

AVPlayer

One of the primary uses for smartphones and tablets is to entertain the user by offering them access to quality content. Video is a key form of widely-used content, especially for tablet devices.

To achieve this on Apple devices, you can use an AVPlayer, which plays video files on iOS. Read on to learn more!

AVPlayer for video playback

The iOS software development kit (SDK) includes two classes that make video playback implementation on iOS devices extremely easy to implement when developing applications.

The AVPlayer is the most efficient way to display and play a video in an Apple app. Whenever you add this component to the layout of an activity, it provides a surface onto which a video may be played.

The following video formats are currently supported on iOS:

- H.263
- H.264 AVC
- MPEG-4 SP
- VP8

Aside from the surface to display a video, it has a wide range of methods that may be called in order to control the playback of video.

Methods for controlling video playback

Some of the more commonly used methods for controlling the playback of video include:

AVPlayer(url: URL(fileURLWithPath: path))

This specifies the path (as a URI) of the video media to be played. This can be either a remote video file URL or a local video file on the device.

play()

When this function is used, it plays the video.

currentTime()

When this function is used, it gets the current time of playback.

pause()

This function pauses video playback.

isMuted ()

This function returns a Boolean value that indicates whether a video is currently muted.

AVPlayerViewController

The AVPlayerViewController is the UI to control the video. Specifically, it is a view that contains media controls such as "Play/Pause", "Rewind", "Fast Forward" and a progress slider. It synchronizes the controls with the state of the MediaPlayer. By default, the AVPlayerViewController comes with a default set of controls. If you want to make changes to it, you must implement the AVPlayerViewController programmatically within your Swift code.