

檔案操作實作(File)

□ Matis Hsiao

學習目標

□ 學習使用 Phonegap File 相關API

課程大綱

- File API功能及參數說明
- 程式碼解說
- 使用手機或平板實作

File API功能及參數說明

□ DirectoryEntry

- ❖ 建立目錄結構物件,供目錄處理使用
- ❖ 方法說明:
 - getMetadata:取得目錄修改日期
 - moveTo:移動目錄
 - copyTo:複製目錄
 - toURI:顯示該目錄的URI路徑
 - remove:移除該目錄(該目錄必須為空目錄)
 - getParent:取得上一層目錄
 - createReader:建立目錄讀取物件(DirectoryReader)
 - getDirectory:取得該目錄下的特定目錄或是建立目錄
 - getFile:取得該目錄下的特定檔案或是建立檔案
 - removeRecursively:移除該目錄包含其所擁有的檔案及目錄

□ DirectoryReader

- ❖ 讀取目錄供處理
 - readEntries:讀取該目錄下的所有檔案及目錄

□ File

- ❖ 檔案資源物件(取得檔案相關資訊)

File API功能及參數說明

□ FileEntry

- ❖ 建立檔案結構物件,供檔案處理使用
- ❖ 方法說明:
 - getMetadata:取得檔案的修改日期
 - moveTo:移動檔案
 - copyTo:複製檔案
 - toURI:顯示該目錄的URI路徑
 - remove:移除該檔案
 - getParent:取得上一層目錄
 - createWriter :建立檔案寫入器(FileWriter)
 - file :取得檔案的相關資訊

□ FileReader

- ❖ 讀取檔案內容

□ FileSystem

- ❖ 取得檔案系統資源

□ FileTransfer

- ❖ 檔案上傳及下載API

□ FileWriter

- ❖ 檔案寫入器

□ LocalFileSystem

- ❖ 本地端檔案系統資源

API範例HTML程式碼

```
<Button onClick="getDirEntry();">建立temp目錄</Button>
<Button onClick="showAllDirectory();">顯示目前目錄檔案清單</Button><br />
<Button onClick="openFile();">開啟檔案</Button>
<p>另存檔案名稱</p>
<input type="text" id="copyName" name="copyName" size="10">
<Button onClick="saveAsNewFile();">另存檔案</Button><br />
<p>重新命名名稱</p>
<input type="text" id="newName" name="newName" size="10">
<Button onClick="renameFile();">重新命名</Button><br />
<Button onClick="getFileData();">讀取檔案內容</Button>
<p>檔案寫入內容</p>
<input type="text" id="dataInput" name="dataInput" size="20"><br />
<Button onClick="writeFile();">寫入檔案</Button>
<Button onClick="writeFileEnd();">寫入到檔案結尾</Button>
<div id="log" width="<script>window.innerWidth</script>"></div>
```

API範例程式碼說明-1

```
/**宣告變數*/  
var fileRoot = null;//宣告檔案系統Root資源物件  
var opFileEntry = null;//宣告開啟檔案後要存放的檔案資源物件  
var writeMode = 0;//宣告寫入檔案的模式 0 = 直接覆蓋 1=檔案結尾  
var nowDir = null;//目前使用的資料夾目錄  
var nowFile = null;//目前使用的檔案  
var openFileName = "FileTest.txt";//檔案名稱  
//監聽PhoneGap裝置初始化完成事件  
document.addEventListener("deviceready",onDeviceReady,false);  
//裝置初始化完成  
function onDeviceReady() {  
    console.log(device.phonegap);  
    //取得本機檔案系統資源  
    window.requestFileSystem(LocalFileSystem.PERSISTENT, 0, gotFileSystemSuccess, fail);  
}  
//取得本機檔案系統資源成功  
function gotFileSystemSuccess(fileSystem) {  
    fileRoot = fileSystem.root;  
    showMessage("取得檔案系統資源:" + fileSystem.root.name);  
    var uri = fileSystem.root.toURI();  
    nowDir = fileRoot;  
    console.log("fileSystem.URI:" + uri);  
}
```

