Legolas,  
Gimli] expected:<9> but was:<8>  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertEquals(9, members.size, "Members: $members")  
}
```

✓ Complete failure message

```
java.lang.AssertionError: Members: [Frodo, Sam, Merry, Pippin, Gandalf, Aragorn, Legolas,  
Gimli] expected:<9> but was:<8>  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertEquals(9, members.size, "Members: $members")  
}
```

✓ Complete failure message

✗ Forget to add message

```
java.lang.AssertionError: Members: [Frodo, Sam, Merry, Pippin, Gandalf, Aragorn, Legolas,  
Gimli] expected:<9> but was:<8>  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertEquals(9, members.size, "Members: $members")  
}
```

✓ Complete failure message

✗ Forget to add message

✗ Maintenance burden

```
java.lang.AssertionError: Members: [Frodo, Sam, Merry, Pippin, Gandalf, Aragorn, Legolas,  
Gimli] expected:<9> but was:<8>  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertThat(members).hasSize(9)  
}
```

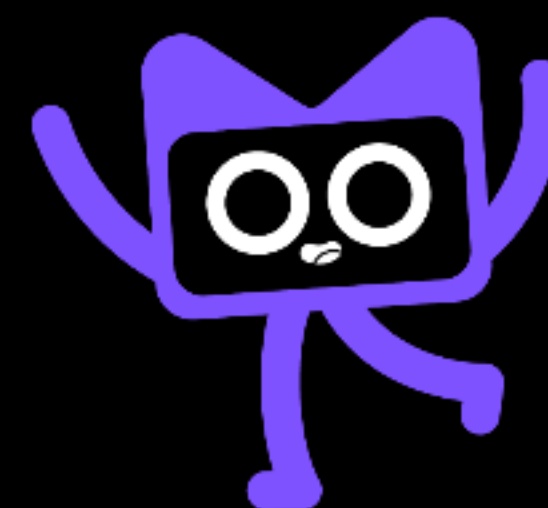


Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertThat(members).hasSize(9)  
}
```



```
org.opentest4j.AssertionFailedError: expected [size]:<[9]> but was:<[8]> ([Frodo, Sam, Merry,  
Pippin, Gandalf, Aragorn, Legolas, Gimli])  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertThat(members).hasSize(9)  
}
```

org.opentest4j.AssertionFailedError: expected [size]:<[9]> but was:<[8]> ([Frodo, Sam, Merry,
Pippin, Gandalf, Aragorn, Legolas, Gimli])
at [...]



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertThat(members).hasSize(9)  
}
```

✓ Complete failure message

```
org.opentest4j.AssertionFailedError: expected [size]:<[9]> but was:<[8]> ([Frodo, Sam, Merry,  
Pippin, Gandalf, Aragorn, Legolas, Gimli])  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertThat(members).hasSize(9)  
}
```

✓ Complete failure message

✗ Mental load for functions

```
org.opentest4j.AssertionFailedError: expected [size]:<[9]> but was:<[8]> ([Frodo, Sam, Merry,  
Pippin, Gandalf, Aragorn, Legolas, Gimli])  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertThat(members).hasSize(9)  
}
```

✓ Complete failure message

✗ Mental load for functions

✗ Assertions for custom types

```
org.opentest4j.AssertionFailedError: expected [size]:<[9]> but was:<[8]> ([Frodo, Sam, Merry,  
Pippin, Gandalf, Aragorn, Legolas, Gimli])  
    at [...]
```



Crafting an Assertion

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertThat(members).hasSize(9)  
}
```

✓ Complete failure message

✗ Mental load for functions

✗ Assertions for custom types

✗ Library choice fatigue

```
org.opentest4j.AssertionFailedError: expected [size]:<[9]> but was:<[8]> ([Frodo, Sam, Merry,  
Pippin, Gandalf, Aragorn, Legolas, Gimli])  
    at [...]
```



Crafting an Assertion



Why Power-Assert?



