

Table of Contents

| | |
|------------------------------------|---|
| Heading 2 | 1 |
| Heading 3 | 1 |
| Heading 4 | 1 |
| The Dangers of Wolpertingers | 1 |
| Including code snippets | 2 |
| Include a csv file | 4 |
| References | 5 |

This is an example asciidoc file. It demonstrates basic features of the asciidoc format.

Some basic formatting

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Bold

Italics

Bold and italic

Monospace or code

Code block

```
println("Hello asciidoctor!")
```

From <https://docs.asciidoctor.org/asciidoctor/latest/get-started/>

The Dangers of Wolpertingers

Wolpertingers are [ravenous beasts](#).

From <https://docs.asciidoctor.org/asciidoc/latest/lists/unordered/>

Possible DefOps manual locations

- West wood maze
 - Maze heart
 - Reflection pool
 - Secret exit
- Untracked file in git repository

From <https://docs.asciidoctor.org/asciidoc/latest/blocks/sidebars/>

Sidebars are used to visually separate auxiliary bits of content that supplement the main text.

Optional Title

Sidebars are used to visually separate auxiliary bits of content that supplement the main text.

TIP They can contain any type of content.

Source code block in a sidebar

```
const { expect, expectCalledWith, heredoc } = require('../test/test-utils')
```

Including code snippets

This is awesome - we can insert snippets of code from files. There are several ways to do this, but here I'm showing how to do it by using tags in the source code.

You can tag the source code like this:

```
// tag::tag1[]  
awesome code goes here...  
// end::tag1[]
```

And then you can reference the tag in an include:

```
include:../../test/kotlin/org/example/LibraryTest.kt[tags=tag1]
```

And you'll get this:

```
@Test fun someLibraryMethodReturnsTrue() {  
    val classUnderTest = Library()  
    assertTrue(classUnderTest.someLibraryMethod(), "someLibraryMethod should return  
'true'")  
}
```

Or, using the source attribute (specified in gradle and in intellij plugin settings):

