Java NIO Path (with Examples)

> https://howtodoinjava.com/java/nio/how-to-define-path-in-java-nio/

The Path class, introduced in the Java SE 7 release, is one of the primary entry points of the java.nio.file package. If our application uses Java New IO, we should learn more about the powerful features available in this class.

In this Java tutorial, we are learning 6 ways to create a Path.

Table of Contents

> Prerequisite: I am building path for a file in location – "C:/Lokesh/Setup/workspace/NIOExamples/src/sample.txt ". I have created this file beforehand and will create Path to this file in all examples.
>
> 前提条件:我正在构建的文件路径的位置- " C:/Lokesh/Setup/workspace/NIOExamples/src/sample.txt "。我已经预先创建了这个文件，并将在所有示例中创建到这个文件的路径。

1. Building Absolute Path

An absolute path always contains the root element and the complete directory hierarchy required to locate the file. There is no more information required further to access the file or path.

To create an absolute path to a file, use getPath() method.

```
/**
* Converts a path string, or a sequence of strings that when joined
form a path string, to a Path. If more does not specify any elements
then the value of the first parameteris the path string to convert.
If more specifies one or more elements then each non-empty
* string, including first, is considered to be a sequence of
name elements and is joined to form a path string.
*/
public static Path get(String first, String... more);
```

Example 1: Create an absolute Path to a file in Java NIO

In all given examples, we are creating the absolute path for the same file, in different ways.

```
//Starts with file store root or drive
Path absolutePath1 = Paths.
get("C:/Lokesh/Setup/workspace/NIOExamples/src", "sample.txt");

Path absolutePath2 = Paths.
get("C:/Lokesh/Setup/workspace", "NIOExamples/src", "sample.txt");

Path absolutePath3 = Paths.
get("C:/Lokesh", "Setup/workspace", "NIOExamples/src", "sample.txt");
```

2. Building path relative to file store root

Path relative to file store root starts with a forward-slash (“/”) character.

Example 2: Create relative Path to a given file

```
//How to define path relative to file store root (in windows it is c:/)
Path relativePath1 = FileSystems
        .getDefault()
        .getPath("/Lokesh/Setup/workspace/NIOExamples/src", "sample.txt");

Path relativePath2 = FileSystems
        .getDefault()
        .getPath("/Lokesh", "Setup/workspace/NIOExamples/src",
 "sample.txt");
```

3. Building path relative to current working directory

To define the path relative to the current working directory, do not use either file system root (c:/ in windows) or slash (“/”).

Example 3: Create relative Path to current working directory

In given example, the current working directory is `NIOExamples`.

```
//How to define path relative to current working directory
Path relativePath1 = Paths.get("src", "sample.txt");
```

4. Building path from URI scheme

Not frequently, but at times we might face a situation where we would like to convert a file path in format “file:///src/someFile.txt” to NIO path.

Example 4: Get the absolute path of a file using file URI in Java NIO

```
//Writing c:/ is optional
//URI uri = URI.
create("file:///c:/Lokesh/Setup/workspace/NIOExamples/src/sample.txt");

URI uri = URI.
create("file:///Lokesh/Setup/workspace/NIOExamples/src/sample.txt");

String scheme =  uri.getScheme();
if (scheme == null)
    throw new IllegalArgumentException("Missing scheme");

//Check for default provider to avoid loading of installed providers
if (scheme.equalsIgnoreCase("file"))
{
    String absPath = FileSystems.getDefault()
                    .provider()
                    .getPath(uri)
                    .toAbsolutePath()
                    .toString();

    System.out.println(absPath);
}

//If you do not know scheme then use this code.
//This code check file scheme as well if available.
for (FileSystemProvider provider: FileSystemProvider.installedProviders())
{
    if (provider.getScheme().equalsIgnoreCase(scheme))
    {
        String absPath = provider.getPath(uri)
                    .toAbsolutePath()
                    .toString();

        System.out.println(absPath);
    }
}
```

5. Building path using file system default

This is another variation of above examples where instead of using `Paths.get()`, we can use `FileSystems.getDefault().getPath()` method.

The rules for absolute and relatives paths are the same as the above methods.

Example 5: Get absolute path of a file using system defaults

```
FileSystem fs = FileSystems.getDefault();

//relative path
Path path1 = fs.getPath("src/sample.txt");

//absolute path
Path path2 = fs.
getPath("C:/Lokesh/Setup/workspace/NIOExamples/src", "sample.txt");
```

6. Building path using System.getProperty()

Well, this is off the course, but good to know. We can use system-specific `System.getProperty()` also to build Path for specific files.

Example 6: Get path of a file in the system download folder

```
Path path1 = FileSystems.getDefault()
 .getPath(System.getProperty("user.home"), "downloads", "somefile.txt");
```