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assert(members.size == 9)  
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assert(members.size == 9)  
}
```

java.lang.AssertionError: Assertion failed



Why Power-Assert?

```
@Test fun `test`  
    val members :  
    assert(member  
}
```

```
java.lang.AssertionE
```

build.gradle.kts

```
plugins {  
    kotlin("jvm") version "2.0.0"  
}
```



Why Power-Assert?

```
@Test fun `test`  
    val members :  
    assert(member  
}
```

```
java.lang.AssertionE
```



 build.gradle.kts

```
plugins {  
    kotlin("jvm") version "2.0.0"  
    kotlin("plugin.power-assert") version "2.0.0"  
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assert(members.size == 9)  
}
```

java.lang.AssertionError: Assertion failed



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assert(members.size == 9)  
}
```

```
java.lang.AssertionError: Assertion failed  
assert(members.size == 9)  
|         |         |  
|         |         false  
|         8  
[Frodo, Sam, Merry, Pippin, Gandalf, Aragorn, Legolas, Gimli]
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assert(members.any { it.name == "Boromir" } &&  
        members.any { it.name == "Aragorn" } ||  
        members.any { it.name == "Elrond" })  
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assert(members.any { it.name = "Boromir" } &&  
           members.any { it.name = "Aragorn" } ||
```

```
java.lang.AssertionError: Assertion failed  
assert(members.any { it.name = "Boromir" } &&  
       |  
       |  
       | false  
[Frodo, Sam, Merry, Pippin, Gandalf, Aragorn, Legolas, Gimli]  
members.any { it.name = "Aragorn" } ||  
members.any { it.name = "Elrond" })  
|  
|  
| false  
[Frodo, Sam, Merry, Pippin, Gandalf, Aragorn, Legolas, Gimli]  
at [...]
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    val aragorn = members.find { it.name == "Aragorn" }  
    assertTrue(aragorn != null)  
    assertTrue(aragorn.age < 60)  
}
```



Why Power-Assert?

```
@Test fun `test`  
    val members :  
    val aragorn :  
    assertTrue(a  
    assertTrue(a  
}
```



build.gradle.kts

```
plugins {  
    kotlin("jvm") version "2.0.0"  
    kotlin("plugin.power-assert") version "2.0.0"  
}
```



Why Power-Assert?

```
@Test fun `test`  
    val members :  
    val aragorn :  
    assertTrue(a  
    assertTrue(a  
}
```



```
plugins {  
    kotlin("jvm") version "2.0.0"  
    kotlin("plugin.power-assert") version "2.0.0"  
}  
  
@OptIn(ExperimentalKotlinGradlePluginApi::class)  
powerAssert {  
}
```



Why Power-Assert?

```
@Test fun `test` {  
    val members :  
    val aragorn :  
    assertTrue(a  
    assertTrue(a  
}
```



```
plugins {  
    kotlin("jvm") version "2.0.0"  
    kotlin("plugin.power-assert") version "2.0.0"  
}  
  
@OptIn(ExperimentalKotlinGradlePluginApi::class)  
powerAssert {  
    functions = listOf(  
        "kotlin.test.assertTrue",  
    )  
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    val aragorn = members.find { it.name == "Aragorn" }  
    assertTrue(aragorn != null)  
    assertTrue(aragorn.age < 60)  
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    val aragorn = members.find { it.name == "Aragorn" }  
    assertTrue(aragorn != null)
```

```
java.lang.AssertionError: Assertion failed  
assertTrue(aragorn.age < 60)  
    |           |      |  
    |           |      false  
    |           87  
    Aragorn  
at [...]
```



Why Power-Assert?

```
fun addToFellowship(character: Character) {  
    require(character.alive)  
    members.add(character)  
}
```



Why Power-Assert?

```
fun addToFollows(  
    require(chara  
    members.add(  
)
```



