

Documentació App2App

Pau Castells

4 de setembre de 2013

1 Introducció

La comunicació entre dues apps consta de dues parts. La primera és la que es porta a terme en l'aplicació que fa la crida per obrir l'altre. La segona es duu a terme en l'aplicació receptora. Aquesta segona part és la que pot ser una mica més complicada.

En aquest exemple veurem com en una app (emisor) fas un login mitjançant OAuth2 i accedeixes a la segona app (receptor) on consultes els tutors de l'usuari.

2 Emisor

L'emisor simplement haurà de tenir una funció que quan es cridi, mitjançant un botó, cella d'una taula o una altra acció, obri la segona aplicació. Aquesta funció basicament crida a la URL identificadora que se l'hi hagi donat a l'app receptora i se l'hi afegeixen els paràmetres que pugui necessitar. En aquest cas l'identificador del receptor és **receptor1** i necessitem l'**access_token**. Per això la URL que creem és **receptor1://tutors?access_token=valorAccessToken**. Si volguéssim afegir més paràmetres s'afegeixen separats per & després del primer ?.

```
- (IBAction)openReceptor1
{
    NSString *urlString = [NSString stringWithFormat:
        @"receptor1://tutors?access_token=%@", self.auth.accessToken];
    NSURL *myURL = [NSURL URLWithString:urlString];
    // Comprovem que es pugui obrir l'aplicacio que volem.
    if ([[UIApplication sharedApplication] canOpenURL:myURL] == YES){
        [[UIApplication sharedApplication] openURL:myURL];
    }
}
```

3 Receptor

El cas del receptor és bastant més complicat que el de l'emisor. Primer caldrà definir la URL que identifiqui l'app en el fitxer **Info.plist**.

Hem afegit el camp de URL Schemes i hi hem afegit l'Item 0 amb el valor receptor1.

▼ URL types	Array	(1 item)
▼ Item 0 (Editor)	Dictionary	(3 items)
Document Role	String	Editor
URL identifier	String	com.uoc.openApi
▼ URL Schemes	Array	(1 item)
Item 0	String	receptor1

En el fitxer **AppDelegate.m** hem tingut que afegir la funció que s'encarrega de rebre les crides del sistema quan es vol obrir aquesta app des de una altra. També hem afegit una funció per passar els paràmetres rebuts a un diccionari.

```
- (BOOL) application:(UIApplication *)application openURL:(NSURL *)url
    sourceApplication:(NSString *)sourceApplication
    annotation:(id)annotation
{
    //Obtenim els valors que hem rebut per la query
    NSDictionary *dict = [self parseQueryString:url query];

    //Creem un avis que ens permetra comunicar-nos amb l'activitat
    principal.
    [[NSNotificationCenter defaultCenter]
     postNotificationName:@"segueListener" object:nil userInfo:dict];
    return YES;
}

// Aquesta funcio serveix per guardar tots els valors que tens en la
// query de la URL rebuda en un diccionari. En aquest cas ens servira
// per guardar l'access_token
- (NSDictionary *)parseQueryString:(NSString *)query {
    NSMutableDictionary *dict = [[NSMutableDictionary alloc]
    initWithCapacity:6];
    NSArray *pairs = [query componentsSeparatedByString:@"&"];
    for (NSString *pair in pairs) {
        NSArray *elements = [pair componentsSeparatedByString:@"="];
        NSString *key = [[elements objectAtIndex:0]
        stringByReplacingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
        NSString *val = [[elements objectAtIndex:1]
        stringByReplacingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
        [dict setObject:val forKey:key];
    }
    return dict;
}
```

En la vista principal **ViewController.m** caldrà afegir una línia en la funció **viewDidLoad**.

```
- (void)viewDidLoad
{
    self.loadedAuth = 0;
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a
    nib.
    // Mirem si tenim alguna crida al centre de notifikacions.
    [[NSNotificationCenter defaultCenter] addObserver:self
        selector:@selector(segueFromBoards:) name:@"segueListener"
        object:nil];
}
```

I també ens caldrà implementar la funció **segueFromBoards**.

```
// Aquesta funcio gestiona la notifikacio rebuda del centre de
notifikacions
-(void) segueFromBoards:(NSNotification *)notification
{
    NSDictionary *file = [notification userInfo];

    TutorsViewController *tutorsViewController = [self.storyboard
        instantiateViewControllerWithIdentifier:@"tutorsView"];
    // inicialitzem l'autentificacio.
    [self myCustomAuth];
    // En aquest cas ens interessa obtenir el access_token per a no
    tenir-nos de autenticar un altre cop.
    self.auth.accessToken = [file objectForKey:@"access_token"];
    tutorsViewController.auth = self.auth;
    self.loadedAuth = 1;
    [self.navigationController pushViewController:tutorsViewController
        animated:YES];
}
```

4 Altres

Com ja hem vist podem obrir l'aplicació receptora des de una altra app, però també podem iniciar aquesta de manera normal. Per a que l'app sàpiga com s'ha iniciat es pot passar una valor booleà. Posteriorment podem utilitzar aquest valor booleà per saber en quin punt de l'app hem de sortir o anar a la vista anterior. En l'exemple el valor s'identifica per **isLoad**. Aquí hi ha un exemple del codi afegit a la vista dels tutors **TutorsViewController.m**.

```
-(void) viewWillAppear:(BOOL)animated {
    if ([self.navigationController.viewControllers
        indexOfObject:self]==NSNotFound) {
```

```

        // back button was pressed. We know this is true because self is
        // no longer
        // in the navigation stack.
        if(self.isLoad == NO) exit(0);
    }
    [super viewWillDisappear:animated];
}

```

5 Recursos

Es poden trobar els fitxers utilitzats en l'**Exemple App2App**. Els fitxers de l'aplicació emisora es poden trobar en l'**Exemple Call** i els de l'aplicació receptora en l'**Exemple Receptor**.