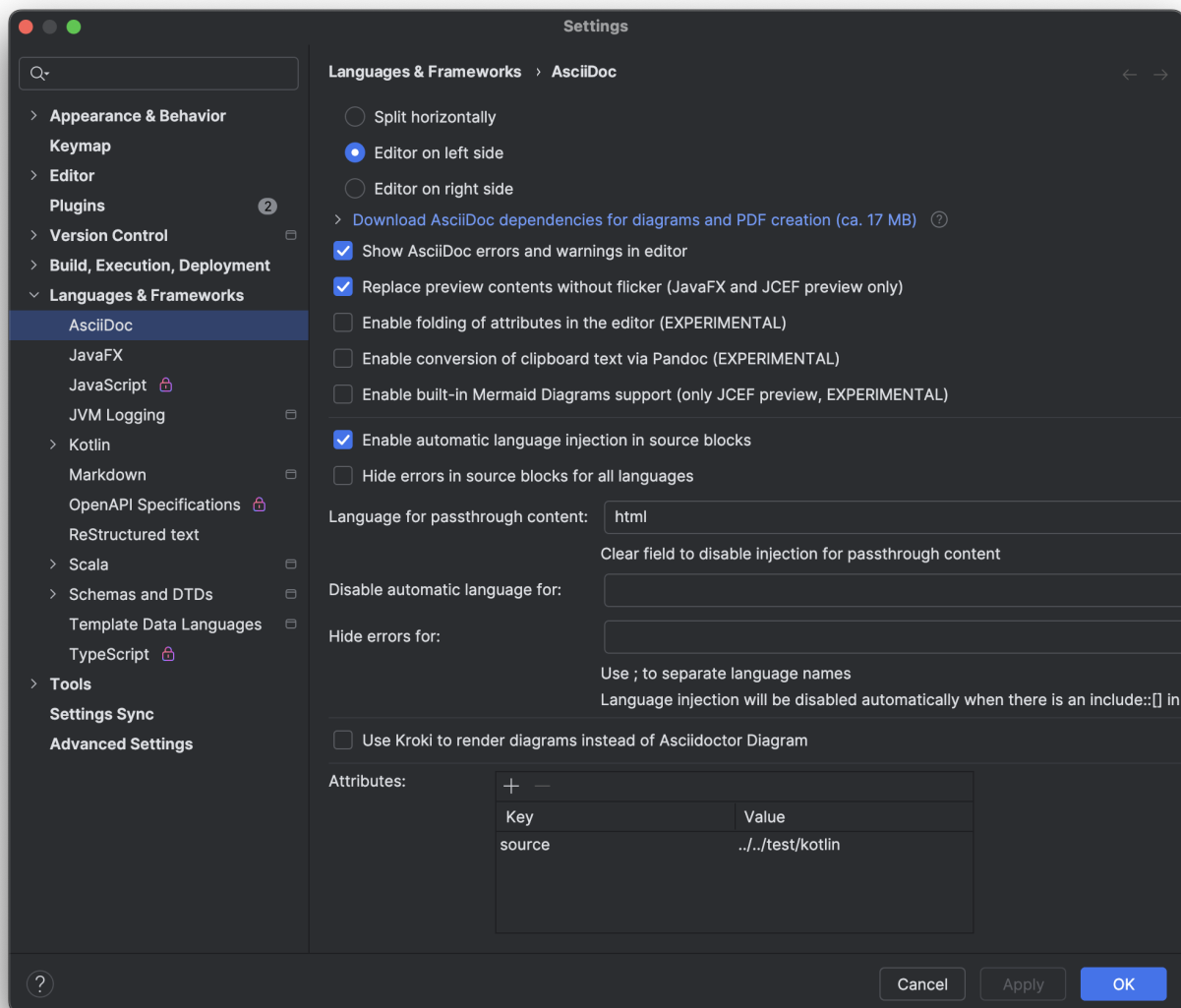
This

```
include:: {source}/org/example/LibraryTest.kt [tags=tag1]
```

Will produce:

```
@Test fun someLibraryMethodReturnsTrue() {  
    val classUnderTest = Library()  
    assertTrue(classUnderTest.someLibraryMethod(), "someLibraryMethod should return  
'true'")  
}
```

The intellij plugin will show you the result in a preview pane, and you can click on the tag to jump to it. If the link doesn't resolve, you'll see the error highlighted in the editor.



See:

- <https://docs.asciidoctor.org/asciidoc/latest/directives/include/>
- <https://docs.asciidoctor.org/asciidoc/latest/directives/include-tagged-regions/>
- <https://docs.asciidoctor.org/asciidoc/latest/directives/include-with-indent/>

Include a csv file

```
.Sample CSV
[source, csv]
,===
id,firstname,desc
1,Homer,Dad
2,Marge, Mother
,===
```

Output:

Table 1. Sample CSV

| id | firstname | desc |
|----|-----------|--------|
| 1 | Homer | Dad |
| 2 | Marge | Mother |

References

- AsciiDoctor
 - <https://asciidoctor.org>
 - <https://docs.asciidoctor.org>
- Gradle Support
 - <https://asciidoctor.github.io/asciidoctor-gradle-plugin/master/>
 - <https://github.com/asciidoctor/asciidoctor-gradle-examples/>
- IntelliJ Support
 - <https://matthewsetter.com/asciidoc-plugin-for-intellij-review/>
 - <https://github.com/asciidoctor/asciidoctor-intellij-plugin>
- More
 - Spring Rest Docs
 - <https://docs.spring.io/spring-restdocs/docs/current/reference/htmlsingle/#working-with-asciidoctor>
 - <https://intellij-asciidoc-plugin.ahus1.de/docs/users-guide/features/advanced/spring-rest-docs.html>