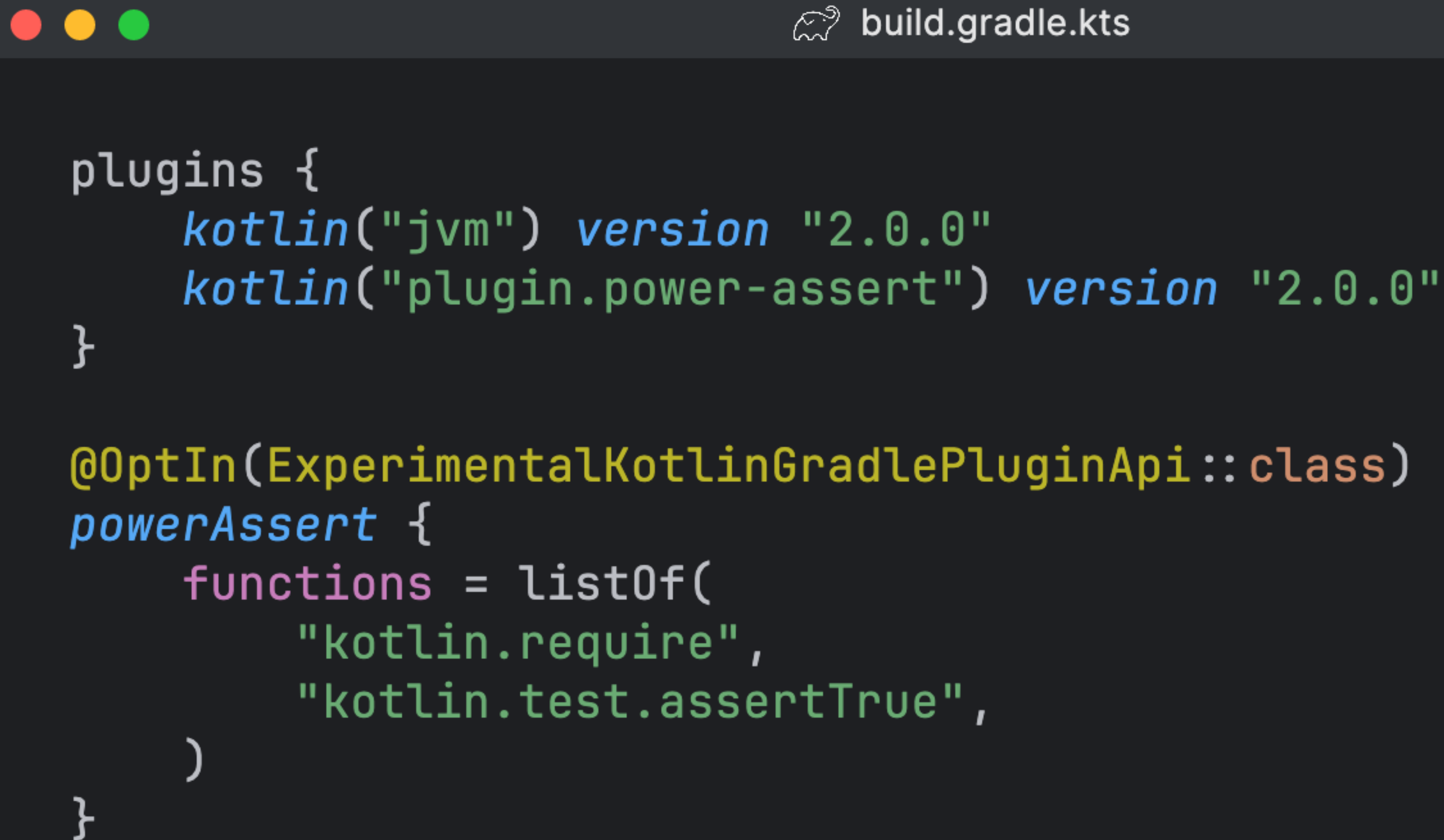
 build.gradle.kts

```
plugins {  
    kotlin("jvm") version "2.0.0"  
    kotlin("plugin.power-assert") version "2.0.0"  
}  
  
@OptIn(ExperimentalKotlinGradlePluginApi::class)  
powerAssert {  
    functions = listOf(  
        "kotlin.test.assertTrue",  
    )  
}
```



