WILEY

Copyright © 2015 by John Wiley & Sons, Inc.
All rights reserved.

Copyright © 2015 by John Wiley & Sons, Inc. All rights reserved.
No part of this may be reproduced, used, stored in a retrieval system or transmitted in any form by any means without prior authorization or written permission of the Publisher.

Accounts和Social框架

互连

同社交媒体工具整合的应用

- iOS 5 – **Twitter**框架

- iOS 6 – **Social**框架

- iOS 8 – 支持社交框架工具：**推特、脸书、新浪微博**和**腾讯微博**

- Accounts框架
- Social框架

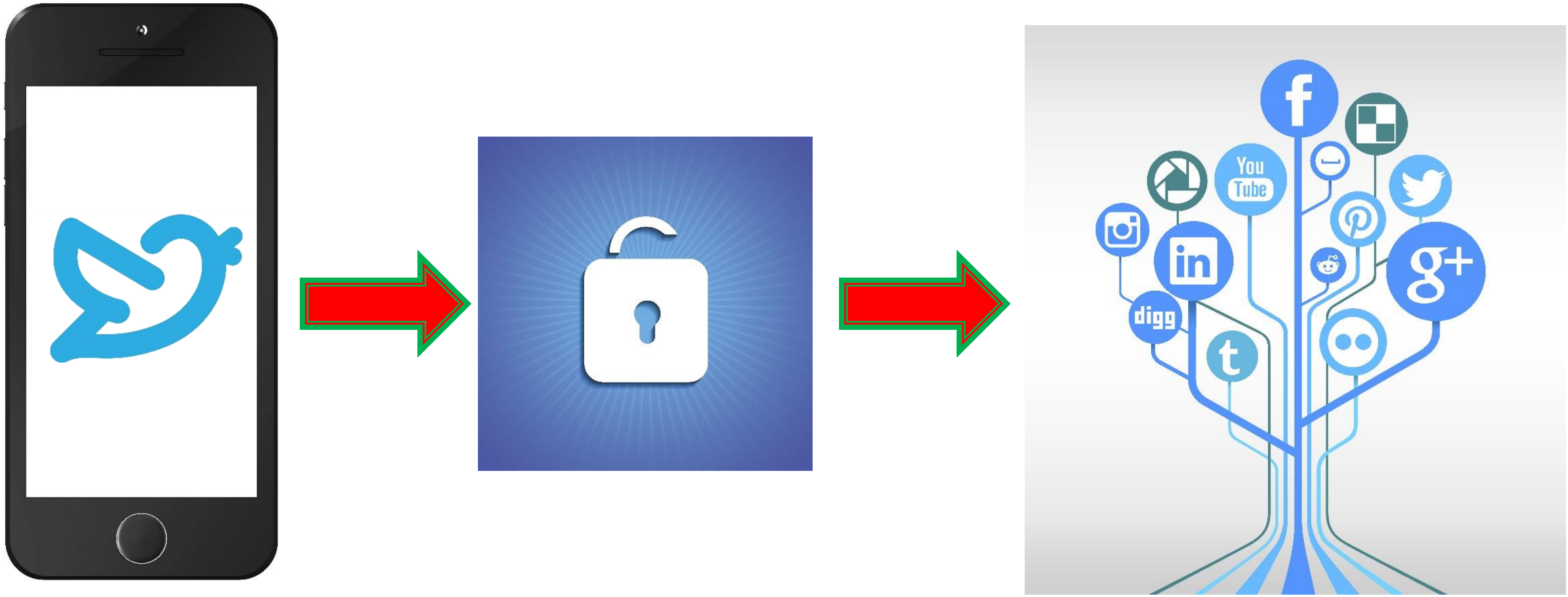
解释Accounts框架

解释Social框架

Accounts框架

- Accounts框架提供 Accounts 数据库内帐户数据的轻松访问
- 帐户存储着特定服务的登陆凭证

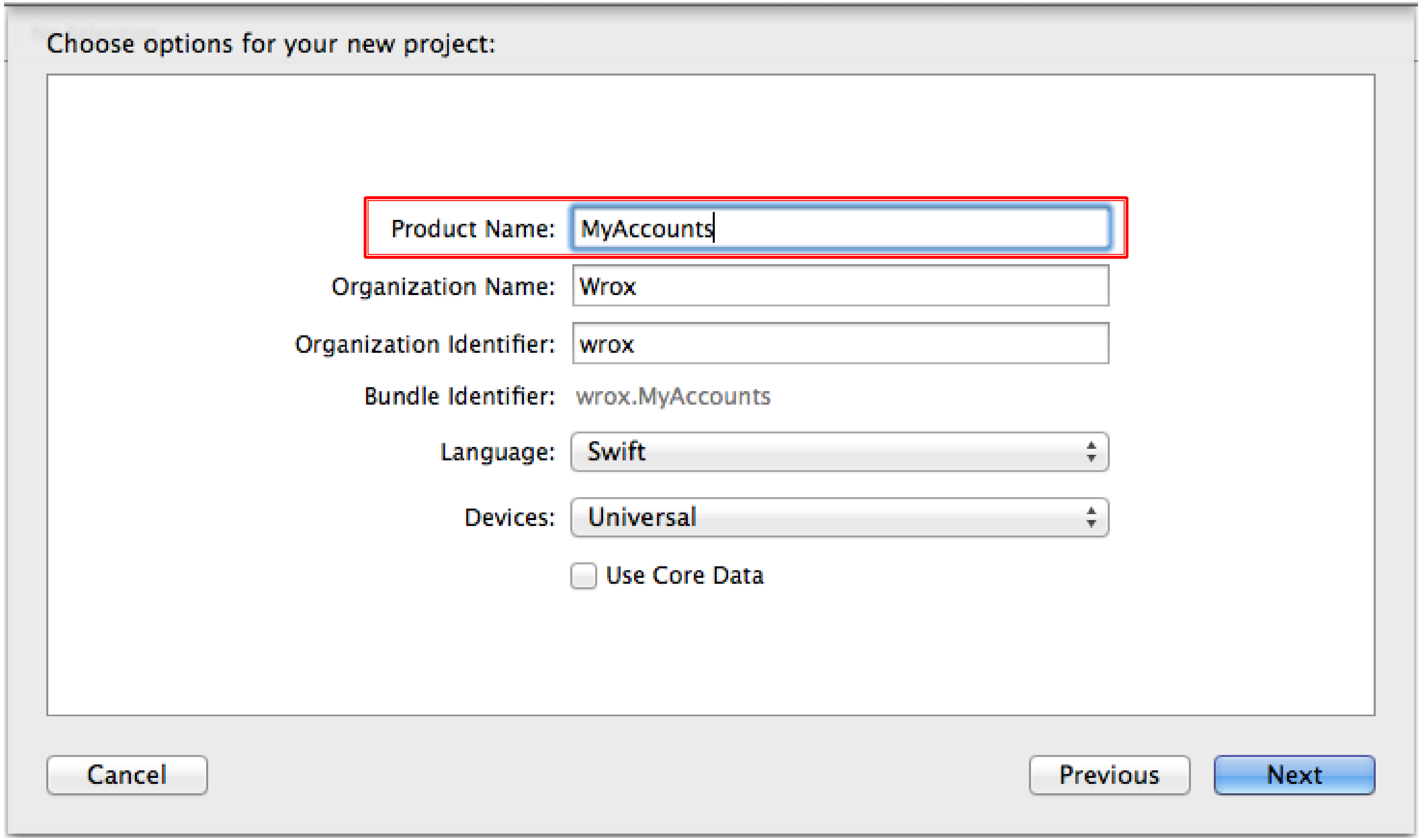
Accounts框架



Accounts框架类

- ACAccount
- ACAccountCredential
- ACAccountStore
- ACAccountType

Accounts框架



Choose options for your new project:

Product Name: MyAccounts

Organization Name: Wrox

Organization Identifier: wrox

Bundle Identifier: wrox.MyAccounts

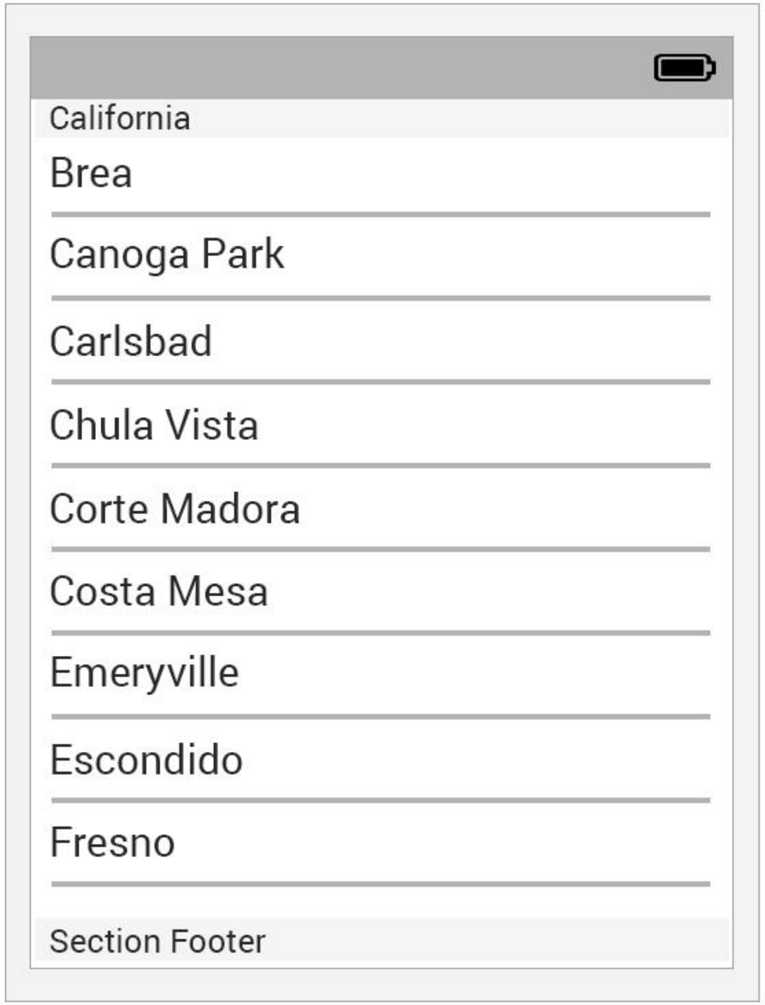
Language: Swift

Devices: Universal

□ Use Core Data

Accounts框架

- 使用UITableView创建用户界面
- 打开YDViewController.xib文件



California

Brea

Canoga Park

Carlsbad

Chula Vista

Corte Madora

Costa Mesa

Emeryville

Escondido

Fresno

Section Footer

Accounts框架

```
#import <UIKit/UIKit.h>
#import <Accounts/Accounts.h>
@interface YDViewController : UIViewController
@property (weak, nonatomic) IBOutlet UITableView *mTableView;
@property(nonatomic,strong) ACAccountStore* accountStore;
@property(nonatomic,strong) ACAccountType* twitterAccountType;
@property(nonatomic,strong) NSArray* twitterAccounts;
@end
```

Accounts框架

```
-(void)requestAccessToAccountStore
{
self.accountStore = [[ACAccountStore alloc] init];
self.twitterAccountType = [self.accountStore
accountTypeWithAccountTypeIdentifier:
ACAccountTypeIdentifierTwitter];
NSDictionary* options = nil;
[self.accountStore
    requestAccessToAccountsWithType:
self.twitterAccountType options:options
completion:^(BOOL granted, NSError *error)
{
if (granted == YES)
{
// do what you need to do here
dispatch_async(dispatch_get_main_queue(),^{
//call loadAccounts here now you know the user
    has granted access
[self loadAccounts];
});
}
else
{
dispatch_async(dispatch_get_main_queue(),^{
[self showAuthorizationError];
});
}
}];
}
```

Accounts框架

```
-(void)requestAccessToAccountStore
{
self.accountStore = [[ACAccountStore alloc] init];
self.twitterAccountType = [self.accountStore
accountTypeWithAccountTypeIdentifier:
ACAccountTypeIdentifierTwitter];
NSDictionary* options = nil;
[self.accountStore
    requestAccessToAccountsWithType:
self.twitterAccountType options:options
completion:^(BOOL granted, NSError *error)
{
if (granted == YES)
{
```

```
// do what you need to do here
dispatch_async(dispatch_get_main_queue(),^{
//call loadAccounts here now you know the user
    has granted access
[self loadAccounts];
});
}
else
{
dispatch_async(dispatch_get_main_queue(),^{
[self showAuthorizationError];
});
}
}];
}
```

Accounts框架

NSDictionary:

- 推特帐户 – 传递一个nil值
- 脸书或新浪微博帐户 – 传递选项

NSInvalidArgumentException

Accounts框架

```
NSDictionary *options = @{
ACFacebookAppIdKey:
    @"012345678901234",
ACFacebookPermissionsKey:
    @[@"publish_stream",
    @"publish_actions"],
ACFacebookAudienceKey:
    ACFacebookAudienceFriends };
```

```
ACFacebookAudienceEveryone

ACFacebookAudienceFriends

ACFacebookAudienceOnlyMe
```

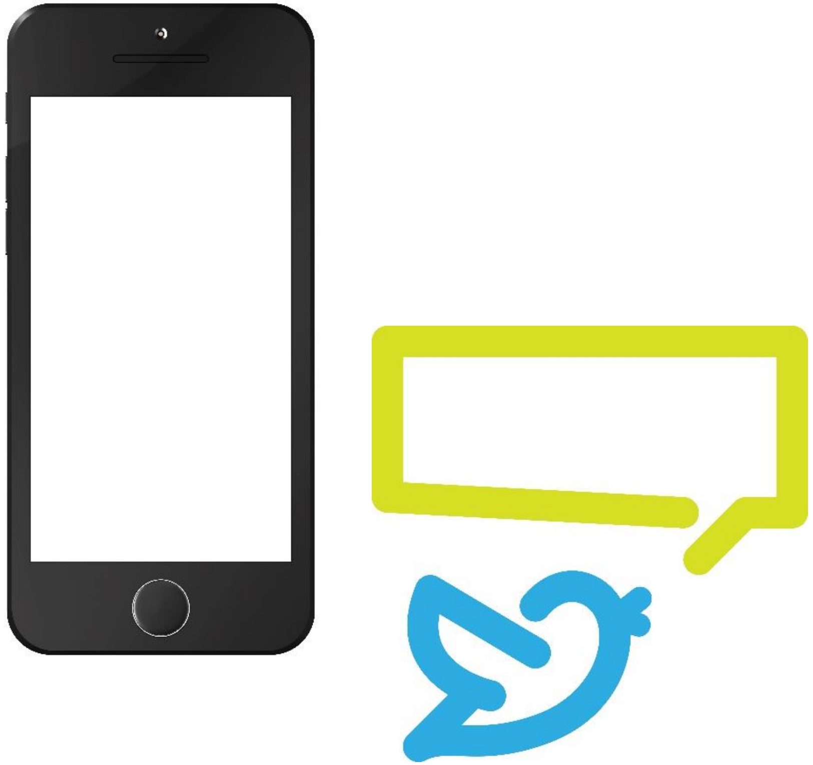
Accounts框架

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
return 1;
}
- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)sectionIndex
{
return [self.twitterAccounts count];;
}
- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
static NSString *CellIdentifier = @"Cell";
```

Accounts框架

```
UITableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:CellIdentifier];
if (cell == nil)
{
cell = [[UITableViewCell alloc]
initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:CellIdentifier];
}
//Get the currentAccount from the NSArray
ACAccount* currentAccount = [self.twitterAccounts objectAtIndex:indexPath.row];
//Display the accountDescription
cell.textLabel.text = currentAccount.accountDescription;
return cell;
}
```

Social框架



- **SLComposeViewController类**
- **SLRequest类**
- 两个类都能帮助发帖

使用SLComposeViewController

```
- (void) viewDidLoad
{
[super viewDidLoad];
// Do any additional setup after loading the view, typically from a nib.
UIBarButtonItem *addEventButtonItem =
[[UIBarButtonItem alloc] initWithBarButtonSystemItem:
UIBarButtonSystemItemAdd target:self action:@selector(newTweet:)];
self.navigationItem.rightBarButtonItem = addEventButtonItem;
}
- (void) newTweet:(id)sender
{
if ([SLComposeViewController isAvailableForServiceType:SLServiceTypeTwitter])
```

使用SLComposeViewController

```
{
//Start by setting up a completion handler
SLComposeViewControllerCompletionHandler __block
completionHandler=^(SLComposeViewControllerResult
    result){
//dismiss the view controller
[self.composeVC dismissViewControllerAnimated:YES
    completion:nil];
switch(result){
case SLComposeViewControllerResultCancelled:
{
//Cancel action
}
```

