

FILESYSTEM.FILE

filesystem.file class provides methods that have general functions of a file.(create, delete of file directory and move, copy of files etc.)

Add the following line for filesystem.file class into a html file your own :

```
<script type="text/javascript" src="$MANAGER_WIDGET/Common/webapi/1.0/deviceapis.js"></script>
```

You can declare filesystem.file class like this :

```
ex) var filesystem = deviceapis.filesystem.file;
```

Contents

Methods

[createDirectory](#)

[deleteDirectory](#)

[copyTo](#)

[moveTo](#)

[createFile](#)

[deleteFile](#)

Methods

createDirectory

Description

Creates a new directory.

Parameters

■dirPath

- DOMString
- The relative directory path, it should only contain characters supported by the underlying filesystem.

Return

■File

- the file handler of the new directory

Emulator Support

Y

SDK Constraint

None

Example

```
var newDir = dir.createDirectory("newDir");  
var anotherNewDir = dir.createDirectory("newDir1/subNewDir1");
```

deleteDirectory

Description

Deletes a specified directory and directory tree if specified.	
Parameters	<ul style="list-style-type: none"> ■directory <ul style="list-style-type: none"> - DOMString - The full virtual path to the directory to be deleted (must be under the current one). ■recursive <ul style="list-style-type: none"> - Boolean - true : recursive deletion. delete all data in all subdirectories. Use with caution. ■onsuccess (Optional) <ul style="list-style-type: none"> - Function - Called if the directory is successfully deleted. ■onerror (Optional) <ul style="list-style-type: none"> - Function - Called if an error occurred.
Return	■Void
Emulator Support	Y
SDK Constraint	None
<p>Example</p> <pre> var documentsDir; function onsuccess(files) { for(var i = 0; i < files.length; i++) { if (files[i].isDirectory) documentsDir.deleteDirectory(files[i].fullPath, false, function(){ console.log("Directory Deleted"); }, function(e) { console.log("Error" + e.message); }); else documentsDir.deleteFile(files[i].fullPath, function(){ console.log("File Deleted"); }, function(e) { console.log("Error" + e.message); }); } } </pre>	
<h2>copyTo</h2>	
Description	
Copies (and overwrites if possible and specified) a file or a directory from a specified location to another specified location.	
Parameters	<ul style="list-style-type: none"> ■originFilePath <ul style="list-style-type: none"> - DOMString - The origin full virtual file or directory path, it must be under the current directory. ■destinationFilePath <ul style="list-style-type: none"> - DOMString - The new full virtual file path or directory path. ■overwrite <ul style="list-style-type: none"> - Boolean - true : enforces overwriting an existing file. ■onsuccess (Optional) <ul style="list-style-type: none"> - Function - Called when the file has been copied. ■onerror (Optional) <ul style="list-style-type: none"> - Function - Called if an error occurred.
Return	■Void
Emulator Support	Y

SDK Constraint	None
Example	
<pre> var documentsDir; function onSuccess(files) { for(var i = 0; i < files.length; i++) { if (files[i].isDirectory == false) documentsDir.copyTo(files[i].fullPath, "images/backup/"+files[i].name, false, function(){console.log("file copied");}); } } function onerror(error) { console.log("The error " + error.message + " occurred when listing the files in the selected folder"); </pre>	

moveTo

Description

Moves a file or a directory from a specified location to another.

Parameters	<ul style="list-style-type: none"> ■originFilePath <ul style="list-style-type: none"> - DOMString - The origin full virtual file or directory path, it must be under the current directory. ■destinationFilePath <ul style="list-style-type: none"> - DOMString - The new full virtual file path or directory path. ■overwrite <ul style="list-style-type: none"> - Boolean - true : enforces overwriting an existing file. ■onsuccess (Optional) <ul style="list-style-type: none"> - Function - Called when the file has been moved. ■onerror (Optional) <ul style="list-style-type: none"> - Function - Called if an error occurred.
Return	■Void
Emulator Support	Y
SDK Constraint	None

Example

```

var documentsDir;
function onSuccess(files) {
  for(var i = 0; i < files.length; i++) {
    if (files[i].isDirectory == false)
      documentsDir.moveTo(files[i].fullPath,
        "images/newFolder/"+files[i].name,
        false,
        function(){console.log("file moved");});
  }
}

```

createFile

Description

Creates a new empty file in a specified location.

Parameters	<ul style="list-style-type: none"> ■filePath <ul style="list-style-type: none"> - DOMString - The new file path, it should only contain characters supported by the underlying filesystem.
------------	--

Return	■File - the file handler of the new file
Emulator Support	Y
SDK Constraint	None
Example	
<pre>var newFile = dir.createFile("newFilePath");</pre>	

deleteFile

Description

Deletes a specified file.

Parameters	■file - DOMString - The full virtual path to the file to be deleted (must be under the current directory). ■onsuccess (Optional) - Function - Called if the file is successfully deleted. ■onerror (Optional) - Function - Called if an error occurred.
Return	■Void
Emulator Support	Y
SDK Constraint	None

Example

```
function onsuccess(files) {
  for(var i = 0; i < files.length; i++) {
    if (files[i].isDirectory)
      documentsDir.deleteDirectory(
        files[i].fullPath,
        false,
        function(){
          console.log("File Deleted");
        }, function(e) {
          console.log("Error" + e.message);
        });
    else
      documentsDir.deleteFile(
        files[i].fullPath,
        function(){
          console.log("Directory Deleted");
        }, function(e) {
          console.log("Error" + e.message);
        });
  }
}
```