Why Power-Assert?

```
fun addToFollows(  
    require(chara  
    members.add(  
)
```



```
build.gradle.kts  
  
plugins {  
    kotlin("jvm") version "2.0.0"  
    kotlin("plugin.power-assert") version "2.0.0"  
}  
  
@OptIn(ExperimentalKotlinGradlePluginApi::class)  
powerAssert {  
    functions = listOf(  
        "kotlin.require",  
        "kotlin.test.assertTrue",  
    )  
}
```



Why Power-Assert?

```
fun addToFollows(  
    require(chara  
    members.add(  
)
```



 build.gradle.kts

```
plugins {  
    kotlin("jvm") version "2.0.0"  
    kotlin("plugin.power-assert") version "2.0.0"  
}  
  
@OptIn(ExperimentalKotlinGradlePluginApi::class)  
powerAssert {  
    functions = listOf(  
        "kotlin.require",  
        "kotlin.test.assertTrue",  
    )  
    includedSourceSets = listOf("main", "test")  
}
```



Why Power-Assert?

```
fun addToFellowship(character: Character) {  
    require(character.alive)  
    members.add(character)  
}
```



Why Power-Assert?

```
fun addToFellowship(character: Character) {  
    require(character.alive)  
    members.add(character)  
}
```



```
java.lang.IllegalArgumentException: Assertion failed  
require(character.alive)  
      |           |  
      |           false  
      |  
      Boromir  
at [...]
```



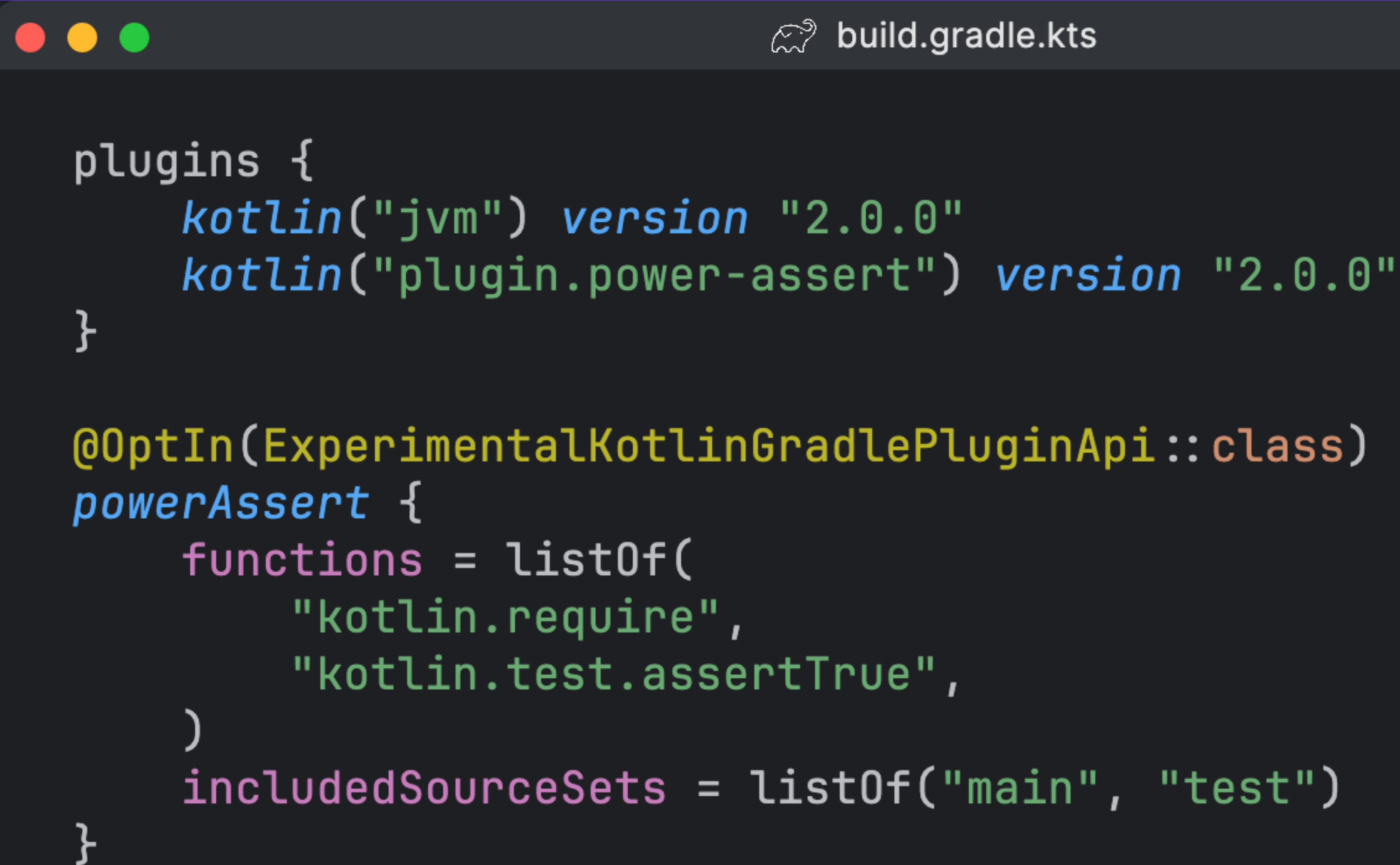
Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    val aragorn = members.find { it.name == "Aragorn" }  
    val boromir = members.find { it.name == "Boromir" }  
    assertEquals(aragorn?.race, boromir?.race)  
}
```



Why Power-Assert?