```
break;
case SLComposeViewControllerResultDone:
{
UIAlertView *alertView = [[UIAlertView alloc]
initWithTitle:@"Congratulations"
message:@"Your message have been posted to
    Twitter."
delegate:nil cancelButtonTitle:@"OK"
    otherButtonTitles:nil];
[alertView show];
}
break;
}};
```

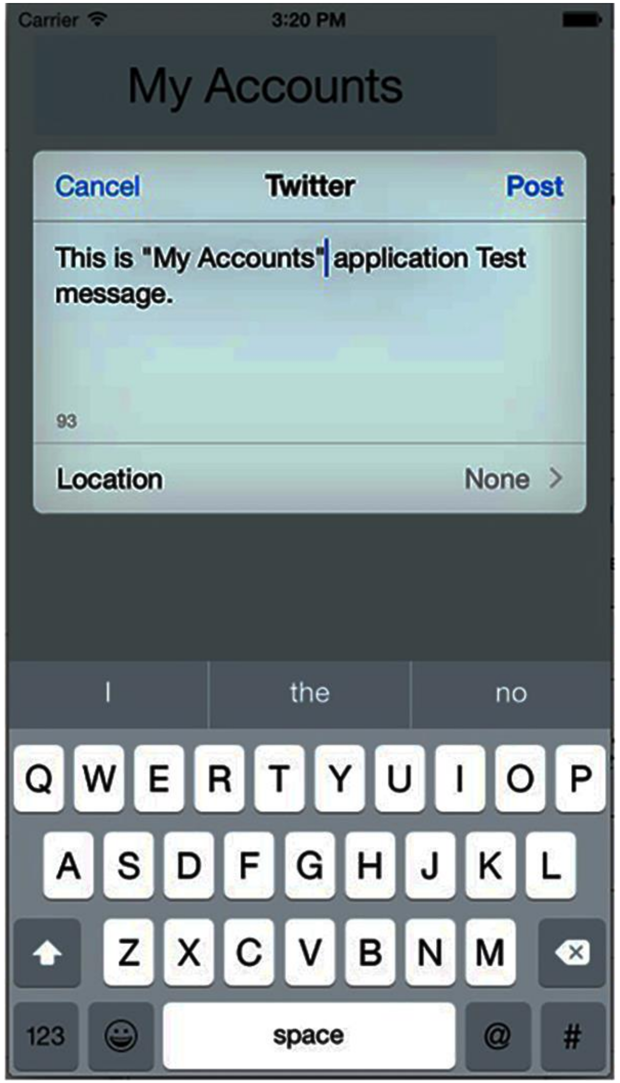
使用SLComposeViewController

```
// Initialize Compose View Controller
self.composeVC = [SLComposeViewController
composeViewControllerForServiceType:SLServiceTypeTwitter];
// Configure the compose View Controller
[self.composeVC setInitialText:@"A plane just flew over my head."];
UIImage* imageToPost = [UIImage imageNamed:@"plane.jpg"];
[self.composeVC addImage:imageToPost];
[self.composeVC addURL:[NSURL
URLWithString:@"http://www.yourdeveloper.net/"]];
[self.composeVC setCompletionHandler:completionHandler];
[self.navigationController presentViewController:
self.composeVC animated:YES completion:nil];
}
```

使用SLComposeViewController

