First post from my iPhone app http://www.appcoda.com

Album iOS Photos

Location None

Audience Friends

CHAPTER 5

HOW TO INTEGRATE TWITTER AND FACEBOOK SHARING

Learn how to integrate Twitter and Facebook sharing in your apps

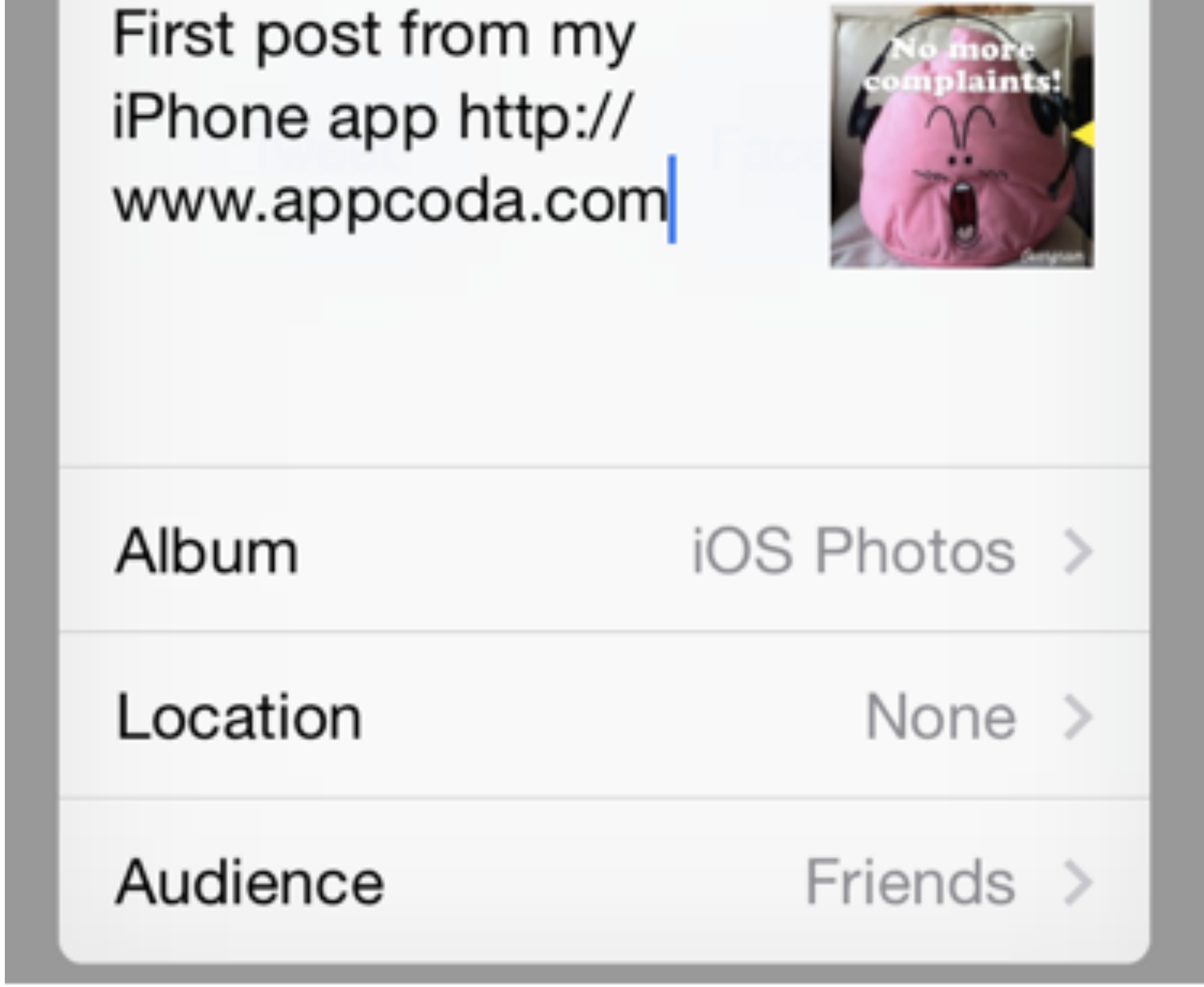
SOCIAL FRAMEWORK OVERVIEW

With the advent of social networks, you may want to provide social network sharing in your apps. In the past, developers have had to make use of the Facebook and Twitter API (or other social networks) in order to implement the sharing feature.