```
@Test fun `test` {  
    val members :  
    val aragorn :  
    val boromir :  
    assertEquals  
}
```



```
build.gradle.kts  
  
plugins {  
    kotlin("jvm") version "2.0.0"  
    kotlin("plugin.power-assert") version "2.0.0"  
}  
  
@OptIn(ExperimentalKotlinGradlePluginApi::class)  
powerAssert {  
    functions = listOf(  
        "kotlin.require",  
        "kotlin.test.assertTrue",  
    )  
    includedSourceSets = listOf("main", "test")  
}
```



Why Power-Assert?

```
@Test fun `test` {  
    val members :  
    val aragorn :  
    val boromir :  
    assertEquals  
}
```



```
plugins {  
    kotlin("jvm") version "2.0.0"  
    kotlin("plugin.power-assert") version "2.0.0"  
}  
  
@OptIn(ExperimentalKotlinGradlePluginApi::class)  
powerAssert {  
    functions = listOf(  
        "kotlin.require",  
        "kotlin.test.assertTrue",  
        "kotlin.test.assertEquals",  
    )  
    includedSourceSets = listOf("main", "test")  
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    val aragorn = members.find { it.name == "Aragorn" }  
    val boromir = members.find { it.name == "Boromir" }  
    assertEquals(aragorn?.race, boromir?.race)  
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    val aragorn = members.find { it.name == "Aragorn" }  
    val boromir = members.find { it.name == "Boromir" }
```

```
java.lang.AssertionError:  
assertEquals(aragorn?.race, boromir?.race)  
    |           |           |           |  
    |           |           | null  
    |           | null  
    | Dúndan  
    Aragorn  
expected:<Dúndan> but was:<null>  
    at [...]
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    val aragorn = members.find { it.name == "Aragorn" }  
    val boromir = members.find { it.name == "Boromir" }  
    assertEquals(aragorn?.race, boromir?.race)  
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    val aragorn = assertNotNull(members.find { it.name == "Aragorn" })  
    val boromir = assertNotNull(members.find { it.name == "Boromir" })  
    assertEquals(aragorn.race, boromir.race)  
}
```



Why Power-Assert?