```
else {
UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Error"
message:@"Can't connect to your Twitter account"
delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil];
[alertView show];
}
}
- (void )didReceiveMemoryWarning
{
[super didReceiveMemoryWarning];
// Dispose of any resources that can be recreated.
}
@end
```

使用SLComposeViewController

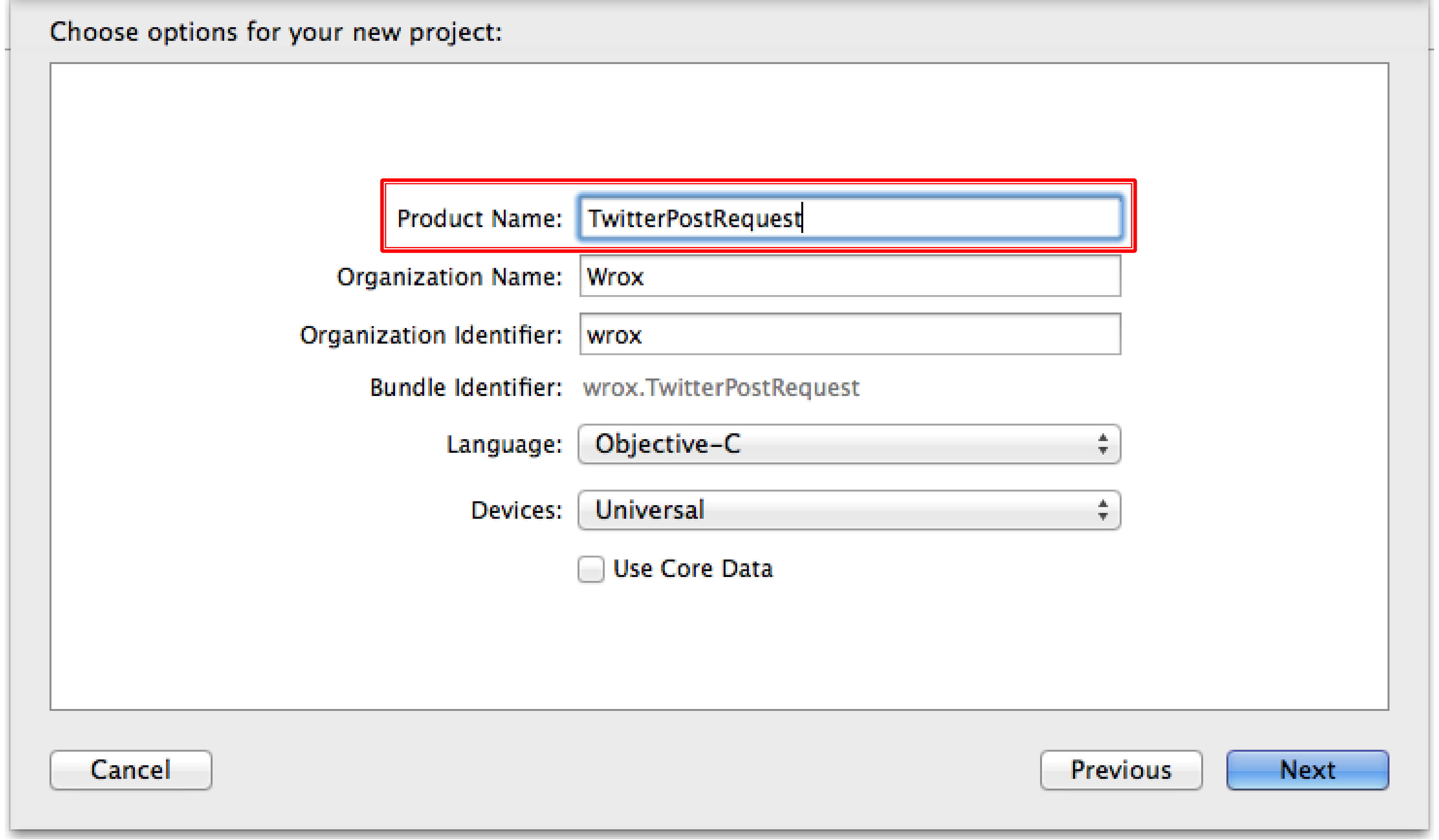


This is "My Accounts" application Test message.

模拟器上显示My Accounts应用消息

使用SLRequest

- 开启Xcode并使用Single View Application模板创建新项目
- 创建一个UINavigationController类型的强属性
- 实现UINavigationController来呈现YDViewController



Choose options for your new project:

Product Name: TwitterPostRequest

Organization Name: Wrox

Organization Identifier: wrox

Bundle Identifier: wrox.TwitterPostRequest

Language: Objective-C

Devices: Universal

□ Use Core Data

使用SLRequest

```
#import "YDViewController.h"
@interface YDViewController ()
@end
@implementation YDViewController
- (void)viewDidLoad
{
[super viewDidLoad];
// Do any additional setup after loading the view, typically from a nib.
[self requestAccessToAccountStore];
UIBarButtonItem *addEventButtonItem =
[[UIBarButtonItem alloc] initWithBarButtonSystemItem:
UIBarButtonSystemItemAdd target:self action:@selector(newTweet:)];
self.navigationItem.rightBarButtonItem = addEventButtonItem;
}
```

使用SLRequest

