



POLITÉCNICA

ETSIT  
UPM

*dit*  
UPM

# Desarrollo de Apps para iOS Web Views

IWEB 2016-2017  
Santiago Pavón

ver: 2016.09.13

# UIWebView

- Es una UIView que implementa un navegador web.
- Puede mostrar distintos tipos de contenido:
  - la página web apuntada por un URL.
  - un string conteniendo HTML.
  - bytes + tipo MIME.
    - PDFs.
- Basado en WebKit
  - Proyecto Open Source
  - funcionalidad limitada
  - Soporta JavaScript (límite almacenamiento 10 MB).

# Mostrar Contenidos

- Petición asíncrona a un URL:

```
func loadRequest(_ request: URLRequest)
```

- Carga un string conteniendo HTML:

```
func loadHTMLString(_ string: String,  
                    baseURL: URL?)
```

- Carga bytes:

```
func loadData(_ data: Data,  
              mimeType mimeType: String,  
              textEncodingName: String,  
              baseURL: URL)
```

# Ejemplos

```
let webView = UIWebView()
```

```
if let url = URL(string: "http://www.upm.es") {  
    let req = URLRequest(url: url)  
    webView.loadRequest(req)  
}
```

```
webView.loadHTMLString("<a href='upm.es'>UPM</a>",  
                        baseURL: nil)
```

# Más Propiedades y Métodos

- Propiedades:

- `delegate (IBOutlet)`
- `request`
- `isLoading`
- `canGoBack`
- `canGoForward`
- `scalesPageToFit`
- `dataDetectorTypes`
- `scrollView`
- `. . .`

- Métodos:

- `stopLoading() (IBAction)`
- `reload() (IBAction)`
- `goBack() (IBAction)`
- `goForward() (IBAction)`
- `stringByEvaluatingJavaScript(from: String)`
- `. . .`

# UIWebViewDelegate

- Se llama si falla la carga del contenido.

```
func webViewDidFinishLoad(_ webView: UIWebView)
```

- Se llama antes de empezar a cargar contenido.

```
func webView(_ webView: UIWebView,  
             shouldStartLoadWith request: URLRequest,  
             navigationType: UIWebViewNavigationType) -> Bool
```

- Se llama tras terminar la carga del contenido.

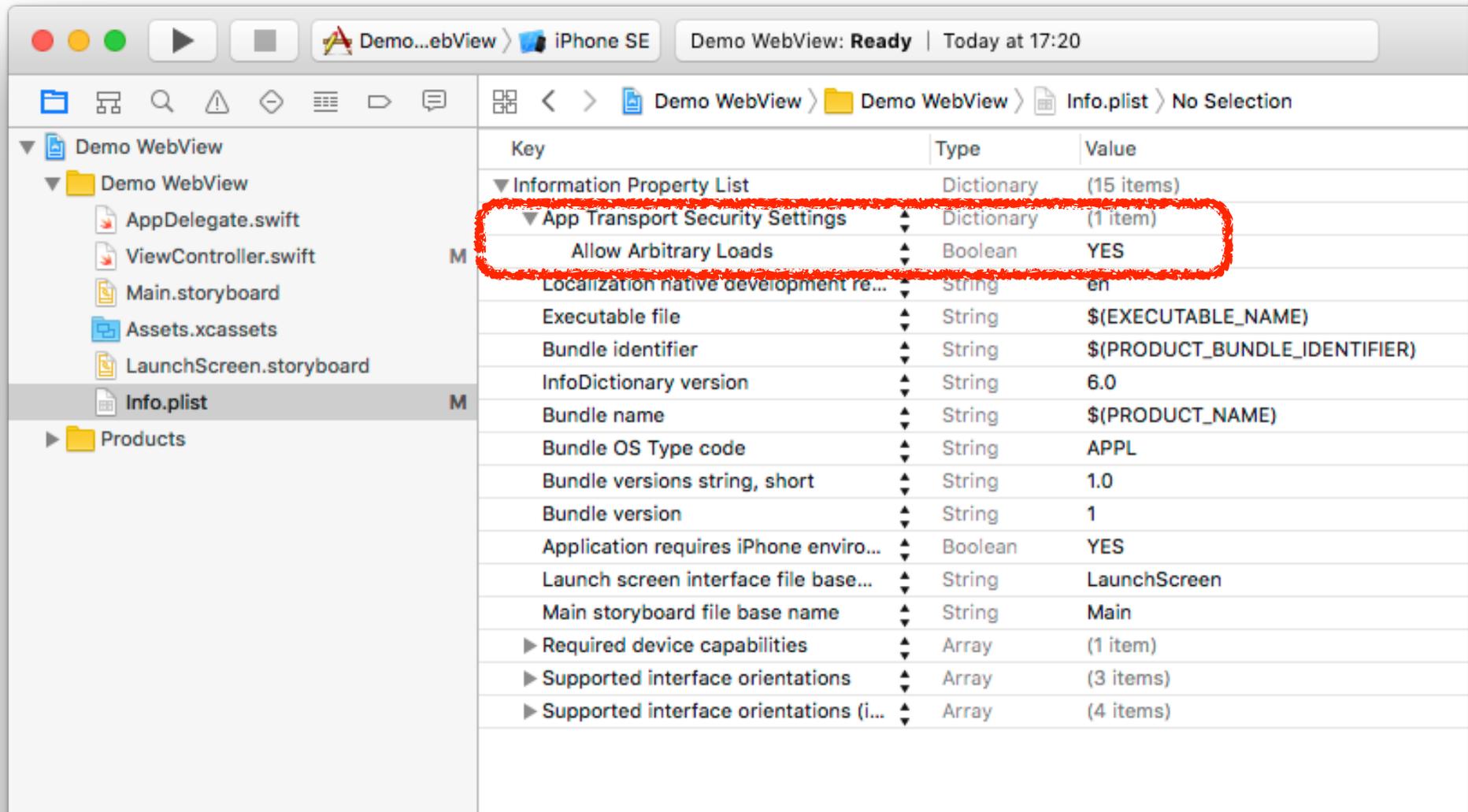
```
func webViewDidFinishLoad(_ webView: UIWebView)
```

