

Kotlin – Create File – Examples

Create File in Kotlin

Kotlin Create File – In Kotlin, new file could be created using `File.createNewFile()`, `File.writeText(text :String)`, `Files.writeBytes()` etc. There are many other ways to create a file in Kotlin. We shall look into the code implementation for some of them using example Kotlin programs.

Create File using `File.createNewFile()`

`File.createNewFile()` creates a new file if it does not exist already and returns Boolean value of **true**. If the file does exist already at the path provided, the method returns false. The file created is empty and has zero bytes written to it.

Using `File.createNewFile()` is the **best and safest procedure to create a new file**. Most of the other methods, we go through in this tutorial, would overwrite the file if it exists which may result in the loss of existing data which is not desired in some cases.

Example 1 – Kotlin Create File – `File.createNewFile()`

In the following example, we try to create a new file with name **data.txt**. For the first time, the file is created and true is returned. When we try to create the file for second time, as file data.txt is already created, we get false.

example.kt

```
import java.io.File

fun main(args: Array<String>) {

    val fileName = "data.txt"

    var file = File(fileName)

    // create a new file
    val isNewFileCreated :Boolean = file.createNewFile()

    if(isNewFileCreated){
        println("$fileName is created successfully.")
    } else{
        println("$fileName already exists.")
    }
}
```

```
// try creating a file that already exists
val isFileCreated :Boolean = file.createNewFile()

if(isFileCreated){
    println("$fileName is created successfully.")
} else{
    println("$fileName already exists.")
}

}
```

Output

```
data.txt is created successfully.
data.txt already exists.
```

Create File using File.writeText()

File.writeText() creates a new file if it does not exist already and writes the text (string argument) to the file. If an empty string is provided, the file is created and nothing is written to it. By default the file is encoded as UTF-8. Passing any other charset as second argument encodes the file accordingly.

Note : In case, the file already exists, it is overwritten and the existing data is lost.

Use this method if you are sure that the file does not exist already or overwriting the existing data does not affect your application.

Example 2 – File.writeText()

In this example, we will use File.writeText() to create a new File.

example.kt

```
import java.io.File

fun main(args: Array<String>) {

    val fileName = "data.txt"

    var file = File(fileName)

    // create a new file
    file.writeText("")

}
```

We have given an empty string, as argument to writeText(), to write data to file. You may provide the string you would like to write into this file.

Create File using File.writeText()

File.writeBytes() creates a new file if it does not exist already and writes the raw bytes of the ByteArray provided to the file created. If an empty ByteArray is provided, the file is created and nothing is written to it.

Note : In case, the file already exists, it is overwritten and the existing data is lost.

Use this method if you are sure that the file does not exist already or overwriting the existing data does not affect your application.

Example 3 – File.writeBytes()

In this example, we will use File.writeBytes() to create a new File.

example.kt

```
import java.io.File

fun main(args: Array<String>) {

    val fileName = "data.txt"

    var file = File(fileName)

    // create a new file
    file.writeBytes(ByteArray(0))
}
```

We have given an empty byte array, as argument to writeBytes(), to write data to file. You may provide the ByteArray you would like to write into this file.

Conclusion

In this [Kotlin Tutorial](#) – Kotlin Create File, we have learnt how to create a new file using File.createNewFile(). Also we have seen how File.writeText() and File.writeBytes() could be used to create a file and the side effects of overwriting the existing file if it already exists, with ample Examples.

Kotlin Java

◆ [Kotlin Tutorial](#)

Getting Started

- ◆ Setup Kotlin(Java) Project
- ◆ Kotlin Example Program
- ◆ Convert Java to Kotlin
- ◆ Kotlin Main Function
- ◆ Kotlin Loops
- ◆ Kotlin For Loop
- ◆ Kotlin While, Do While Loops
- ◆ Kotlin Repeat
- ◆ Kotlin Ranges
- ◆ Kotlin When

Object Oriented Concepts

Classes

- ◆ Kotlin - Class, Primary and Secondary Constructors
- ◆ Kotlin Sealed Class
- ◆ Kotlin Data Class
- ◆ Kotlin Enum
- ◆ Kotlin - Extension Functions

Inheritance

- ◆ Kotlin Inheritance
- ◆ Kotlin Override Method of Super Class

Abstraction

- ◆ Kotlin Abstraction
- ◆ Kotlin Abstract Class
- ◆ Kotlin - Interfaces
- ◆ Kotlin Null Safety

Exception Handling

- ◆ Kotlin Try Catch
- ◆ Kotlin Throw Exception
- ◆ Kotlin Custom Exception

Fix Compilation Errors

- ◆ Kotlin - Variable must be initialized
- ◆ Kotlin - Primary Constructor call expected
- ◆ Kotlin - Null can not be a value of a non-null type String
- ◆ Kotlin - Cannot create an instance of an abstract class

Kotlin - String Operations

- ◆ Kotlin - Compare Strings
- ◆ Kotlin - Replace String
- ◆ Kotlin - Split String
- ◆ Kotlin - Split String to Lines
- ◆ Kotlin - String Capitalize

Kotlin - Functions

- ◆ Kotlin Function - Default Arguments
- ◆ Kotlin - Use Function

Kotlin Collections

Kotlin List

- ◆ Kotlin List
- ◆ Kotlin List forEach

Kotlin File Operations

⇒ Kotlin - Create File

- ◆ Kotlin - Read File
- ◆ Kotlin - Read File as List of Lines
- ◆ Kotlin - Write to File
- ◆ Kotlin - Append Text to File
- ◆ Kotlin - Check if File Exists
- ◆ Kotlin - Copy a File to Other
- ◆ Kotlin - Iterate through all files in a directory
- ◆ Kotlin - Delete Recursively
- ◆ Kotlin - Get File Extension

▼ Kotlin - Get the Extension

Kotlin Interview Q/A

◆ [Kotlin Interview Questions](#)

Kotlin Android

◆ [Kotlin Android Tutorial](#)

Useful Resources

◆ [How to Learn Programming](#)