```
@Test fun `test` {  
    val members :  
    val aragorn :  
    val boromir :  
    assertEquals  
}
```



```
plugins {  
    kotlin("jvm") version "2.0.0"  
    kotlin("plugin.power-assert") version "2.0.0"  
}  
  
@OptIn(ExperimentalKotlinGradlePluginApi::class)  
powerAssert {  
    functions = listOf(  
        "kotlin.require",  
        "kotlin.test.assertTrue",  
        "kotlin.test.assertEquals",  
    )  
    includedSourceSets = listOf("main", "test")  
}
```



Why Power-Assert?

```
@Test fun `test` {  
    val members :  
    val aragorn :  
    val boromir :  
    assertEquals  
}
```



```
plugins {  
    kotlin("jvm") version "2.0.0"  
    kotlin("plugin.power-assert") version "2.0.0"  
}  
  
@OptIn(ExperimentalKotlinGradlePluginApi::class)  
powerAssert {  
    functions = listOf(  
        "kotlin.require",  
        "kotlin.test.assertTrue",  
        "kotlin.test.assertEquals",  
        "kotlin.test.assertNotNull",  
    )  
    includedSourceSets = listOf("main", "test")  
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    val aragorn = assertNotNull(members.find { it.name == "Aragorn" })  
    val boromir = assertNotNull(members.find { it.name == "Boromir" })  
    assertEquals(aragorn.race, boromir.race)  
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    val aragorn = assertNotNull(members.find { it.name == "Aragorn" })  
    val boromir = assertNotNull(members.find { it.name == "Boromir" })  
}
```

```
java.lang.AssertionError:  
assertNotNull(members.find { it.name == "Boromir" })  
    |           |  
    |           null  
    [Frodo, Sam, Merry, Pippin, Gandalf, Aragorn, Legolas, Gimli]  
at [...]
```



Why Power-Assert?

```
package example
```

```
fun <R> assertSoftly(block: AssertScope.() → R): R
```

```
interface AssertScope {  
    fun assert(assertion: Boolean)  
}
```



Why Power-Assert?

```
package example
```

```
fun <R> assertSo
```

```
interface Assert  
    fun assert(a  
}
```

```
build.gradle.kts  
  
kotlin("jvm") version "2.0.0"  
kotlin("plugin.power-assert") version "2.0.0"  
}  
  
@OptIn(ExperimentalKotlinGradlePluginApi::class)  
powerAssert {  
    functions = listOf(  
        "kotlin.require",  
        "kotlin.test.assertTrue",  
        "kotlin.test.assertEquals",  
        "kotlin.test.assertNotNull",  
    )  
    includedSourceSets = listOf("main", "test")  
}
```



Why Power-Assert?

```
package example
```

```
fun <R> assertScope {
```

```
interface AssertScope {  
    fun assert(a: Boolean)  
}
```

```
build.gradle.kts  
  
kotlin("jvm") version "2.0.0"  
kotlin("plugin.power-assert") version "2.0.0"  
}  
  
@OptIn(ExperimentalKotlinGradlePluginApi::class)  
powerAssert {  
    functions = listOf(  
        "kotlin.require",  
        "kotlin.test.assertTrue",  
        "kotlin.test.assertEquals",  
        "kotlin.test.assertNotNull",  
        "example.AssertScope.assert",  
    )  
    includedSourceSets = listOf("main", "test")  
}
```



Why Power-Assert?