Since the release of iOS 6, Apple introduced a new framework known as *Social Framework*. The Social framework lets you integrate your app with any supported social networking service. Currently, it supports Facebook, Twitter, Sina Weibo, and Tencent Weibo. The framework gives you a standard composer to create posts for different social networks, and shields you from learning the APIs of the social networks. You don't even need to know how to initiate a network request or handle single sign-on. The Social Framework simplifies everything. You just need to write a few lines of code to bring up the composer for users to tweet / publish Facebook posts within your app.

The framework comes with a very handy class called *SLComposeViewController*. Similar to the MFMailComposeViewController, the SLComposeViewController class presents a standard view controller for users to compose tweets or Facebook posts. It also allows developers to preset the initial text, attach images, and add a URL to the post. If you just want to implement a simple sharing feature, this is the only class you need to know.

If you're not familiar with the SLComposeViewController, the sample shown on your right is what it looks like within your app.

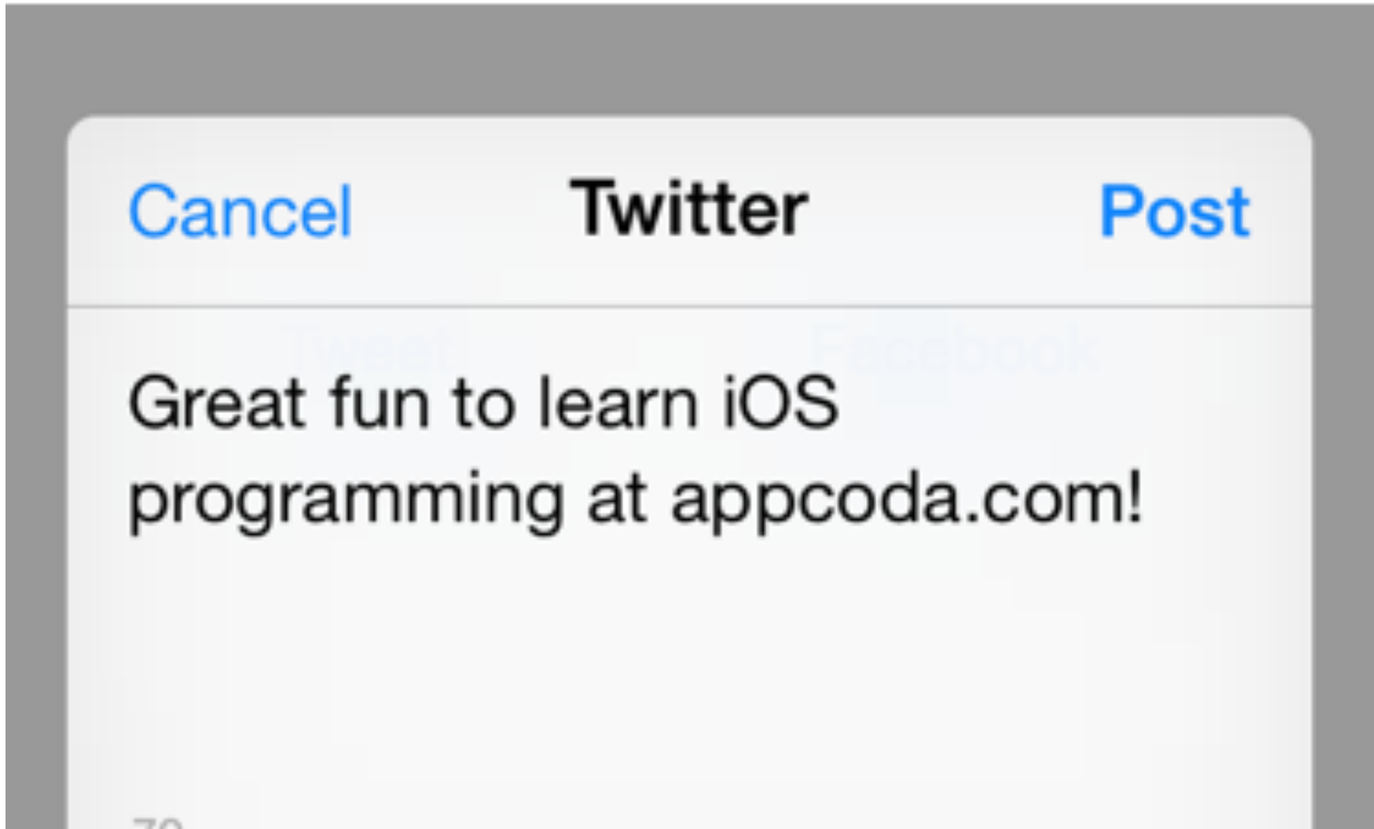


First post from my iPhone app http://www.appcoda.com

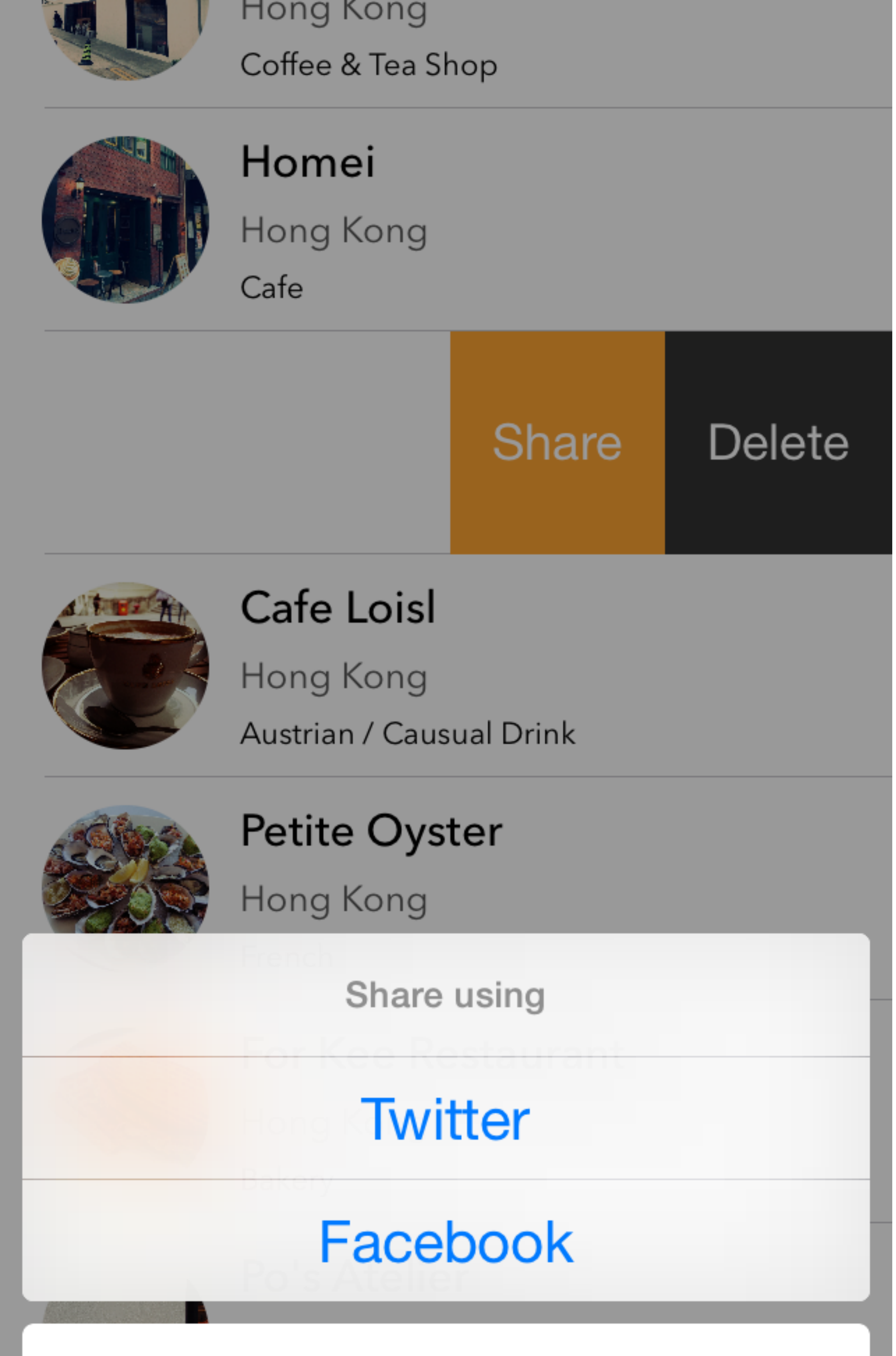
Album iOS Photos >

Location None >

Audience Friends >



Great fun to learn iOS programming at appcoda.com!



CREATE THE DEMO PROJECT AND DESIGN THE INTERFACE

Now you should have a basic idea about the framework. Let's get started and see how to add Twitter and Facebook sharing in a sample app.

To begin, you can download the project template from https://www.dropbox.com/s/v2k0sy25x4txsbq/SocialDemoTemplate.zip?dl=0. You should be very familiar with the sample project if you read the beginner book. The app displays a list of restaurants in the main screen. When a user swipes a cell and taps the Share button, the app allows the user to share the selected restaurant on Facebook or Twitter.

The Facebook and Twitter sharing features are not yet implemented in the template. And these are what we're going to work on.

ADDING TWITTER SUPPORT

Let's start with the implementation of a Twitter button. Open the RestaurantTableViewController.swift and look into the *editActionsForRowAtIndexPath* method. You should find the code snippet shown below that instantiates the UIAlertAction instances of Twitter and Facebook actions. For both UIAlertAction instances, the handler is set to nil. Now we'll implement the twitterAction for users to tweet:

```
var shareAction = UITableViewRowAction(style: UITableViewRowActionStyle.Default, title: "Share", handler: { (action:UITableViewRowAction!,
indexPath:NSIndexPath!) -> Void in

    let shareMenu = UIAlertController(title: nil, message: "Share using", preferredStyle: .ActionSheet)
    let twitterAction = UIAlertAction(title: "Twitter", style: UIAlertActionStyle.Default, handler: nil)
    let facebookAction = UIAlertAction(title: "Facebook", style: UIAlertActionStyle.Default, handler: nil)
    let cancelAction = UIAlertAction(title: "Cancel", style: UIAlertActionStyle.Cancel, handler: nil)

    shareMenu.addAction(twitterAction)
    shareMenu.addAction(facebookAction)
    shareMenu.addAction(cancelAction)

    self.presentViewController(shareMenu, animated: true, completion: nil)
    }
)
```

Because the SLComposeViewController class is provided by the Social framework, you have to import the Social framework at the very top of the RestaurantTableViewController class:

```
import Social
```

Next, update the twitterAction variable to the following:

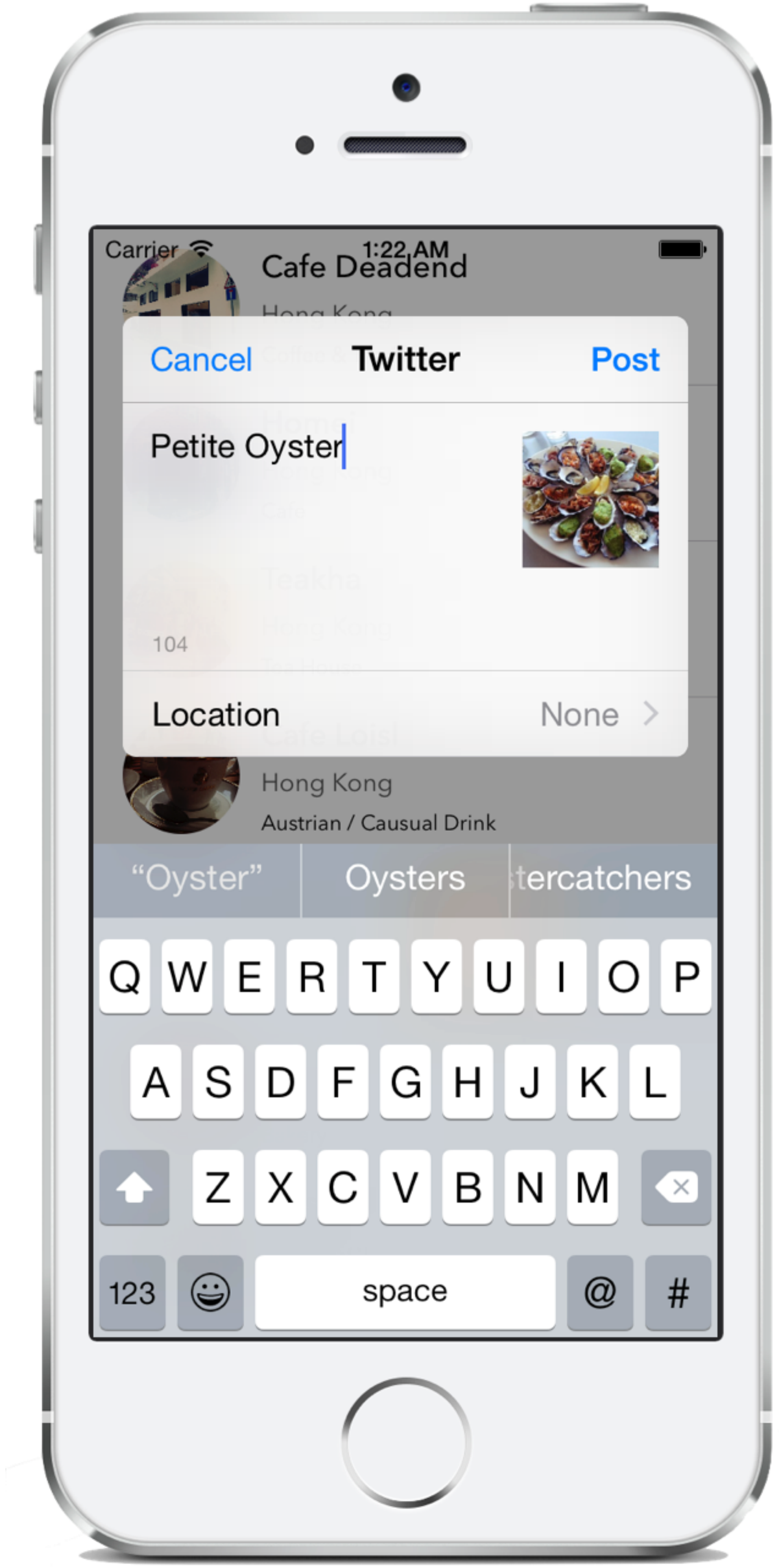
```
let twitterAction = UIAlertAction(title: "Twitter", style: UIAlertActionStyle.Default, handler: { (action) -> Void in

    if SLComposeViewController.isAvailableForServiceType(SLServiceTypeTwitter) {
        let tweetComposer = SLComposeViewController(forServiceType: SLServiceTypeTwitter)
        tweetComposer.setInitialText(self.restaurantNames[indexPath.row])
        tweetComposer.addImage(UIImage(named: self.restaurantImages[indexPath.row]))
        self.presentViewController(tweetComposer, animated: true, completion: nil)
    } else {
        let alertMessage = UIAlertController(title: "Twitter Unavailable", message: "You haven't
registered your Twitter account. Please go to Settings > Twitter to create one.",
preferredStyle: .Alert)
        alertMessage.addAction(UIAlertAction(title: "OK", style: .Default, handler: nil))
        self.presentViewController(alertMessage, animated: true, completion: nil)
    }
})
```

