

Launching an Application

Fabrice.Kordon@lip6.fr



As an introduction...

Something useful

- Allows to launch an App from another one
- Strong constraints (security concerns)

Mainly possible for designated Apps

- URL (Safari)
- Calls/Facetime
- Messages (SMS)
- Mail
- Map

Example provided on these ones

- iTunes
- Youtube (in Safari?)
- Etc.

Principles

In UIApplication

- Launch (via an URL)
- Check for availability

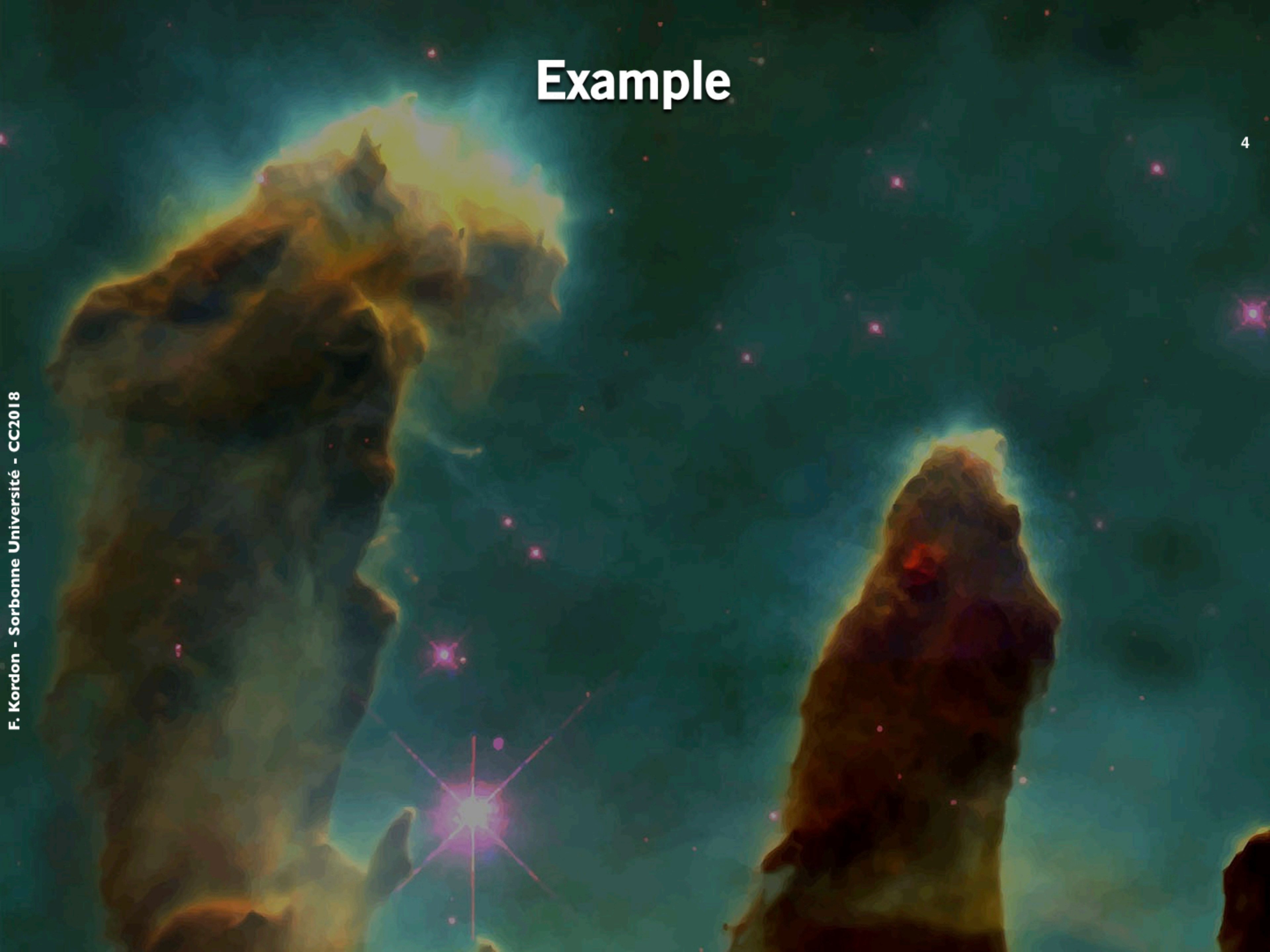
```
- (void)openURL:(NSURL *)url  
    options:(NSDictionary<NSString *,id> *)options  
    completionHandler:(void (^)(BOOL success))completion;
```

```
- (BOOL)canOpenURL:(NSURL *)url;
```

```
func open(_ url: URL,  
         options: [String : Any] = [:],  
         completionHandler completion: ((Bool) -> Void)? = nil)
```

```
func canOpenURL(_ url: URL) -> Bool
```


Example



ViewController



Sake of simplicity...

