|  |
| --- |
| ImageViewController  #import <UIKit/UIKit.h>  @interface ImageViewController : UIViewController  @property (nonatomic, strong) NSURL \*imageURL;  @end  //  // ImageViewController.m  // PictureFly  //  // Created by Jennifer Clark on 12/13/12.  // Copyright (c) 2012 Jennifer Clark. All rights reserved.  //  #import "ImageViewController.h"  @interface ImageViewController ()  @property (weak, nonatomic) IBOutlet UIImageView \*imageView;  @property (weak, nonatomic) IBOutlet UIActivityIndicatorView \*spinner;  @end  @implementation ImageViewController  @synthesize imageView = \_imageView, spinner = \_spinner, imageURL = \_imageURL;  - (id)initWithNibName:(NSString \*)nibNameOrNil bundle:(NSBundle \*)nibBundleOrNil  {  self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];  if (self) {  // Custom initialization  }  return self;  }  - (void)loadImage  {  [self.spinner startAnimating];  dispatch\_queue\_t downloadQueue = dispatch\_queue\_create("flickr photo download", NULL);  dispatch\_async(downloadQueue, ^{  NSData \*imageData = [NSData dataWithContentsOfURL:self.imageURL];  UIImage \*image = [UIImage imageWithData:imageData];  dispatch\_async(dispatch\_get\_main\_queue(), ^{  [self.spinner hidesWhenStopped];  [self.spinner stopAnimating];  self.imageView.image = image;  });  });  //dispatch\_release(downloadQueue);  }  - (void)setImageURL:(NSURL \*)imageURL  {  \_imageURL = imageURL;  if (self.imageView.window) [self loadImage];  }  - (void)viewWillAppear:(BOOL)animated  {  [super viewWillAppear:animated];  [self loadImage];  }  - (void)viewDidLoad  {  [super viewDidLoad];  // Do any additional setup after loading the view.  }  - (void)didReceiveMemoryWarning  {  [super didReceiveMemoryWarning];  [self setImageView:nil];  [self setSpinner:nil];  }  @end |
| TableViewController  #import <UIKit/UIKit.h>  @interface FlickrPhotoTableViewController : UITableViewController  @end  //  // FlickrPhotoTableViewController.m  // PictureFly  //  // Created by Jennifer Clark on 12/12/12.  // Copyright (c) 2012 Jennifer Clark. All rights reserved.  //  #import "FlickrPhotoTableViewController.h"  #import "FlickrFetcher.h"  #import "ImageViewController.h"  @interface FlickrPhotoTableViewController ()  // keys: photographer NSString, values: NSArray of photo NSDictionary  @property (nonatomic, strong) NSArray \*photos; //array of Flickr photo dictionaries  //@property (nonatomic, strong) NSDictionary \*photosByPhotographer;  @end  @implementation FlickrPhotoTableViewController  @synthesize photos = \_photos;  //@synthesize photosByPhotographer = \_photosByPhotographer;  //- (void)updatePhotosByPhotographer  //{  // NSMutableDictionary \*photosByPhotographer = [NSMutableDictionary dictionary];  // for (NSDictionary \*photo in self.photos) {  // NSString \*photographer = [photo objectForKey:FLICKR\_PHOTO\_OWNER];  // NSMutableArray \*photos = [photosByPhotographer objectForKey:photographer];  // if (!photos) {  // photos = [NSMutableArray array];  // [photosByPhotographer setObject:photos forKey:photographer];  // }  // [photos addObject:photo];  // }  // //self.photosByPhotographer = photosByPhotographer;  //}  - (void)setPhotos:(NSArray \*)photos  {  if (\_photos != photos) { // if the model is changed  \_photos = photos; // then reset the model  //[self updatePhotosByPhotographer];  if (self.tableView.window) [self.tableView reloadData]; // and reload the table data only IF the table view is onscreen (prevents issue from happening if reload is pressed and then a row is tapped to segue away    }  }  #pragma mark - Target/Action  - (IBAction)refresh:(id)sender  {  // might want to use introspection to be sure sender is UIBarButtonItem  // (if not, it can skip the spinner)  // that way this method can be a little more generic  UIActivityIndicatorView \*spinner = [[UIActivityIndicatorView alloc]initWithActivityIndicatorStyle:UIActivityIndicatorViewStyleGray];  [spinner startAnimating];  self.navigationItem.rightBarButtonItem = [[UIBarButtonItem alloc]initWithCustomView:spinner];    dispatch\_queue\_t downloadQueue = dispatch\_queue\_create("flickr downloader", NULL); //create the queue - NULL defaults to the serial queue type  dispatch\_async(downloadQueue, ^ { //async takes two arguments, the name of the queue and the block  NSArray \*photos = [FlickrFetcher recentGeoreferencedPhotos];  dispatch\_async(dispatch\_get\_main\_queue(), ^ {  self.photos = photos; //this needs to go on the main queue becuase it calls set photos, which reloads the table view, which is UIKit    self.navigationItem.rightBarButtonItem = sender;  });  });  //dispatch\_release(downloadQueue); //to prevent memory leak  }  #pragma mark - Segue  - (void) prepareForSegue:(UIStoryboardSegue \*)segue sender:(id)sender  {  if ([segue.destinationViewController isKindOfClass:[ImageViewController class]]) {    ImageViewController \*ivc = (ImageViewController \*)segue.destinationViewController;  NSIndexPath \*indexPath = [self.tableView indexPathForCell:sender];  NSDictionary \*photo = [self.photos objectAtIndex:indexPath.row];  ivc.imageURL = [FlickrFetcher urlForPhoto:photo format:FlickrPhotoFormatLarge];  ivc.title = [photo objectForKey:FLICKR\_PHOTO\_TITLE];  }    }    #pragma mark - Table view data source  //- (NSString \*)photographerForSection:(NSInteger)section  //{  // return [[self.photosByPhotographer allKeys] objectAtIndex:section];  //}  //  //- (NSString \*)tableView:(UITableView \*)tableView titleForHeaderInSection:(NSInteger)section  //{  // return [self photographerForSection:section];  //}  //  //- (NSInteger)numberOfSectionsInTableView:(UITableView \*)tableView  //{  // return [self.photosByPhotographer count];  //}  - (NSInteger)tableView:(UITableView \*)tableView numberOfRowsInSection:(NSInteger)section  {  return [self.photos count];  //NSString \*photographer = [self photographerForSection:section];  //NSArray \*photosByPhotographer = [self.photosByPhotographer objectForKey:photographer];  //return [photosByPhotographer count];  }  - (UITableViewCell \*)tableView:(UITableView \*)tableView cellForRowAtIndexPath:(NSIndexPath \*)indexPath  {  static NSString \*CellIdentifier = @"Flicker Photo";    UITableViewCell \*cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier forIndexPath:indexPath];  if (cell == nil) {  cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier];  }    // Configure the cell...  NSDictionary \*photo = [self.photos objectAtIndex:indexPath.row];  // NSString \*photographer = [self photographerForSection:indexPath.section];  // NSArray \*photosByPhotographer = [self.photosByPhotographer objectForKey:photographer];  // NSDictionary \*photo = [photosByPhotographer objectAtIndex:indexPath.row];  cell.textLabel.text = [photo objectForKey:FLICKR\_PHOTO\_TITLE];  cell.detailTextLabel.text = [photo objectForKey:FLICKR\_PHOTO\_OWNER];    return cell;    }  @end |