Before testing the app, let's go through the above code line by line. First, we use the *isAvailableForServiceType* method to verify whether or not the Twitter service (SLServiceTypeTwitter) is available. One reason why users can't access the Twitter service is because they haven't signed into their Twitter accounts in Settings. If the Twitter service is unavailable, we simply prompt an error message and instruct the user to sign on the account in iOS.

If the service is accessible, we then create an instance of the SLComposeViewController of the Twitter service, followed by setting the initial text and image in the composer.

Lastly, we invoke the *presentViewController* method to bring up the Twitter composer.

That's the code we need to let users tweet within your app. It's much easier than you thought, right? Cool! It's ready to go. Hit the Run button to compile and execute the app. Swipe a restaurant record and tap the Share button. Once you select Twitter, the app shows you a Tweet composer, populated with the restaurant name and image.



Petite Oyster

Location None

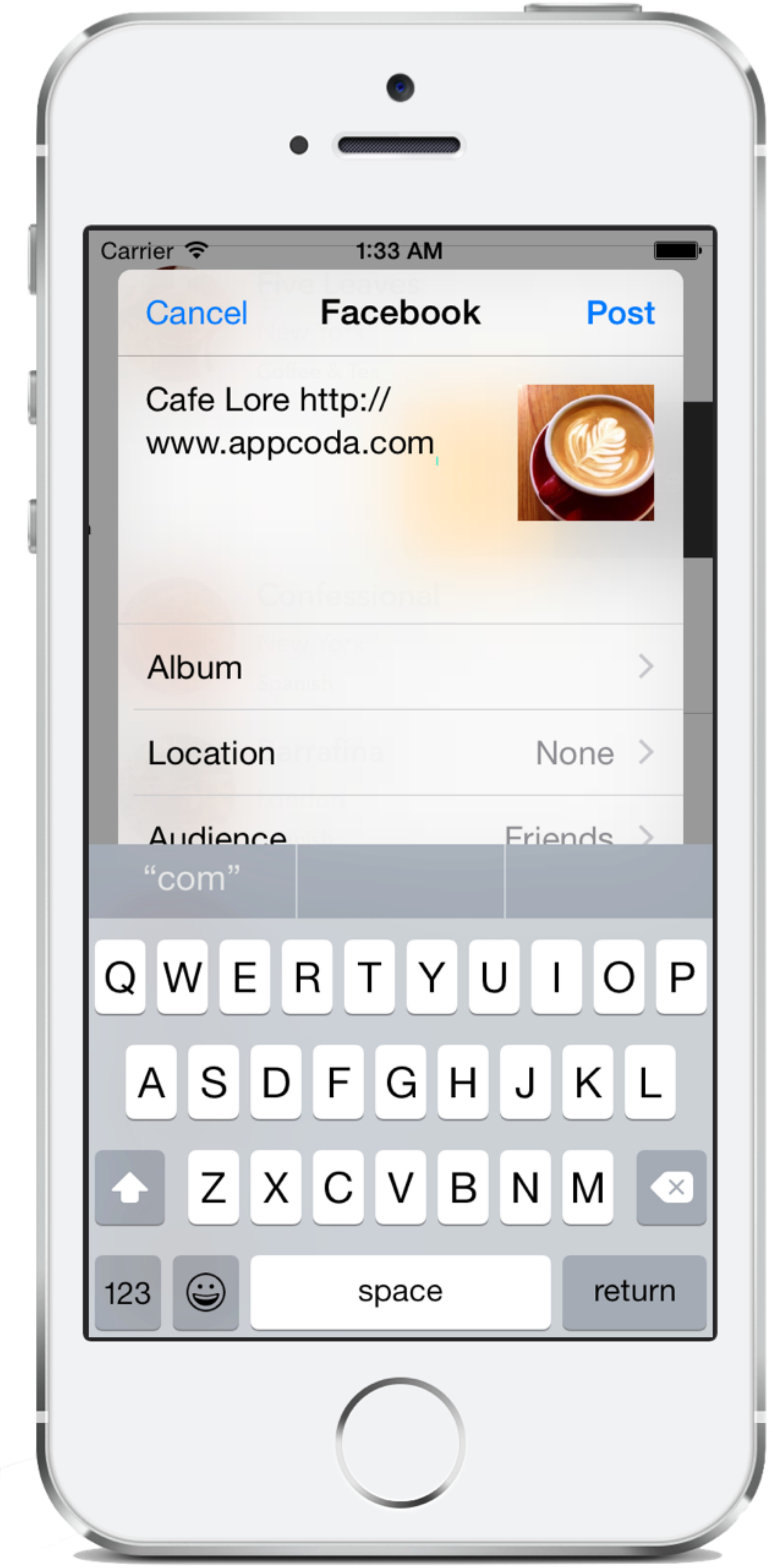
ADDING FACEBOOK SUPPORT

Next, we'll implement the Facebook action for publishing a wall post on Facebook. In the editActionsForRowAtIndexPath method of RestaurantTableViewController.swift, replace the facebookAction with the following code:

```
let facebookAction = UIAlertAction(title: "Facebook", style: UIAlertActionStyle.Default, handler:
{ (action) -> Void in

    if SLComposeViewController.isAvailableForServiceType(SLServiceTypeFacebook) {
        let facebookComposer = SLComposeViewController(forServiceType: SLServiceTypeFacebook)
        facebookComposer.setInitialText(self.restaurantNames[indexPath.row])