API範例程式碼說明-2

```
//顯示目前所在位置的所有目錄
function showAllDirectory(){
var directoryReader = fileRoot.createReader();
directoryReader.readEntries(success,fail);
}
//開啟檔案
function openFile(){
var flagOptions = {create: true, //如果找不到檔案或目錄則建立
exclusive: false //是否獨佔此資源
};
fileRoot.getFile(openFileName, flagOptions, gotFileEntry, fail);
}
//寫入檔案,並且取代之前的內容
function writeFile(){
    if(opFileEntry != null){
        //預設模式為取代內容
        writeMode = 0;
        opFileEntry.file(gotFile, fail);
        opFileEntry.createWriter(gotFileWriter, fail);
    }
}
```


API範例程式碼說明-3

```
function writeFileEnd(){//寫入資料到檔案
    if(opFileEntry != null){//寫入資料到目前的檔案結尾後
        writeMode = 1;
        opFileEntry.file(gotFile, fail);
        opFileEntry.createWriter(gotFileWriter, fail);
    }
}

function gotFileWriter(writer){//寫入資源取得成功開始寫檔
    if(writeMode == 1) //如果寫入模式為寫入檔案結尾則使用seek跳到檔案結尾在寫入檔案
    writer.seek(writer.length);
    writer.onwriteend = function(evt) { //當檔案寫入完成
    showMessage("檔案長度[1]:" + writer.length);

    /*//限制檔案長度為10,超過則捨棄
    writer.truncate(10);
    //需重新加入onwriteend事件讓其跳出寫檔結束的事件,否則會進入無窮迴圈
    writer.onwriteend = function(evt) {
        showMessage("splice size to 2");
    };
    showMessage("檔案長度[2]:" + writer.length);
    */
}
```

API範例程式碼說明-4

```
//寫入檔案
var dataInput = document.getElementById('dataInput');
    if(dataInput != null)
writer.write(dataInput.value);
}
//回傳FileEntry跟DirectoryEntry陣列
function success(entries) {
    var i;
    for (i=0; i < entries.length; i++) {
        if(entries[i].isFile == true)
            showMessage("File:" + entries[i].name);
        else if(entries[i].isDirectory == true)
            showMessage("Dir:" + entries[i].name);
    }
}
//取得FileEntry資源成功
function gotFileEntry(fileEntry) {
//設定現在開啟的FileEntry
opFileEntry = fileEntry;
//取得檔案資訊
fileEntry.getMetadata(getMetadataSuccess, fail);
}
```

API範例程式碼說明-5

```
function getMetadataSuccess(metadata){//取得檔案資訊
console.log("修改時間:" + metadata.modificationTime);
}
function getFileData(){//開啟檔案內容
if(opFileEntry != null)
opFileEntry.file(gotFile, fail);
}
function gotFile(file){//檔案內容取得成功
    nowFile = file; //設定目前檔案
    readDataUrl(file); //讀取檔案內容並使用base64編碼讀取
    readAsText(file); //讀取檔案內容並使用文字模式讀取
}
/**讀取檔案並轉換成為Base64編碼字串*/
function readDataUrl(file) {
    var reader = new FileReader();
    //開始讀取監聽事件
    reader.onload = function(evt){
console.log("Load file to data url:" + file.name + " file path:" + file.fullPath);
    }
}
```

API範例程式碼說明-6

```
//讀取完成事件
reader.onloadend = function(evt) {
  //showMessage("讀取檔案資源(DATA URL)");
  //showMessage(evt.target.result);
};
//讀取檔案資源為data url
reader.readAsDataURL(file);
}
/**讀取檔案並轉換成為文字,預設文字編碼為utf-8*/
function readAsText(file) {
  var reader = new FileReader();
  //開始讀取監聽事件
  reader.onload = function(evt){
    console.log("Load file to text:" + file.name + " file path:" + file.fullPath);
  }
  //讀取完成事件
  reader.onloadend = function(evt) {
    //showMessage("讀取檔案資源(文字模式)");
    showMessage("Text:" + evt.target.result);
  };
  //讀取資料為文字
  reader.readAsText(file);
}
```

