

Desarrollo de aplicaciones móviles con PhoneGap



- Media
- Capture API

Media



The Media object provides the ability to record and play back audio files on a device.

Utilizando este API, una aplicación tiene la capacidad de reproducir y grabar ficheros de audio.

En cuanto a grabación de audio, esta API tiene capacidades limitadas si la comparamos con Capture API (que veremos a continuación)

Se recomienda no usarla para grabación de audio, únicamente para reproducción.

Todo el funcionamiento de esta API se base en el Media object.

Media Object

Antes de poder reproducir un fichero de audio, es necesario crear un objeto Media que apunte al fichero de audio:

```
theMedia = new Media(mediaFileURI, onSuccess, [onError], [onStatus])
```

El parámetro mediaFileURI apunta al fichero de audio al que se accederá. En el caso de que apunte a un fichero del sistema de archivos del dispositivo, hay que usar el File API.

Por otra parte, una aplicación también puede empaquetar los ficheros de audio en el propio directorio www de la aplicación que estamos desarrollando (y tendremos acceso directo)

```
theMedia = new Media("/thefile.mp3", onSuccess)
```

Parámetros adicionales

```
theMedia = new Media(mediaFileURI, onSuccess, [onError], [onStatus])
```

IMPORTANTE (esto es diferente a como trabajan el resto de APIs de PhoneGap):

La función `onSuccess` se ejecuta cada vez que el fichero de audio se reproduzca correctamente.

La función `onError` se ejecuta cada vez que que hay un error a la hora de reproducir el fichero de audio.

La función `onStatus` se ejecuta cada vez que hay un cambio de estado en la reproducción del fichero, por ejemplo cuando lo pausamos o terminamos la reproducción.

Reproducción de ficheros de audio

```
theMedia.play()
```

```
theMedia.stop()
```

```
theMedia.seekTo(3600);
```

Cuando una aplicación no necesita utilizar más un fichero de audio, debe liberar el objeto Media que ha creado:

```
theMedia.release()
```

Actualización de la UI mientras reproducimos audio

La aplicación necesita establecer un temporizador inmediatamente después de llamar al método play, en caso de que quiera realizar modificaciones a la interfaz mientras se está reproduciendo el audio.

```
var theTimer = window.setInterval(updateUI, 1000);
```

updateUI es una función que será ejecutada cada segundo. En esta función podemos es donde debemos actualizar la UI dependiendo de la posición de nuestro fichero de audio.

```
function updateUI() {  
    theMedia.getCurrentPosition(  
        function(position) {  
            // whatever you want  
        },  
        function(error) {  
            console.log(error);  
        }  
    )  
}
```

Grabación de ficheros de audio

Para la grabación de ficheros de audio, tras crear un objeto de tipo Media, se pueden ejecutar los siguientes métodos

```
theMedia.startRecord()
```

```
theMedia.stopRecord()
```

REPETIMOS

Para grabar ficheros de audio es mucho más recomendable usar el API Capture

Capture



Provides access to the audio, image, and video capture capabilities of the device.

El Capture API permite a una aplicación capturar audio, video e imágenes usando la aplicación nativa incorporada en el dispositivo.

La aplicación de cámara de fotos es la utilizada para tomar imágenes y videos, y la grabadora de voz será utilizada para capturar clips de audio.

Camera Vs Capture

Tenemos algo de solapamiento con las funcionalidades implementadas por Camera API y por Capture API, ya que las dos se pueden usar para tomar imágenes con la cámara de fotos.

Camera API fue implementada en primer lugar, y todavía se mantiene por retrocompatibilidad con las aplicaciones existentes.

Aunque con las dos APIs se pueden usar para tomar imágenes de la cámara de fotos, funcionan de diferente manera, pero permite especificar diversas fuentes a la hora de tomar imágenes: la cámara de fotos, la librería de fotografías o álbumes de fotos.

Capture API únicamente permite tomar imágenes de la cámara de fotos., pero nos permite también capturar audio y video/

Captura de audio:

```
navigator.device.capture.captureAudio(onCaptureSuccess,  
onCaptureError, captureOptions);
```

Captura de imágenes:

```
navigator.device.capture.captureImage(onCaptureSuccess,  
onCaptureError, captureOptions);
```

Captura de video:

```
navigator.device.capture.captureVideo(onCaptureSuccess,  
onCaptureError, captureOptions);
```

En las tres funciones anteriores, la función `onCaptureSuccess` es ejecutada cuando ha finalizado la correspondiente captura de video, de imagen o audio.

Un ejemplo de implementación sería el siguiente:

```
function OnCaptureSuccess(fileList) {  
    var len, i;  
    len = fileList.length;  
    if (len>0) {  
        for (i=0; i<len; i++) {  
            // do something with the file  
        }  
    }  
}
```

Cada elemento del array `fileList` soporta las siguiente propiedades: `name`, `fullPath`, `type`, `lastModifiedDate`, `size`.

Una aplicación puede usar estas propiedades para realizar operaciones con los ficheros devueltos al finalizar la captura.

Opciones de captura

Cada uno de estos métodos acepta un objeto `captureOptions` que controla ciertos aspectos de como la captura es efectuada. Todas las opciones no están soportadas por todos los tipos de captura.

	Duration	Limit	Mode
Audio	X	X	X
Imagen		X	X
Video	X	X	X

Un objeto `captureOption` se definiría de la siguiente manera:

```
var captureOptions = {duration: 5, limit: 3};
```

duration

Permite especificar el máximo número de segundos que se pueden tomar en la captura. En Android esta opción no está soportada.

limit

De acuerdo a la documentación, define el número máximo de capturas que se pueden realizar, permitiendo hacer menos que el máximo permitido.

mode

Cuando un dispositivo soporta múltiples formatos de fichero para un tipo particular de captura (por ejemplo, PNG o JPEG para imágenes), con esta propiedad podemos especificar que tipo es usado en la captura.