Code located in a
ViewController

ViewController

```
import UIKit

class ViewController: UIViewController {

    fileprivate let b1 = UIButton(type: .system)
    fileprivate let b2 = UIButton(type: .system)
    fileprivate let b3 = UIButton(type: .system)
    fileprivate let b4 = UIButton(type: .system)
    fileprivate let b5 = UIButton(type: .system)
    fileprivate let l = UILabel();
```


ViewController

```
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.
    self.view = UIView(frame:UIScreen.main.bounds)
    self.view.backgroundColor = UIColor.white
    b1.setTitle("Phone", for: .normal)
    b1.addTarget(self, action: #selector(phone), for: .touchDown)
    b1.frame = CGRect(x:Int(UIScreen.main.bounds.width) / 2 - 40,
                      y:50, width:80, height:20)

    b2.setTitle("Email", for: .normal)
    b2.addTarget(self, action: #selector(mail), for: .touchDown)
    b2.frame = CGRect(x:Int(UIScreen.main.bounds.width) / 2 - 40,
                      y:90, width:80, height:20)

    b3.setTitle("SMS", for: .normal)
    b3.addTarget(self, action: #selector(sms), for: .touchDown)
    b3.frame = CGRect(x:Int(UIScreen.main.bounds.width) / 2 - 40,
                      y:130, width:80, height:20)

    b4.setTitle("WWW", for: .normal)
    b4.addTarget(self, action: #selector(browseWeb), for: .touchDown)
    b4.frame = CGRect(x:Int(UIScreen.main.bounds.width) / 2 - 40,
                      y:170, width:80, height:20)

    b5.setTitle("Paris to Sain-Malo", for: .normal)
    b5.addTarget(self, action: #selector(map), for: .touchDown)
    b5.frame = CGRect(x:Int(UIScreen.main.bounds.width) / 2 - 80,
                      y:210, width:160, height:20)

    l.frame = CGRect(x:Int(UIScreen.main.bounds.width) / 2 - 100,
                     y:250, width:200, height:20)
    l.textAlignment = .center
}
```


ViewController

```
self.view.addSubview(b1)  
self.view.addSubview(b2)  
self.view.addSubview(b3)  
self.view.addSubview(b4)  
self.view.addSubview(b5)  
self.view.addSubview(l)
```

}

ViewController

```
@objc func phone () {
    UIApplication.shared.open(URL(string:"tel:01-44-27-44-27")!,
        options: [:], // Empty dictionary
        completionHandler: { (ok) in
            if ok {
                self.l.text = "Phone success"
            } else {
                self.l.text = "Phone failure"
            }
        })
}

func phoneVariant () { // variant
    if UIApplication.shared.canOpenURL(URL(string:"tel:01-44-27-44-27")!) {
        UIApplication.shared.open(URL(string:"tel:01-44-27-44-27")!,
            options: [:], // Empty dictionary
            completionHandler: { (ok) in
                if ok {
                    self.l.text = "Phone success"
                } else {
                    self.l.text = "Phone failure"
                }
            })
    }
}
```


ViewController

```
@objc func mail () {
    UIApplication.shared.open(URL(string:"mailto:Fabrice.Kordon@lip6.fr")!,
        options: [:], // Empty dictionary
        completionHandler: { (ok) in
        if ok {
            self.l.text = "Mail success"
        } else {
            self.l.text = "Mail failure"
        }
        })
}
```

```
@objc func sms () {
    if UIApplication.shared.canOpenURL(URL(string:"sms:06-05-04-03-02")!) {
        UIApplication.shared.open(URL(string:"sms:06-05-04-03-02")!,
            options: [:], // Empty dictionary
            completionHandler: { (ok) -> Void in
            if ok {
                self.l.text = "SMS sccess"
            } else {
                self.l.text = "SMS failure"
            }
        })
    }
}
```