API範例程式碼說明-7

```
function saveAsNewFile(){//另存新檔到目前temp目錄(使用copy複製)
    console.log("複製路徑:" + nowDir.fullPath);
    parentEntry = new DirectoryEntry(); //建立存檔目錄資源
    parentEntry.fullPath = nowDir.fullPath;
    var copyName = document.getElementById('copyName');
    console.log("parentEntry:" + parentEntry.fullPath + " newName:" + copyName.value);
    if(copyName != null && copyName.value != ""){//複製檔案
        opFileEntry.copyTo(parentEntry, copyName.value, saveAsNewFileSuccess, fail);
    }else
        showAlert("檔名為空值");}
function renameFile(){//重新命名檔案(使用移動)
showMessage("複製路徑:" + fileRoot.fullPath);
parentEntry = new DirectoryEntry(); //建立存檔目錄資源
    parentEntry.fullPath = fileRoot.fullPath;
    var newName = document.getElementById('newName');
    console.log("parentEntry:" + parentEntry.fullPath + " newName:" + newName.value);
    if(newName != null && newName.value != ""){
//複製檔案到新目錄
        opFileEntry.moveTo(parentEntry, newName.value , saveAsNewFileSuccess, fail);
    }else
        showAlert("檔名不能為空值");
}
```

API範例程式碼說明-8

```
function saveAsNewFileSuccess(entry){//存檔完成
showAlert("另存新檔完成:" + entry.fullPath);
}
function getDirEntry(){//取得特定目錄資源
    if(fileRoot != null){
        var flagOptions = {create: true,//如果找不到檔案或目錄則建立
            exclusive: false //是否獨佔此資源
        };
        fileRoot.getDirectory("temp", flagOptions , getDirEntrySuccess, fail);
    }
}
function getDirEntrySuccess(dirEntry) {//取得成功
    showMessage("目錄資源: " + dirEntry.name);
    showMessage("目錄路徑: " + dirEntry.fullPath);
    nowDir = dirEntry;
}
function fail(error) {//取得失敗
showMessage("Failed:" + showErrorMessage(error.code));
}
function showMessage(_msg){//顯示訊息
var log = document.getElementById('log');
    if(log != null)
        log.innerHTML += _msg + '<br />';
}
```

API範例程式碼說明-9

```
//顯示通知
function showAlert(msg) {
  navigator.notification.alert(
    msg, // 訊息
    alertDismissed,// 呼叫函式
    '訊息',// 標題
    '確定'// 按鈕名稱
  );
}
//使用者點擊關閉訊息
function alertDismissed() {
  console.log("alertDismissed");
}
//轉換錯誤代碼
function showErrorMessage(code){
  if(code == FileError.NOT_FOUND_ERR)
    return "FileError.NOT_FOUND_ERR";
  else if(code == FileError.SECURITY_ERR)
    return "FileError.SECURITY_ERR";
  else if(code == FileError.ABORT_ERR)
    return "FileError.ABORT_ERR";
  else if(code == FileError.NOT_READABLE_ERR)
    return "FileError.NOT_READABLE_ERR";
```

API範例程式碼說明-10

```
//接續上頁
else if(code == FileError.ENCODING_ERR)
return "FileError.ENCODING_ERR";
else if(code == FileError.NO_MODIFICATION_ALLOWED_ERR)
return "FileError.NO_MODIFICATION_ALLOWED_ERR";
else if(code == FileError.INVALID_STATE_ERR)
return "FileError.INVALID_STATE_ERR";
else if(code == FileError.SYNTAX_ERR)
return "FileError.SYNTAX_ERR";
else if(code == FileError.INVALID_MODIFICATION_ERR)
return "FileError.INVALID_MODIFICATION_ERR";
else if(code == FileError.QUOTA_EXCEEDED_ERR)
return "FileError.QUOTA_EXCEEDED_ERR";
else if(code == FileError.TYPE_MISMATCH_ERR)
return "FileError.TYPE_MISMATCH_ERR";
else if(code == FileError.PATH_EXISTS_ERR)
return "FileError.PATH_EXISTS_ERR";
else
return "UnkownError_Code:" + code;
}
```