```
package example
```

```
fun <R> assertSoftly(block: AssertScope.() → R): R
```

```
interface AssertScope {  
    fun assert(assertion: Boolean)  
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertSoftly {  
        for (member in members) {  
            assert(member.age < 100)  
        }  
    }  
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertSoftly {  
        for (member in members) {
```

w: file:/// [...] Unable to find overload of function example.AssertScope.assert for power-assert transformation callable as:

- example.AssertScope.assert(String)
- example.AssertScope.assert(() → String)
- example.AssertScope.assert(kotlin.Boolean, String)
- example.AssertScope.assert(kotlin.Boolean, () → String)

Multiple failed assertions

java.lang.AssertionError: Multiple failed assertions

at [...]

Suppressed: java.lang.AssertionError

at [...]

Suppressed: java.lang.AssertionError

at [...]

Suppressed: java.lang.AssertionError



Why Power-Assert?

```
package example
```

```
fun <R> assertSoftly(block: AssertScope.() → R): R
```

```
interface AssertScope {  
    fun assert(assertion: Boolean)  
}
```



Why Power-Assert?

```
package example

fun <R> assertSoftly(block: AssertScope.() → R): R

interface AssertScope {
    fun assert(
        assertion: Boolean,
        message: (() → String)? = null,
    )
}
```



Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertSoftly {  
        for (member in members) {  
            assert(member.age < 100)  
        }  
    }  
}
```



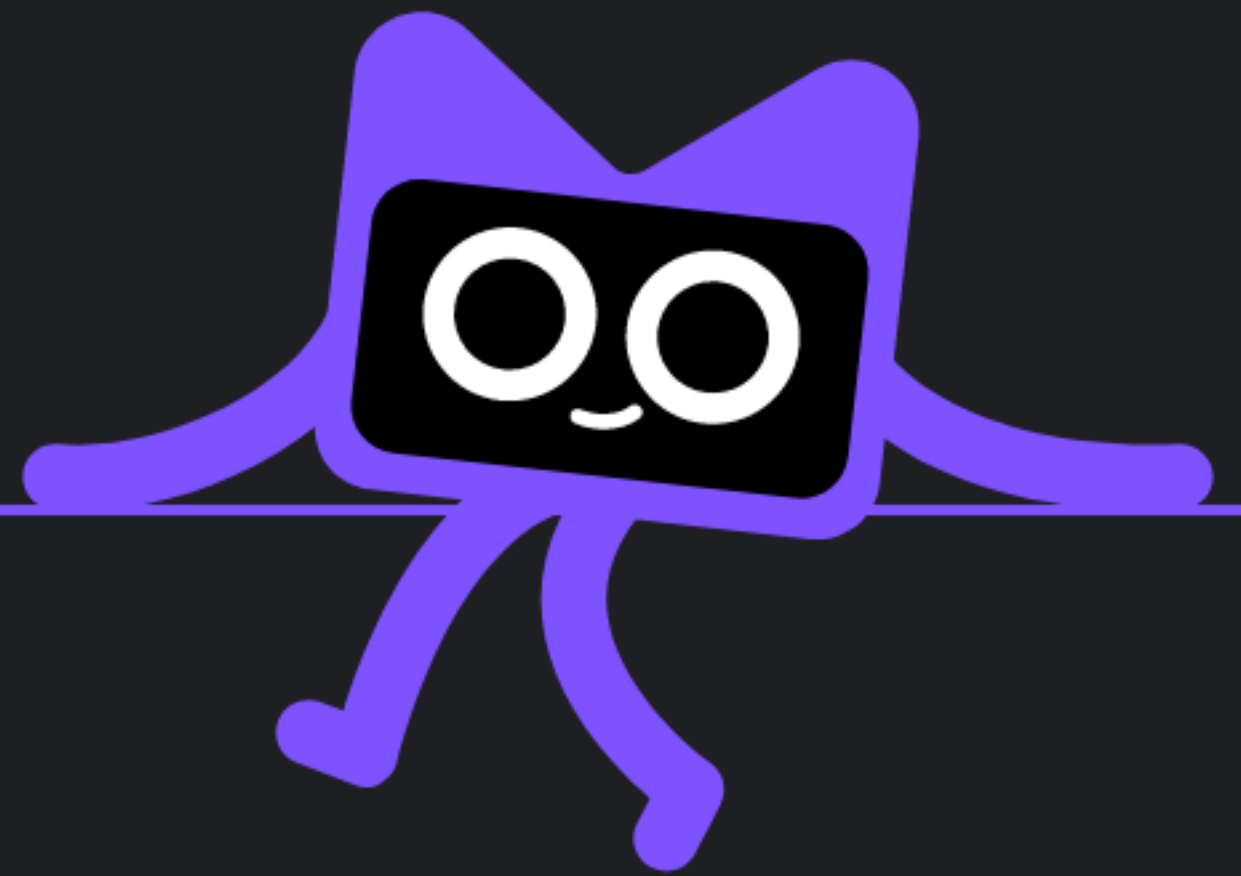
Why Power-Assert?

```
@Test fun `test members of the fellowship`() {  
    val members = fellowshipOfTheRing.getCurrentMembers()  
    assertSoftly {  
        for (member in members) {
```

```
Multiple failed assertions  
java.lang.AssertionError: Multiple failed assertions  
    at [...]  
    Suppressed: java.lang.AssertionError: Assertion failed  
assert(member.age < 100)  
    |         |         |  
    |         |         false  
    |         10000  
    Gandalf  
    at [...]  
    Suppressed: java.lang.AssertionError: Assertion failed  
assert(member.age < 100)  
    |         |         |  
    |         |         false  
    |         2931
```