ViewController

```
@objc func browseWeb () {
    if UIApplication.shared.canOpenURL(
        URL(string:"https://sorbonne-universite.fr")!) {
        UIApplication.shared.open(URL(string:"https://sorbonne-universite.fr")!,
            options: [:], // Empty dictionary
            completionHandler: { (ok) -> Void in
                if ok {
                    self.l.text = "https success"
                } else {
                    self.l.text = "https failure"
                }
            })
    }
}
```


ViewController

```
@objc func map () {  
    // From Paris to Saint-Malo  
    let directionsURL = "http://maps.apple.com/?saddr=Paris&daddr=Saint-Malo"  
    // Can be also coordinates  
    // http://maps.apple.com/?saddr=48.846441,48.846441&daddr=48.852968,2.349902  
    // Or designate current location  
    // http://maps.apple.com/?saddr=Current%%20Location&daddr=Yokohama  
    if UIApplication.shared.canOpenURL(URL(string:directionsURL)!) {  
        UIApplication.shared.open(URL(string:directionsURL)!,  
                                   options: [:], // Empty dictionary  
                                   completionHandler: { (ok) -> Void in  
                    if ok {  
                        self.l.text = "Map success"  
                    } else {  
                        self.l.text = "Map failure"  
                    }  
                })  
    }  
}
```


You may open preferences



URL format is...

- prefs:root=General&path=About
- prefs:root=General&path=ACCESSIBILITY
- prefs:root=AIRPLANE_MODE
- prefs:root=General&path=AUTOLOCK
- prefs:root=Brightness
- prefs:root=Bluetooth
- prefs:root=General&path=DATE_AND_TIME
- prefs:root=FACETIME
- prefs:root=General
- prefs:root=General&path=Keyboard
- prefs:root=General&path=INTERNATIONAL
- prefs:root=LOCATION_SERVICES
- prefs:root=ACCOUNT_SETTINGS
- prefs:root=MUSIC
- prefs:root=MUSIC&path=EQ
- prefs:root=MUSIC&path=VolumeLimit
- prefs:root=General&path=Network
- prefs:root=NIKE_PLUS_IPOD
- prefs:root=General&path=USAGE/CELLULAR_USAGE
- prefs:root=General&path=ManagedConfigurationList
- prefs:root=General&path=SOFTWARE_UPDATE_LINK
- prefs:root=General&path=USAGE
- prefs:root=VIDEO
- prefs:root=NOTES
- prefs:root=NOTIFICATIONS_ID
- prefs:root=Phone
- prefs:root=Photos
- prefs:root=General&path=Reset
- prefs:root=Sounds&path=Ringtone
- prefs:root=Safari
- prefs:root=General&path=Assistant
- prefs:root=Sounds
- prefs:root=STORE
- prefs:root=TWITTER
- prefs:root=FACEBOOK
- prefs:root=General&path=Network/VPN
- prefs:root=Wallpaper
- prefs:root=WIFI
- prefs:root=INTERNET_TETHERING
- prefs:root=Phone&path=Blocked
- prefs:root=DO_NOT_DISTURB

