



Scala: Working with resources, folders and files

👤 Alex Fruzenshtein (<http://fruzenshtein.com/author/fruzenshtein/>) 🕒 03/02/2016 ❤️ Nope!

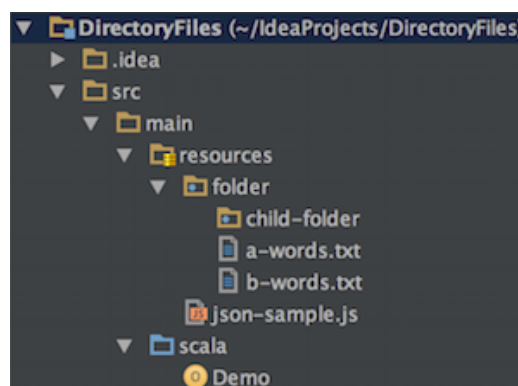
💬 7 Comments (http://fruzenshtein.com/scala-working-with-resources-folders-files/#disqus_thread)



Let's briefly discuss how to deal with a resources folder and directories in Scala project. It's pretty frequent case in a programming, when you need to interact with file system, Scala isn't an exception. So how it can be done on practice?

If you want to read about the most powerful way of reading & writing files in Scala (<http://198.211.104.161/scala-work-with-files-folders/>), please follow the link.

I'm going to demonstrate a short example on a real Scala project with a such structure:



As you see it has the **resources** folder with files and directories inside of it.

Access the resources folder

In order to get a path of files from the *resources* folder, I need to use following code:

```
1 |  
2 | object Demo {  
3 |  
4 |     def main(args: Array[String]): Unit = {  
5 |         val resourcesPath = getClass.getResource("/json-si  
6 |         println(resourcesPath.getPath)  
7 |     }  
8 | }
```

An output of the code below, is something like this:

```
1 |  
  | /Users/Alex/IdeaProjects/DirectoryFiles/out/production
```

Read file line by line

In some sense a pure path to a file is useless. Let's try to read the file from resources line by line. Scala can do it well:

```
1 |  
2 | import scala.io.Source  
3 |  
4 | object Demo {  
5 |  
6 |     def main(args: Array[String]): Unit = {  
7 |  
8 |         val fileStream = getClass.getResourceAsStream("/:  
9 |         val lines = Source.fromInputStream(fileStream).ge  
10 |        lines.foreach(line => println(line))  
11 |  
12 |     }  
13 | }
```

The output is

```
1 |  
2 | {  
3 |     "name": "Alex",  
4 |     "age": 26  
  | }
```

List files in directory

The final important and popular task is to list files from a directory.

```
1 |  
2 | import java.io.File  
3 |  
4 | object Demo {
```

Looks simple and efficient as well.

I hope this article was useful for you. Leave your comments and like us on facebook!

Let's learn Scala together!

Get a free copy of the "Scala OOP Basics"



It's a mini video course for Java developers. Within it you are able to become acquainted with basic concepts of Scala OOP and compare it to Java approach.

Write less, do more ;)

We promise to not use your email for spam!

📁 Categories: [Scala \(http://fruzenshtein.com/category/scala/\)](http://fruzenshtein.com/category/scala/)

🏷 Tags: [files \(http://fruzenshtein.com/tag/files/\)](http://fruzenshtein.com/tag/files/), [I/O \(http://fruzenshtein.com/tag/io/\)](http://fruzenshtein.com/tag/io/), [resources \(http://fruzenshtein.com/tag/resources/\)](http://fruzenshtein.com/tag/resources/)

(http://www.facebook.com/f	(https://twitter.com/t	(https://plus.google.com/p/g	(http://www.linkedin.com/in
---	---	---	---

ABOUT THE AUTHOR



(<http://fruzenshtein.com/author/fruzenshtein/>)

Alex Fruzenshtein (<http://fruzenshtein.com/author/fruzenshtein/>)

Mathematician, programmer, wrestler, last action hero... Java / Scala architect,
trainer, entrepreneur, author of this blog

(f) (t) (p) (l)

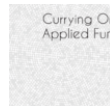
([h](http://)):

RELATED POSTS



Scala in Java Maven project
(<http://fruzenshtein.com/scala-in-java-maven-project/>)

