

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Control de autenticación y
 acceso (Spring Security)
 UDDI

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Web Services
 Rest Services
 Social SSO
 SSO (Cas)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



E-mail

Contraseña

Entrar

[Deseo registrarme](#)
[Olvíde mi contraseña](#)
» Estás en: [Inicio](#) [Tutoriales](#) [Intercomunicación de aplicaciones en IOS](#)**Rubén Aguilera Díaz-Heredero**

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática, especialidad en Ingeniería del Software

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)**Catálogo de servicios Autentia**

Fecha de publicación del tutorial: 2014-02-04

Tutorial visitado 3 veces [Descargar en PDF](#)

Intercomunicación de aplicaciones en IOS

0. Índice de contenidos.

- 1. Entorno
- 2. Introducción
- 3. Vamos al lío
- 4. Conclusiones

1. Entorno

Este tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Mac Book Pro 17" (2,6 Ghz Intel Core i7, 8 GB DDR3)
- Sistema Operativo: Mac OS X Snow Leopard 10.6.4
- Xcode 5.0.2
- InterAppCommunication 1.0

2. Introducción

Algo que cada vez resulta más común en el desarrollo con IOS es la necesidad de que unas aplicaciones interactuen con otras pasándose información. Esto en IOS se puede conseguir gracias a la especificación `x-callback-url`.Cada día más aplicaciones en el Apple Store publican sus `openUrls` para que nuestras aplicaciones puedan interactuar con ellas. Aquí tenéis un web donde podéis encontrar un listado actualizado con este tipo de aplicaciones: handleopenurl.com

Nosotros vamos a ver cómo se hace esta interacción con dos aplicaciones propias para que podáis ver todo el código necesario.

3. Vamos al lío

Vamos a crear dos aplicaciones sencillas basadas en el "HolaMundo" que vimos en este [tutorial](#). La primera de ellas invocará a la segunda pasándole un texto que será mostrado por pantalla.Para realizar la interacción nos vamos a basar en la librería `InterAppCommunication` que nos proporciona un API sencillo para trabajar con el estándar `x-callback-url`.La librería la podemos encontrar [aquí](#) y tenemos dos formas de incorporarla a nuestros proyectos: copiando directamente los fuentes o vía `CocoaPods`; por simplicidad en este tutorial vamos a copiar directamente los fuentes.

Ahora vamos a crear la aplicación que va a llamar a la otra pasándole el texto y que la vamos a conocer como "App1".

Creamos un nuevo proyecto llamado "App1" y creamos un `ViewController` llamado "App1ViewController" con un `xib` donde vamos a añadir un botón y un cuadro de texto donde poder introducir el texto que queremos transmitir a la otra aplicación. La pantalla quedaría así:**Síguenos a través de:****Últimas Noticias**

» IX Autentia Cycling Day (ACTUALIZADO)

» Buscamos quien nos ayude en Autentia con WordPress

» XXII Charla Autentia - PhoneGap/Cordova ¡qué bueno que viniste!

» Buscamos CM en @autentia (Enero 2014)

» La historia de la informática
[Histórico de noticias](#)**Últimos Tutoriales**

» App iOS para conectar con periférico bluetooth 4.0

» Crashlytics en IOS

» Construir un cliente REST con PowerBuilder .NET

» Acceso a la cámara con PhoneGap

» Mi primera vista en ZK como desarrollador JSF (II).

Últimos Tutoriales del Autor



- » Crashlytics en IOS
- » Acceso a la cámara con PhoneGap
- » Empezando con PhoneGap
- » Hello World en IOS sin StoryBoard
- » Ejecutando MyBatis contra distintas bases de datos

Últimas ofertas de empleo

- 2011-09-08
Comercial - Ventas - MADRID.
- 2011-09-03
Comercial - Ventas - VALENCIA.
- 2011-08-19
Comercial - Compras - ALICANTE.
- 2011-07-12
Otras Sin catalogar - MADRID.
- 2011-07-06
Otras Sin catalogar - LUGO.