You may open preferences

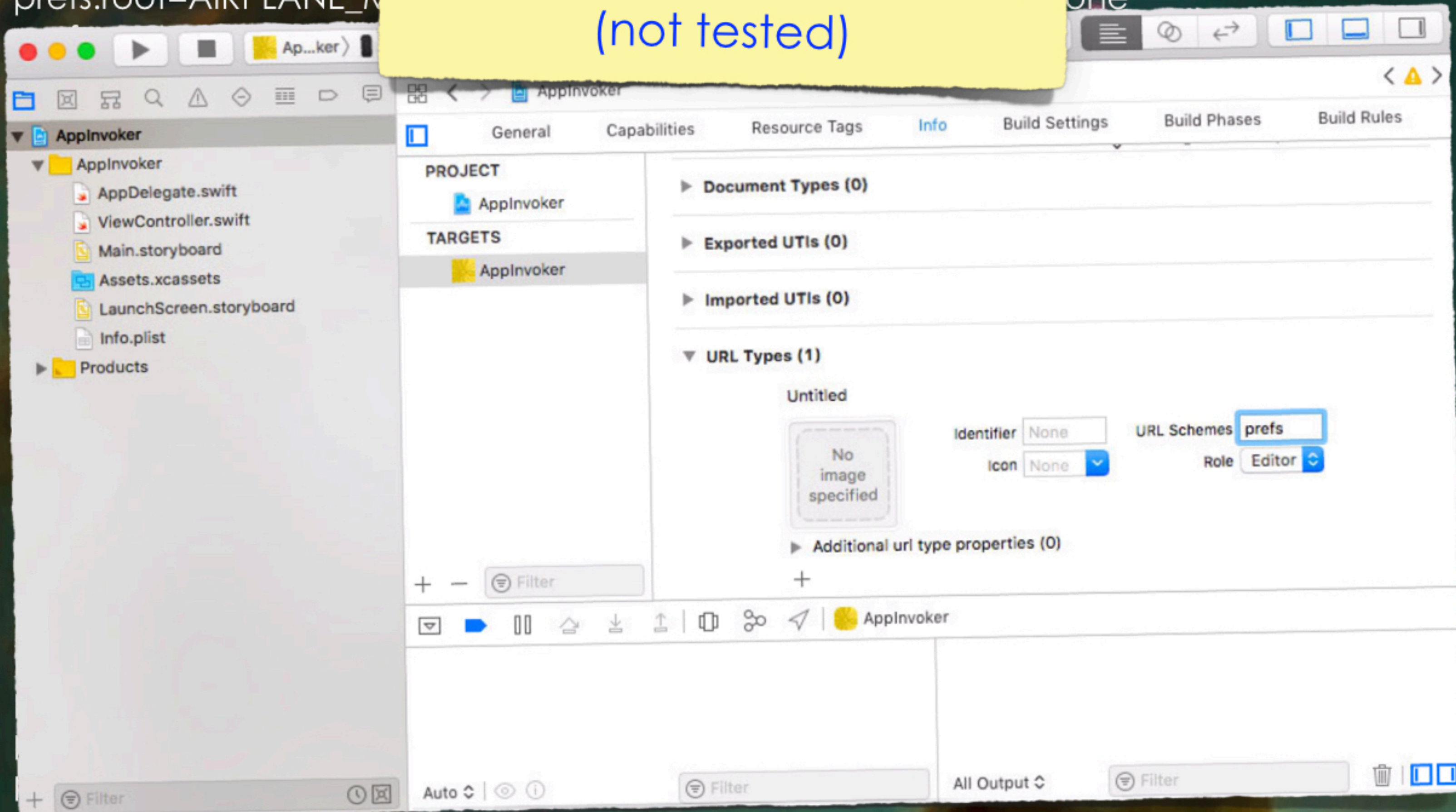
URL format



Before iOS11?
You must set an URL schema
(not tested)

prefs:root=General&po
prefs:root=General&po
prefs:root=AIRPLANE_M

OTES
NOTIFICATIONS_ID
one



A bit more...

7

📱 Possibly some generalization

- 👤 Needs the target application some work

📱 What to be done?

- 👤 Declaration of an URL schema in the info.plist
 - ▶ `LSApplicationQueriesSchemes`
 - ▶ Declare the URL to be invoked
- 👤 Support of a method in the `UIApplicationDelegate`

```
func application(_ app: UIApplication,  
                open url: URL,  
                options: [UIApplication.OpenURLOptionsKey : Any] = [:]) -> Bool
```

```
- (BOOL)application:(UIApplication *)app  
    openURL:(NSURL *)url  
    options:(NSDictionary<UIApplicationOpenURLOptionsKey, id> *)options;
```


A bit more...

Possibly some generalization

- Needs the target application some work

What to be done?

- Declaration of an URL schema in the info.plist
 - ▶ LSApplicationQueriesSchemes
 - ▶ Declare the URL to be invoked
- Support of a method in the UIApplicationDelegate

```
func application(_ app: UIApplication,
                 open url: URL,
                 options: [UIApplication.OpenURLOptionsKey : Any] = [:]) -> Bool
```




```
- (BOOL)application:(UIApplication *)
  openURL:(NSURL *)
  options:(NSDictionary<UIApplicationOpenURLOptionsKey, id> *)options;
```



As a conclusion...



Practical

- Once upon a time limited compared to 
- Provides much more facilities now
- Mainly useful for well known applications
 - ▶ Apple's ones
 - ▶ Known ones (e.g. twitter?)

You must «play» with it!

