



Updates

*Automatically detect app updates &
gently prompt users to upgrade*

Updates

- Sometimes users need a gentle nudge to update:
 - Important new feature
 - Urgent bug fix
 - Certificate used for SSL pinning expiring
- Updates is a framework for iOS that detects app updates and gently prompts users to upgrade.

Remote Configuration

- Specify the configuration URL (accepts remote / file URLs):
 - `Updates.configurationURL = URL(string: "https://.../updates.json")`
- Otherwise will look in app bundle for `Updates.json`.
- `Updates.json` contains configuration settings:

```
{  
  "updates": {  
    "check-for": "automatically",  
    "notify": "once"  
  }  
}
```

Simple Example

```
Updates.checkForUpdates { updateAvailable in  
    if updateAvailable {  
        print("New version available!")  
    }  
}
```

Automatic Mode

- Uses iTunes Search API to retrieve app meta data using app's **bundle identifier**.
- Retrieves:
 - Latest version of app available in App Store.
 - Minimum version of iOS required.
 - Numeric **App Store identifier**.

Manual Mode

- Retrieves the same information but from the configuration:
 - Latest version of app available in App Store.
 - Minimum version of iOS required.
 - Numeric **App Store identifier**.

```
{  
  "updates": {  
    "check-for": "manually",  
    "notify": "once",  
    "app-store-id": 123456,  
    "min-os-required": "9.0",  
    "version": "5.0.1"  
  }  
}
```

Notifications

- If user doesn't upgrade following a prompt, options exists to notify the user a specific number of times.
- Options are:
 - `.once`
 - `.twice`
 - `.thrice`
 - `.always`
 - `.never`

Programmatic Config

- Can alternatively be configured programmatically:
 - `Updates.updateMode = .automatically`
- Or:
 - `Updates.updateMode = .manually`
 - `Updates.appStoreId = "123456"`
 - `Updates.minOSRequired = "9.0"`
 - `Updates.versionString = "5.0.1"`

Updates UI

- An optional UI component to help users update:
 - `UIAlertController` to ask user to update
 - If yes, present `SKStoreProductViewController`
 - In the event product information fails to load, App Store presented via `SFSafariViewController`
- Why separate Updates and UpdatesUI?
 - You may want to write customer UI
 - Updates available through **Swift Package Manager** whereas UpdatesUI is not (due to dependency on iOS frameworks)

Example

```
Updates.checkForUpdates(.automatically, comparingVersions: .major,  
notifying: .twice) { updateAvailable in  
    if updateAvailable {  
        UpdatesUI.presentAppStore(animated: animated)  
    }  
}
```

Versions can be:

- major
- minor
- patch

API

- `Updates.configurationURL` used to set URL to remote or bundled configuration JSON .
- `Updates.checkforUpdates` which accepts a completion handler returning a Bool indicating whether or to let the user know that a new version is available.
- `Updates.countryCode` set automatically but can be overridden manually - used to lookup iTunes Search API metadata for the correct App Store locale.
- `Updates.productName` set automatically but can be overridden manually - used as part of iTunes Search API metadata lookup.

Updates

<https://github.com/rwbutler/Updates>

- Cocoapods

`pod "Updates"`

- Carthage

`github "rwbutler/Updates"`