Why Power-Assert?



A Look at the Future



A Look at the Future

- Improved diagrams



A Look at the Future

- Improved diagrams
 - Diagram formatting improvements ([KT-66807](#))



A Look at the Future

- Improved diagrams
 - Diagram formatting improvements ([KT-66807](#))
 - Diffs for strings and collections ([KT-66806](#))



A Look at the Future

- Improved diagrams
 - Diagram formatting improvements ([KT-66807](#))
 - Diffs for strings and collections ([KT-66806](#))
- Better integration



A Look at the Future

- Improved diagrams
 - Diagram formatting improvements ([KT-66807](#))
 - Diffs for strings and collections ([KT-66806](#))
- Better integration
 - Out-of-box support for kotlin.test ([KT-63622](#))



A Look at the Future

- Improved diagrams
 - Diagram formatting improvements ([KT-66807](#))
 - Diffs for strings and collections ([KT-66806](#))
- Better integration
 - Out-of-box support for kotlin.test ([KT-63622](#))
 - Support for other assertion libraries ([KT-66808](#))



A Look at the Future

- Improved diagrams
 - Diagram formatting improvements ([KT-66807](#))
 - Diffs for strings and collections ([KT-66806](#))
- Better integration
 - Out-of-box support for kotlin.test ([KT-63622](#))
 - Support for other assertion libraries ([KT-66808](#))
- Integration into the language



A Look at the Future

- We're looking for your feedback!



A Look at the Future

- We're looking for your feedback!
 - Try out Power-Assert! Docs: [kotl.in/power-assert](https://kotlinlang.org/docs/power-assert.html)



A Look at the Future

- We're looking for your feedback!
 - Try out Power-Assert! Docs: kotlin.in/power-assert
 - Report any compilation errors or strange diagrams



A Look at the Future

Docs: kotl.in/power-assert



A Look at the Future

Docs: kotl.in/power-assert

Slack: [#power-assert](#) (KotlinLang)



A Look at the Future

Docs: kotl.in/power-assert

Slack: [#power-assert](#) (KotlinLang)

Slides: deck.bnorm.dev/kotlinconf2024

(Powered by Compose Multiplatform!)



Docs: kotl.in/power-assert

Slack: [#power-assert](#) (KotlinLang)

Slides: deck.bnorm.dev/kotlinconf2024

 bnorm@kotlin.social

Thank you,
and don't
forget to vote!

