Оглавление

[ANT 1](#_Toc535930939)

[1. Строка: 1](#_Toc535930940)

[2. getClass().getResourse (“res/test.json”).toExternalForm 1](#_Toc535930941)

[3. ClassLoader 1](#_Toc535930942)

[GRADLE 1](#_Toc535930943)

[Структура 1](#_Toc535930944)

[Пример 1](#_Toc535930945)

# System.getProperty

## user.dir

## Папка проекта (напр., D:\GIT\Эксперименты\JacksonExamples)

## System.getProperty("user.dir")

## user.home

Папка пользователя в системе (С:\Users\user)

# ANT

## 1. Строка:

File **работает от корня**: File file2 = new File("./src/tests/test.json");

getClass().getResource("res").toExternalForm()

## 2. getClass().getResourse (“res/test.json”)

### URL

Path path = Paths.get(url.toURI());

### toExternalForm() -> String

Paths.get(string) – при получении файла выдаёт:

*java.nio.file.InvalidPathException: Illegal char <:> at index 4*

path = Paths.get(new URI(extFormString)); - **работает!!**

### Вывод:

***Поскольку в итоге нужно URI, то проще сразу использовать URL***

## 3. ClassLoader

**Тоже вариант!**(не проверял правда)

URI uri = ClassLoader.getSystemResource("tests/res/test.json").toURI();

String mainPath = Paths.get(uri).toString();

path = Paths.get(mainPath);

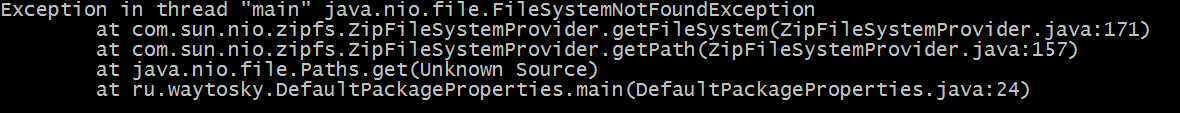
- Читает из файла в дефолтовом пакете

InputStream in = ru.wts.SomeClass.class.getClassLoader().getResourceAsStream("./default\_pkg.properties");

(в проекте канцелярии работало вот так: “default\_pkg.properties”!, может потому что вызов шел из статического метода)

## Запись в файл внутри проекта - Не получится!

**FileSystemNotFoundException!**



# GRADLE

|  |  |
| --- | --- |
| Структура | Пример |
| main  -> java  -> resources  --> subfolder  -----> file.xml | 0) Создаём файл в ресурсах(можно пустой)  1) uri = Object.class.getResource("/subfolder/file.xml")  (Object or AnyOtherClass – NO matter!)  uri будет на адрес **в билде!**:  **build/resources/main**/subfolder/file.xml |

# Failed to execute goal org.codehaus.mojo:exec-maven-plugin:1.2.1:exec (default-cli)

***Caused by: java.nio.file.FileSystemNotFoundException***

***at com.sun.nio.zipfs.ZipFileSystemProvider.getFileSystem(ZipFileSystemProvider.java:171)***

Объяснение: особенности исполнения файла в том, что за ресурсом он лезет в jar-архив

Решение: создание файловой системы для работы с архивом

<https://stackoverflow.com/questions/25032716/getting-filesystemnotfoundexception-from-zipfilesystemprovider-when-creating-a-p>