- Se llama tras empezar la carga del contenido.

```
func webViewDidStartLoad(_ webView: UIWebView)
```

# Permitir Conexiones HTTP

- Añadir en **Info.plist**:



The screenshot shows the Xcode interface for a project named 'Demo WebView' on an iPhone SE simulator. The 'Info.plist' file is open, displaying a list of keys and values. The 'App Transport Security Settings' section is highlighted with a red box, indicating that 'Allow Arbitrary Loads' is set to 'YES'.

Key	Type	Value
Information Property List	Dictionary	(15 items)
App Transport Security Settings	Dictionary	(1 item)
Allow Arbitrary Loads	Boolean	YES
Localization native development re...	String	en
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle version	String	1
Application requires iPhone enviro...	Boolean	YES
Launch screen interface file base...	String	LaunchScreen
Main storyboard file base name	String	Main
Required device capabilities	Array	(1 item)
Supported interface orientations	Array	(3 items)
Supported interface orientations (i...	Array	(4 items)

- Añadir a **Info.plist**:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>NSAppTransportSecurity</key>
  <dict>
    <key>NSAllowsArbitraryLoads</key>
    <true/>
  </dict>
  <key>CFBundleDevelopmentRegion</key>
  <string>en</string>
  <key>CFBundleExecutable</key>
  <string>$(EXECUTABLE_NAME)</string>
  <key>CFBundleIdentifier</key>
  <string>$(PRODUCT_BUNDLE_IDENTIFIER)</string>
  <key>CFBundleInfoDictionaryVersion</key>
  . . .

```

# Demo: Browser

- Hacer un navegador Web.
- Incluir:
  - área para el contenido web
  - barra de direcciones
  - botones back, forward, stop, reload



# Framework : WebKit

# WebKit

- Framework más avanzado que proporciona la vista **WKWebView**.
  - Mejores prestaciones, mejor soporte de gestos, ...
  - Mayor control en el proceso de carga de páginas.
    - Varios tipos de delegados que pueden seguir el progreso de la carga.
  - Propiedades observables (KVO): título, progreso, URL.
  - User Scripts:
    - Podemos inyectar código javascript en las páginas visualizadas.
  - Script Messages:
    - El código nativo puede recibir notificaciones desde el javascript inyectado.
  - ...