(<http://fruzenshtein.com/scala-in-java-maven-project/>)
03/11/2016



Feel The Difference: Currying Or Partially Applied Function In Scala
(<http://fruzenshtein.com/currying-partially-applied-function-scala/>)

(<http://fruzenshtein.com/currying-partially-applied-function-scala/>)
11/03/2017



Scala: recursive functions
(<http://fruzenshtein.com/scala-recursive-function/>)

(<http://fruzenshtein.com/scala-recursive-function/>)
04/06/2016



How I would like to learn Scala?
(<http://fruzenshtein.com/how-i-would-like-to-learn-scala/>)

(<http://fruzenshtein.com/how-i-would-like-to-learn-scala/>)
03/12/2016

7 Comments Fruzenshtein's notes

 Login ▾ Recommend  Share

Sort by Best ▾



Join the discussion...

LOG IN WITH



OR SIGN UP WITH DISQUS (?)

Name **Alexander Glukhovtsev** • 2 years ago

Is it preferred way ?

And what about [scala.io.Source](#)?

^ | ▾ • Reply • Share >

**Alex Zvolinskiy** Mod → Alexander Glukhovtsev • 2 years ago

I described here java-style

Of course you can use much concise way such as:

```
Source.fromFile("file.txt")
```

^ | ▾ • Reply • Share >

**Alexander Glukhovtsev** → Alex Zvolinskiy • 2 years agoI was reading tutorial about [Swing](#)and the interesting part for me was about [ImageIO](#)

In particular,

how I can access the same properties as with

```
ImageReaders
```

in Scala way.

^ | ▾ • Reply • Share >

**Alex Zvolinskiy** Mod → Alexander Glukhovtsev • 2 years ago

Actually if you have a java code sample of something, you can repeat it with Scala syntax, because Scala supports Java libs and as you may guess a lib's API remains the same :)

^ | ▾ • Reply • Share >

**Alexander Glukhovtsev** → Alex Zvolinskiy • 2 years ago

All joking aside,

once upon a time I did the opposite

translating the Scala snippets into Java

(after that I do not like Java anymore).

1 ^ | ▾ • Reply • Share >

**Alex Zvolinskiy** Mod → Alexander Glukhovtsev • 2 years ago

Well agree :)

I prefer Scala over Java too, but sometimes still need to use old J :D

1 ^ | ▾ • Reply • Share >

MENU

› [Become an author \(http://fruzenshtein.com/become-an-author/\)](http://fruzenshtein.com/become-an-author/)

› [Contacts \(http://fruzenshtein.com/contacts/\)](http://fruzenshtein.com/contacts/)

› [About me \(http://fruzenshtein.com/alex-zvolinskiy/\)](http://fruzenshtein.com/alex-zvolinskiy/)

› [Site map \(http://fruzenshtein.com/site-map/\)](http://fruzenshtein.com/site-map/)

› [FruzNotes Magazine \(http://fruzenshtein.com/fruznotes-online-magazine/\)](http://fruzenshtein.com/fruznotes-online-magazine/)

SUPPORT THE BLOG



Donate Bitcoin Via QR-code

MAGAZINE



(/fruznotes-online-magazine/)

CATEGORIES

- › Akka (<http://fruzenshtein.com/category/akka/>)
- › Hibernate (<http://fruzenshtein.com/category/hibernate/>)
- › Java (<http://fruzenshtein.com/category/java/>)
- › MySQL (<http://fruzenshtein.com/category/mysql/>)
- › Other (<http://fruzenshtein.com/category/other-technologies/>)
- › Play Framework (<http://fruzenshtein.com/category/play-framework/>)
- › QA Automation (<http://fruzenshtein.com/category/automation/>)
- › Quick Fix (<http://fruzenshtein.com/category/quick-fix/>)
- › Reviews (<http://fruzenshtein.com/category/reviews/>)
- › Scala (<http://fruzenshtein.com/category/scala/>)
- › Spring (<http://fruzenshtein.com/category/spring/>)

( [https](#)

( [https](#)

( [https](#)

( [https](#)

2018 Fuzenshtein's Notes | Alexey Zvolinskiy aka Alex Fuzenshtein