Creamos un "Action" para cuando pulsamos en el botón llamado "CallApp2OnClick" con el siguiente código:

```
view plain print ?
01. - (IBAction)callApp2OnClick:(id)sender {
02.
03.     BOOL canOpenURL = [[UIApplication sharedApplication]
04.         canOpenURL:[NSURL URLWithString:
[NSString stringWithFormat:@"%s://Test", @"app2"]]];
05.
06.     if (canOpenURL){
07.         IACClient *client = [IACClient clientWithURLScheme:@"app2"];
08.         [client performAction:@"pasarTexto"
09.             parameters:@{@"texto": textoAPasar.text}
10.             onSuccess:^(NSDictionary *resultParams){
11.                 NSLog(@"La app respondio: %@", resultParams);
12.             }
13.             onFailure:^(NSError *error){
14.                 NSLog(@"ERROR: %@", [error localizedDescription]);

```

```

15.     }
16.     };
17.     NSLog(@"Llamada a app2 realizada");
18. }else{
19.     NSLog(@"App2 no está instalada en este dispositivo");
20. }
21. }
22. }

```

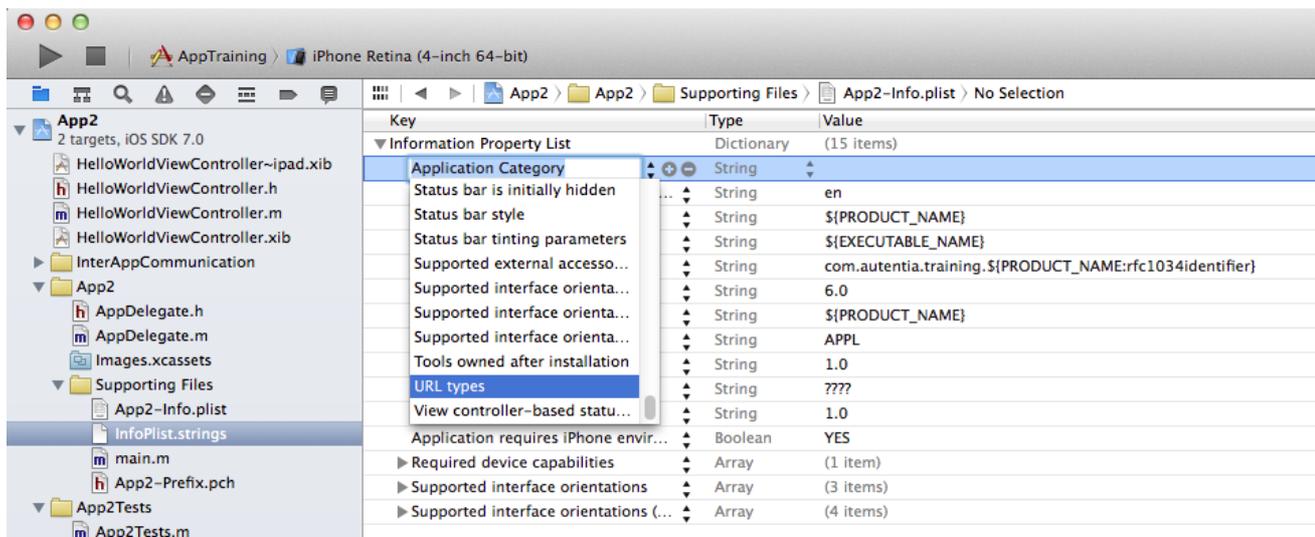
Este código lo que hace es comprobar si se puede realizar la llamada a la otra aplicación. En caso contrario significa que la otra aplicación no está presente en el dispositivo y por tanto mostramos un texto advirtiéndolo.

En caso de que si esté instalada la invocamos con la acción "pasarTexto" y el texto que escribimos que se mostrará por pantalla como veremos un poco más adelante.

En este punto tenemos que conocer la URLScheme de la aplicación a la que queremos llamar, en nuestro caso va a ser "app2".

