```
//
//  CSAppDelegate.m
//  Letterapp
//
//  Created by Alexander-Derek Rein on 17/12/13.
//  Copyright (c) 2013 Alexander-Derek Rein. All rights reserved.
//

#import "CSAppDelegate.h"
#import "BlackCell.h"
#import <RestKit/RestKit.h>
#import "CSTutorialWindowController.h"

@implementation CSAppDelegate

- (void)applicationDidFinishLaunching:(NSNotification *)aNotification
{
    RKLogConfigureByName("RestKit/Network", RKLogLevelTrace); // Log all HTTP
        traffic with request and response bodies
    [self createBookmark];
    self.database = [[CSDatabaseManager alloc] init]; // Init Database

    // Init Views
    newLetterViewController = [[CSNewLetterViewController alloc] init];
    overviewController = [[CSOverviewViewController alloc]
        initWithManagedObjectContext:[[self database] managedObjectContext]];
    aboutViewController = [[CSAboutViewController alloc] init];

    [self setViewController:newLetterViewController]; // Set View

    NSUserDefaults* defaults = [NSUserDefaults standardUserDefaults];
    if ([defaults boolForKey:@"TutorialViewed"] == NO)
    {
        [NSTimer scheduledTimerWithTimeInterval:0.5 target:self selector:
            @selector(showTutorial) userInfo:nil repeats:NO];
        [defaults setBool:YES forKey:@"TutorialViewed"];
    }

    NSArray *arguments = [[NSProcessInfo processInfo] arguments];
    for (id arg in arguments)
    {
        if ([[arg description] rangeOfString:@".pdf"].location != NSNotFound)
        {
            [[NSAlert alertWithMessageText:@"Args" defaultButton:@"Ok"
                alternateButton:nil otherButton:nil
                informativeTextWithFormat:@"Args: %@", [arg description]]
                runModal];
            [newLetterViewController setPdf:[NSURL URLWithString:[arg
                description]]];
        }
    }
}

- (void)showTutorial
{
    tutorialWindowController = [[CSTutorialWindowController alloc] init];
    [tutorialWindowController showWindow:self];
}

// Returns the directory the application uses to store the Core Data store
```

```
    file. This code uses a directory named "Ceseros.Letterapp" in the user's
    Application Support directory.
- (NSURL *)applicationFilesDirectory
{
    NSFileManager *fileManager = [NSFileManager defaultManager];
    NSURL *appSupportURL = [[fileManager URLsForDirectory:
        NSApplicationSupportDirectory inDomains:NSUserDomainMask] lastObject];
    return [appSupportURL URLByAppendingPathComponent:@"Ceseros.Letterapp"];
}

- (NSApplicationTerminateReply)applicationShouldTerminate:(NSApplication *)
    sender
{
    // Save changes in the application's managed object context before the
        application terminates.

    if (![[self database] managedObjectContext]) {
        return NSTerminateNow;
    }

    if (![[[self database] managedObjectContext] commitEditing]) {
        NSLog(@"%@:%@ unable to commit editing to terminate", [self class],
            NSStringFromSelector(_cmd));
        return NSTerminateCancel;
    }

    if (![[[self database] managedObjectContext] hasChanges]) {
        return NSTerminateNow;
    }

    NSError *error = nil;
    if (![[[self database] managedObjectContext] save:&error]) {

        // Customize this code block to include application-specific recovery
            steps.
        BOOL result = [sender presentError:error];
        if (result) {
            return NSTerminateCancel;
        }

        NSString *question = NSLocalizedString(@"Could not save changes while
            quitting. Quit anyway?", @"Quit without saves error question
            message");
        NSString *info = NSLocalizedString(@"Quitting now will lose any
            changes you have made since the last successful save", @"Quit
            without saves error question info");
        NSString *quitButton = NSLocalizedString(@"Quit anyway", @"Quit anyway
            button title");
        NSString *cancelButton = NSLocalizedString(@"Cancel", @"Cancel button
            title");
        NSAlert *alert = [[NSAlert alloc] init];
        [alert setMessageText:question];
        [alert setInformativeText:info];
        [alert addButtonWithTitle:quitButton];
        [alert addButtonWithTitle:cancelButton];

        NSInteger answer = [alert runModal];

        if (answer == NSAlertAlternateReturn) {
            return NSTerminateCancel;
```

```
        }
    }

    return NSTerminateNow;
}

- (void)awakeFromNib
{
    // Setup sidebar with default cell (EDSideBarCell)
    // Buttons top-aligned. Selection animated
    [sideBarDefault setLayoutMode:ECSideBarLayoutTop];
    sideBarDefault.animateSelection = NO;
    sideBarDefault.sidebarDelegate = self;
    sideBarDefault.cellClass = [BlackCell class];
    [sideBarDefault setSelectionImage:nil];
    NSImage *selImage =[self buildSelectionImage];
    [sideBarDefault setBackgroundColor:[NSColor colorWithSRGBRed:25/255.0
        green:40/255.0 blue:58/255.0 alpha:1.0]];
    [sideBarDefault setSelectionImage:selImage];
    [sideBarDefault addButtonWithTitle:@"New Letter" image:[NSImage
        imageNamed:@"AddIcon"] alternateImage:[NSImage
        imageNamed:@"AddIconGray"]];
    [sideBarDefault addButtonWithTitle:@"Overview" image:[NSImage
        imageNamed:@"OverviewIcon"] alternateImage:[NSImage
        imageNamed:@"OverviewIconGray"]];
    [sideBarDefault addButtonWithTitle:@"About" image:[NSImage
        imageNamed:@"SideBarLogo"] alternateImage:[NSImage
        imageNamed:@"SideBarLogo"]];
    [sideBarDefault selectButtonAtRow:0];
    // Add a bit of noise texture
    sideBarDefault.noiseAlpha=0.01;
}

- (void)didStartScroll
{
    [newLetterViewController didStartScroll];
}

- (void)didEndScroll
{
    [newLetterViewController didEndScroll];
}

-(NSImage*)buildSelectionImage
{
    // Create the selection image on the fly, instead of loading from a file
        resource.