        facebookComposer.addImage(UIImage(named: self.restaurantImages[indexPath.row]))
        facebookComposer.addURL(NSURL(string: "http://www.appcoda.com"))

        self.presentViewController(facebookComposer, animated: true, completion: nil)
    } else {
        let alertMessage = UIAlertController(title: "Facebook Unavailable", message: "You haven't
registered your Facebook account. Please go to Settings > Facebook to sign in or create one.",
preferredStyle: .Alert)
        alertMessage.addAction(UIAlertAction(title: "OK", style: .Default, handler: nil))
        self.presentViewController(alertMessage, animated: true, completion: nil)
    }
})
```

That's it. The code is very similar to the code we've used in the *twitterAction*. The only change is the service type. Instead of using *SLServiceTypeTwitter*, we tell SLComposeViewController to use *SLServiceTypeFacebook*. Further, we add a URL to the composer by calling the *addURL* method. Like the initial text and image, the URL is optional.

Let's run the app again and click the "Facebook" button. Your app should bring up the composer for publishing a Facebook post.



Cafe Lore http://
www.appcoda.com

SUMMARY

As you can see from this chapter, it's pretty easy to add Twitter and Facebook features using the Social Framework. If you're building your app, there is no reason why you shouldn't incorporate these social features.

The tutorial introduces the basics of Facebook and Twitter integration. You can try to tweak the sample app and upload multiple images to the social networks. However, if you want to access more advanced features such as displaying a user's Facebook friends, you'll need to make use of the Facebook API.

For your reference, you can download the complete Xcode project from https://www.dropbox.com/s/sky3t2b6kuajwyu/SocialDemo.zip?dl=0.