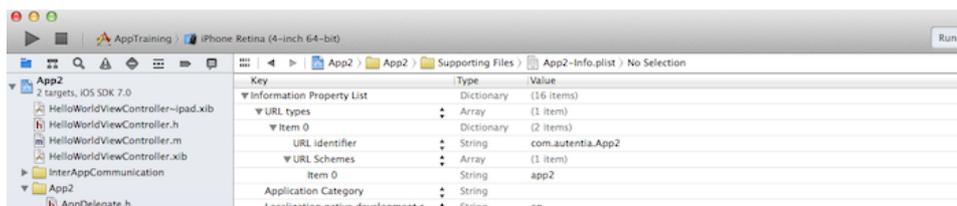
Ahora vamos a crear la "App2". Para ello creamos un nuevo proyecto llamado "App2". Una vez tenemos la estructura del proyecto tenemos que añadir la configuración necesaria para exponer nuestro servicio de "pasarTexto". No olvidar copiar los fuentes de la librería InterAppCommunication, como hicimos en la "App1".

Lo primero es definir el scheme de nuestra aplicación para que pueda ser localizada por la otra. Para ello abrimos el fichero nombreAplicacion-Info.plist, en este caso, App2-Info.plist. En este fichero pulsamos con el botón derecho y seleccionamos "Add Row" y "URL Types".



Rellenamos la información "Url Types" -> "Item 0" -> "URL identifier" con el texto que queremos aunque por convenio se suele poner el mismo identificador que la aplicación, en nuestro caso, com.autentia.App2

Ahora desde "Url identifier" pulsamos en "+" añadiendo la opción "URL Schemes" y dándole el valor "app2" al ítem0, como se muestra en la imagen:



Es muy importante que el nombre del ítem0 de "URL Schemes" coincida exactamente con el nombre que hemos establecido en el URLScheme de la app1.

Ahora vamos a implementar la interfaz con la que se va a conectar la app1. Editamos el fichero AppDelegate.m de la aplicación "App2" añadiendo el siguiente método:

```

view plain print ?
01. - (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
02.     sourceApplication:(NSString *)sourceApplication annotation:(id)annotation {
03.
04.     return [[IACManager sharedManager] handleOpenURL:url];
05. }

```

Dentro del método didFinishLaunchingWithOptions vamos a establecer el URLScheme y el handle de la operación "pasarTexto".

```

view plain print ?
01. - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions
02. {
03.     [[IACManager sharedManager].callbackURLScheme = @"app2";
04.
05.     [[IACManager sharedManager] handleAction:@"pasarTexto"
06.         withBlock:^(NSDictionary *inputParameters, IACSuccessBlock success, I
07.             App2ViewController *app2 = [[App2ViewController alloc] initWithNi
08.                 app2.texto = inputParameters[@"texto"];
09.                 self.window.rootViewController = app2;
10.
11.             }]];
12.
13.     self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen] bounds];

```

```
14.
15. // Override point for customization after application launch.
16. self.window.backgroundColor = [UIColor whiteColor];
17. [self.window makeKeyAndVisible];
18. return YES;
19. }
```

Llegados a este punto solo queda probar la interacción. Para ello tenemos que arrancar primero la app2 que mostrará inicialmente una pantalla en blanco, paramos el simulador y arrancamos la app1.

Que mostrará esta pantalla:



Si todo es correcto al pulsar el botón de la pantalla de la app1 veremos como se carga la app2 con la pantalla que muestra el texto que le hemos pasado.



4. Conclusiones

Cada vez más los desarrolladores en IOS adoptan x-callback-url para dar soporte a otras aplicaciones, con este tutorial ya sabéis como aprovechar la intercomunicación de aplicaciones en IOS.

Cualquier duda o sugerencia en la zona de comentarios.

Saludos.

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

[Share](#) |

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» [Regístrate](#) y accede a esta y otras ventajas «



Esta obra está licenciada bajo licencia [Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

PUSH THIS

Page Pushers

Community

Help?

no clicks

0 people brought clicks to this page

+ + + + + + + +

powered by [karmacacy](#)

Copyright 2003-2014 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