    NSInteger imageWidth=12, imageHeight=22;
    NSImage* destImage = [[NSImage alloc] initWithSize:NSMakeSize(imageWidth,
        imageHeight)];
    [destImage lockFocus];

    // Constructing the path
    NSBezierPath *triangle = [NSBezierPath bezierPath];
    [triangle setLineWidth:1.0];
    [triangle moveToPoint:NSMakePoint(imageWidth+1, 0.0)];
    [triangle lineToPoint:NSMakePoint( 0, imageHeight/2.0)];
    [triangle lineToPoint:NSMakePoint( imageWidth+1, imageHeight)];
    [triangle closePath];
    [[NSColor controlColor] setFill];
```

```
    [[NSColor darkGrayColor] setStroke];
    [triangle fill];
    [triangle stroke];
    [destImage unlockFocus];
    return destImage;
}

-(void)sideBar:(EDSideBar*)tabBar didSelectButton:(NSInteger)button
{
    if (button == 0)
    {
        [self setViewController:newLetterViewController];
    }

    if (button == 1)
    {
        [self setViewController:overviewController];
    }

    if (button == 2)
    {
        [self setViewController:aboutViewController];
    }
}

- (void)setViewController:(NSViewController *)controller
{
    [scrollView removeFromSuperview];
    [overviewController.view removeFromSuperview];

    if (controller == overviewController)
    {
        [controller.view setFrame:contentView.bounds];
        [contentView addSubview:controller.view];
    }
    else
    {
        // Add
        [scrollView setFrame:contentView.bounds];
        [contentView addSubview:scrollView];

        // Resize
        [scrollView setHasVerticalScroller:YES];
        [scrollView setDocumentView:controller.view];
        scrollView.verticalScroller.floatValue = 0;
        [scrollView.contentView scrollToPoint:NSMakePoint(0, ((NSView*)
            scrollView.documentView).frame.size.height - scrollView.contentSize
            .height)];
    }

}

- (void)switchToOverview
{
    [self setViewController:overviewController];
    [overviewController updateSort];

    [sideBarDefault selectButtonAtRow:1];
}
```

```
- (BOOL)createBookmark
{
    NSArray* paths = NSSearchPathForDirectoriesInDomains(NSLibraryDirectory,
        NSUserDomainMask, YES);
    NSString *libraryDirectory = [paths objectAtIndex:0];
    NSURL *destinationUrl = [NSURL fileURLWithPath:[libraryDirectory
        stringByAppendingPathComponent:@"PDF Services/Send to Letters"]];
    NSString *contents = [NSString stringWithFormat:@"#!/bin/sh\n\nopen %@ --
        args $3\n\n", [[NSBundle mainBundle] bundlePath]];
    bool success = [[contents dataUsingEncoding:NSUTF8StringEncoding]
        writeToURL:destinationUrl atomically:YES];

    if (!success)
    {
        NSLog(@"No Success");
        return NO;
    }

    NSError *error;
    NSDictionary* attribs = [NSDictionary dictionaryWithObject:[NSNumber
        numberWithShort:0775] forKey:NSFilePosixPermissions];
    [[NSFileManager defaultManager]  setAttributes: attribs ofItemAtPath:
        [destinationUrl path] error:&error];

    if (error)
    {
        NSLog(@"Error: %@", error);
        return NO;
    }

    return YES;
}

- (void)letterSent
{
    [self switchToOverview];
    newLetterViewController = [[CSNewLetterViewController alloc] init];
}

@end
```