



**INSTITUTO FEDERAL**  
Paraíba

Campus  
João Pessoa



**Alunos: Adriano Amaral, Renatha Victor**

## My apps

[adriano amaral](#) ▾[Create app](#)

ProjetoIOS

Status: Development

Permission type: Full Dropbox

Primeiro devemos registrar o aplicativo no Dropbox que será associado às chamadas da API.

## Create a new app on the DBX Platform

### 1. Choose an API

- ☒ **Dropbox API**  
For apps that need to access files in  
Dropbox. [Learn more](#)



- ☐ **Dropbox Business API**  
For apps that need access to Dropbox  
Business team info. [Learn more](#)



### 2. Choose the type of access you need

[Learn more about access types](#)

- ☐ App folder – Access to a single folder created specifically for your app.
- ☒ Full Dropbox – Access to all files and folders in a user's Dropbox.

### 3. Name your app

## ProjetoIOS

Settings

Branding

Analytics

Status

Development

Apply for production

Development users

Only you

Enable additional users

Permission type

Full Dropbox ⓘ

App key

gmo6f6qjd26o9ct

App secret

[Show](#)

OAuth 2

## Redirect URIs

[Add](#)

## Allow implicit grant ⓘ

## Generated access token ⓘ

[Generate](#)

Chooser/Saver domains

[Add](#)If using the [Chooser](#) or the [Saver](#) on a website, the domain of that site.

Depois de criarmos o projeto no Dropbox, precisamos instalar o gerenciador de dependências.

```
$ gem install cocoapods
```

Após a instalação do cocoapods, é necessário ir na pasta do projeto e executar o seguinte comando para a criação do arquivo “Podfile”:

```
$ pod init
```

Caso já exista, abra-o para adicionar a seguinte dependência:

```
pod 'SwiftyDropbox'
```

Seu arquivo deve ficar parecido com esse:  
use\_frameworks!

```
target 'ProjetoIOS' do  
  pod 'SwiftyDropbox'  
end
```



# SwiftlyDropbox 5.1.0

By Stephen Cobbe

 [dropbox/SwiftyDropbox](#)

## Dropbox for Swift

The Official Dropbox Swift SDK for integrating with Dropbox [API v2](#) on iOS or macOS.

Full documentation [here](#).

Tested	X
Language	Swift
License	MIT
Last Release	Jun 2019
Supports SPM	✓

Maintained by [Ryan Pearl](#),  
[Dropbox, Inc.](#), [Chris Varenhorst](#),  
[Will Kiefer](#).

Installation Guide

Depois só basta executar dois comandos para instalar as dependências e as atualizações da SDK do Dropbox.

```
$ pod install  
$ pod update
```

Para configurar o projeto, se estiver usando a versão iOS SDK 9.0, é preciso adicionar no Info.plist as seguintes linhas, que irá identificar se existe um app do Dropbox instalado, gerando o token de acesso OAuth 2.0.

```
<key>LSApplicationQueriesSchemes</key>  
  <array>  
    <string>dbapi-8-emm</string>  
    <string>dbapi-2</string>  
  </array>
```

OBS.: Existem duas formas, pelo IOS, de obter um token de acesso OAuth 2.0, através do aplicativo(se instalado), ou através do Safari(navegador), o que é mais seguro, pois os dados ficam salvos na sessão, evitando que o usuário precise reinserir seus dados



Depois é preciso adicionar o seguinte código, também ao Info.plist, para o app poder se registrar e manipular o esquema de redirecionamento de URL do Dropbox, substituindo onde tem “<APP\_KEY>” pela App key do projeto.

```
<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>db-<APP_KEY></string>
    </array>
    <key>CFBundleURLName</key>
    <string></string>
  </dict>
</array>
```





No AppDelegate, é iniciado uma instância do DropboxClient usando a App key do seu projeto.

```
import SwiftyDropbox
```

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
```

```
    DropboxClientsManager.setupWithAppKey("<APP_KEY>")
```

```
    return true
```

```
}
```

OBS.: É necessário abrir o projeto no Xcode com o xcworkspace e não com o padrão xcodeproj.

No controller é adicionado um botão para realizar o login.

```
func myButtonInControllerPressed() {  
    DropboxClientsManager.authorizeFromController(UIApplication.shared,  
        controller: self,  
        openURL: { (url: URL) -> Void in  
            UIApplication.shared.openURL(url)  
        })  
}
```

Para poder manipular a volta da aplicação após a autenticação, é possível adicionar esse código também no AppDelegate.

```
func application(_ app: UIApplication, open url: URL, options: [UIApplication.OpenURLOptionsKey : Any] = [:]) -> Bool {  
    if let authResult = DropboxClientsManager.handleRedirectURL(url) {  
        switch authResult {  
        case .success:  
            print("Success! User is logged into Dropbox.")  
        case .cancel:  
            print("Authorization flow was manually canceled by user!")  
        case .error(_, let description):  
            print("Error: \(description)")  
        }  
    }  
    return true  
}
```

## Referências:

- <https://github.com/dropbox/SwiftyDropbox>
- <https://cocoapods.org/pods/SwiftyDropbox>

