CountThings from Photos

InterApp Communication

CountThings from Photos app is prepared to respond to action requests by specifying the appropriate url scheme. This document shows you how to create an x-callback-url scheme for a particular action, how to use it to start a callback in CountThings from Photos app and how to get the results back in your app.

Prerequisites

- iOS app that will call CountThings from Photos
- Install CountThings from Photos and login to an account with a valid subscription
- Install pod 'CallbackURLKit' (used in this document to handle url parsing)

General

The url can be configured in order to:

- Select first screen when CountThings from Photos app will open:
  - Take Picture screen
  - Prepare screen (works only if the URL of the image is not private)
  - Set Active Template screen
- Select settings:
  - Start count process automatically when opening CountThings from Photos (optional)
  - Send the Template ID if the current active template is not suitable for the current count process (optional)
- Select which result information will return to the caller app:
  - JSON Data (object that contains information about results - total number, coordinates, radius and text label)
  - Results image (the caller app will receive the image in memory using the UIPasteboard.image property) after the **Save** button is pressed on **Review screen**.

Usage

Define the following enum structure to select the first when when CountThings from Photos app will open:

```
enum OpenFirstScreen: String {
    case takePicture, prepare, setActiveTemplate
}
```

Subclass the Client class from CallbackURLKit pod and call superclass init with “countthings” as urlScheme

```
class CountThings : Client {
    public init() {
        super.init(urlScheme: “countthings")
    }
}
```

Create a function with this signature:

```
public func open(screen: OpenFirstScreen, countAutomatically: Bool
= false, templateId: Int, imageUrl : URL? = nil,
        onSuccess: SuccessCallback? = nil, onFailure:
FailureCallback? = nil, onCancel: CancelCallback? = nil) throws
```

Define the parameters structure:

```
var parameters = [String:String]()
if let url = imageUrl {
   parameters = ["imgUrl" : url.absoluteString]
}
parameters["templateId"] = "\(templateId)"
parameters["countAutomatically"] = countAutomatically.description
```

Call the CountThings from Photos app:

```
try self.perform(action: screen.rawValue, parameters: parameters,
                onSuccess: onSuccess, onFailure: onFailure,
onCancel: onCancel)
```

In order to retrieve the results from CountThings from Photos app a pooling timer must be implemented as following:

```
var timer:Timer?
static let notificationName =
Notification.Name("onPasteboardChanged")
var lastChangeCount: Int = UIPasteboard.general.changeCount
```

```
public init() {
    super.init(urlScheme: "countthings")

    self.timer = Timer.scheduledTimer(withTimeInterval: 0.05,
repeats: true) { (t) in
            if self.lastChangeCount !=
UIPasteboard.general.changeCount {
                self.lastChangeCount =
UIPasteboard.general.changeCount
                NotificationCenter.default.post(name:
CountThings.notificationName, object: UIPasteboard.general)
            }
        }
 }
```

The results are pasted in general UIPasteboard and the results are available when the caller app didBecome active again. The results can be accessed as following:

```
 @objc func clipboardChanged(_ notification: Notification){
        let pasteboardJson: String? = UIPasteboard.general.string
        let pasteboardImage: UIImage? = UIPasteboard.general.image
        if let theString = pasteboardJson {
            // Json returned as string
        }
        if let theImage = pasteboardImage {
            // Results image returned as UIImage
        }
    }
```

Questions or suggestions should be reported at support@countthings.com