```
- (void)newTweet:(id)sender
{
//check if there are twitter accounts
if ([self.twitterAccounts count] >0)
{
SLRequestHandler requestHandler =
^(NSData *responseData, NSHTTPURLResponse
    *urlResponse, NSError *error) {
if (responseData) {
NSInteger statusCode = urlResponse.statusCode;
if (statusCode >= 200 && statusCode < 300) {
```

```
NSDictionary *postResponseData =
[NSJSONSerialization
    JSONObjectWithData:responseData
options:NSJSONReadingMutableContainers
error:NULL];
NSLog(@"[SUCCESS!] Created Tweet with ID:
    %@",
postResponseData[@"id_str"]);
}
```

使用SLRequest

```
else {
NSLog(@"[ERROR] Server responded: status code %d %@",
statusCode,
[NSHTTPURLResponse
localizedStringForStatusCode:statusCode]);
}
}
else {
NSLog(@"[ERROR] An error occurred while posting: %@", [error
localizedDescription]);
}
};
```

使用SLRequest

```
ACAccount* currentAccount = [self.twitterAccounts objectAtIndex:0];
NSDictionary* message =
@{@"status": @"My first post using the SLRequest class"};
NSURL* requestURL = [NSURL
URLWithString:@"http://api.twitter.com/1/statuses/update.json"];
SLRequest* postRequest = [SLRequest
requestForServiceType:SLServiceTypeTwitter
requestMethod:SLRequestMethodPOST
URL:requestURL parameters:message];
postRequest.account = currentAccount;
[postRequest performRequestWithHandler:requestHandler];
}
```

使用SLRequest

```
else
{
UIAlertView* alert = [[UIAlertView alloc]
    initWithTitle:@"Error"
message:@"Configure a Twitter account"
    delegate:self
cancelButtonTitle:@"OK" otherButtonTitles:nil, nil];
[alert show];
}
}
```

```
-(void)requestAccessToAccountStore
{
self.accountStore = [[ACAccountStore alloc] init];
self.twitterAccountType =
[self.accountStore
    accountTypeWithAccountTypeIdentifier:
ACAccountTypeIdentifierTwitter];
NSDictionary* options = nil;
[self.accountStore
    requestAccessToAccountsWithType:
self.twitterAccountType options:options
completion:^(BOOL granted, NSError *error)
```

使用SLRequest

```
if (granted == YES)
{
// do what you need to do here
dispatch_async(dispatch_get_main_queue(),^{
//call loadAccounts here now you now the user has granted access
[self loadAccounts];
});
}
else
{
dispatch_async(dispatch_get_main_queue(),^{
[self showAuthorizationError];
});
}
}];
}
```

使用SLRequest

```
-(void)loadAccounts
{
//check if we have been granted access to the twitter accounts
if ([self.twitterAccountType accessGranted])
{
self.twitterAccounts = [[NSArray alloc]
initWithArray:[self.accountStore
accountsWithAccountType:self.twitterAccountType]];
}
else
{
[self showAuthorizationError];
}
}
```

```
-(void)showAuthorizationError
{
UIAlertView* alert = [[UIAlertView alloc] initWithTitle:@"Error"
message:@"You are not authorized" delegate:self
cancelButtonTitle:@"OK" otherButtonTitles:nil, nil];
[alert show];
}
- (void)didReceiveMemoryWarning
{
[super didReceiveMemoryWarning];
// Dispose of any resources that can be recreated.
}
@end
```

Accounts框架的哪个类能够帮助创建、更改、删除和更新帐户凭证？

a) **ACAccountCredential**

b) **ACAccountStore**

c) **ACAccountType**

d) **ACAccount**

Accounts框架的哪个类能够帮助创建、更改、删除和更新帐户凭证？

a) ACAccountCredential

[✓ b) ACAccountStore]

c) ACAccountType

d) ACAccount

总结

Accounts框架提供Accounts数据库内帐户数据的轻松访问。

Social框架的SLComposeViewController类提供了编写帖子的用户界面。

Accounts框架包含四个类：

- ACAccount
- ACAccountCredential
- ACAccountStore
- ACAccountType

Social框架的SLRequest类提供了一个模板，你可以用它来进行发帖相关的HTTP请求。

你需要在viewDidLoad方法中调用requestAccessToAccountStore，来提示用户为你访问Accounts数据库授权。

WILEY

All brand names, product names, and logos represented in this course are trademarks, registered trademarks, or trade names of their respective holders. John Wiley & Sons Inc. is not associated with any product or vendor mentioned in this course.