Foundation / FileManager / url(for:in:appropriateFor:create:)

Instance Method

url(for:in:appropriateFor:create:)

Locates and optionally creates the specified common directory in a domain.

iOS 4.0+ | iPadOS 4.0+ | Mac Catalyst 13.1+ | macOS 10.6+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

```
func url(
    for directory: FileManager.SearchPathDirectory,
    in domain: FileManager.SearchPathDomainMask,
    appropriateFor url: URL?,
    create shouldCreate: Bool
) throws -> URL
```

Parameters

`directory`

The search path directory. The supported values are described in `FileManager.SearchPathDirectory`.

`domain`

The file system domain to search. The value for this parameter is one of the constants described in `FileManager.SearchPathDomainMask`. You should specify only one domain for your search and you may not specify the `allDomainsMask` constant for this parameter.

`url`

The file URL used to determine the location of the returned URL. Only the volume of this parameter is used.

This parameter is ignored unless the `directory` parameter contains the value `FileManager.SearchPathDirectory.itemReplacementDirectory` and the `domain` parameter contains the value `userDomainMask`.

`shouldCreate`

Whether to create the directory if it does not already exist.

When creating a temporary directory, this parameter is ignored and the directory is always created.

Return Value

The NSURL for the requested directory. When using Objective-C, if an error occurs, this method returns `nil` and assigns an appropriate error object to the `error` parameter.

Discussion

You typically use this method to locate one of the standard system directories, such as the `Documents`, `Application Support` or `Caches` directories. After locating (or creating) the desired directory, this method returns the URL for that directory. If more than one appropriate directory exists in the specified domain, this method returns only the first one it finds.

> **Important**
>
> Passing a directory and domain pair that makes no sense (for example `FileManager.SearchPathDirectory.desktopDirectory` and `networkDomainMask`) raises an exception.

You can use this method to create a new temporary directory. To do so, specify `FileManager.SearchPathDirectory.itemReplacementDirectory` for the `directory` parameter, `userDomainMask` for the `domain` parameter, and a URL for the `url` parameter which determines the volume of the returned URL.

For example, the following code results in a new temporary directory with a path in the form of `/private/var/folders/d0/h37cw8ns3h1bfr_2gnwq2yyc0000gn/T/Temporary Items/Untitled/`:

Swift Objective-C

```
let desktop = URL(fileURLWithPath: "/Users/jappleseed/Desktop/")

do {
    let temporaryDirectory = try FileManager.default.url(
        for: .itemReplacementDirectory,
        in: .userDomainMask,
        appropriateFor: desktop,
        create: true
    )

    print(temporaryDirectory)
} catch {
    // Handle the error.
}
```

> **Important**
>
> If you use this method to create a temporary directory, you should not rely on the existence of that temporary directory after the app is exited. It is recommended that you remove any temporary directories that are created after they're no longer needed.

> **Handling Errors in Swift**
>
> In Swift, this method returns a nonoptional result and is marked with the `throws` keyword to indicate that it throws an error in cases of failure.
>
> You call this method in a `try` expression and handle any errors in the `catch` clauses of a do statement, as described in Error Handling in The Swift Programming Language and `About Imported Cocoa Error Parameters`.

See Also

Locating system directories

`func urls(for: FileManager.SearchPathDirectory, in: FileManager.SearchPathDomainMask) -> [URL]`

Returns an array of URLs for the specified common directory in the requested domains.

`func NSSearchPathForDirectoriesInDomains(FileManager.SearchPathDirectory, FileManager.SearchPathDomainMask, Bool) -> [String]`

Creates a list of directory search paths.

`func NSOpenStepRootDirectory() -> String`

Returns the root directory of the user’s system.