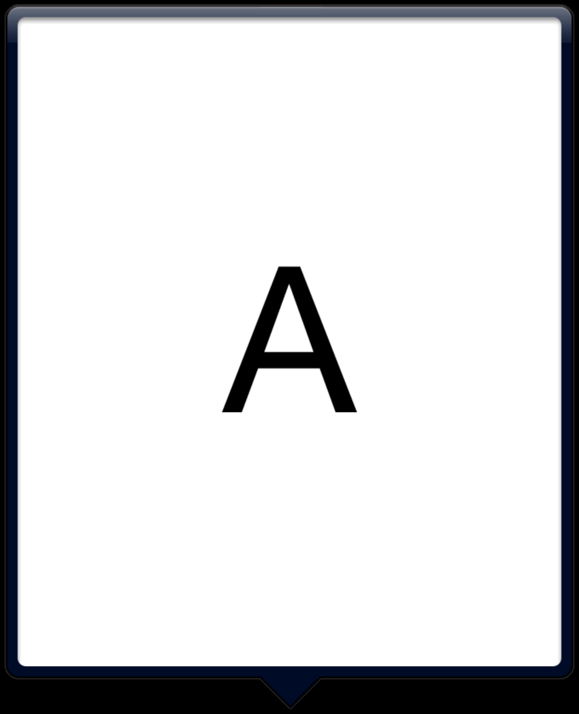
What’s New in Cocoa Touch

Session 200 / 201 / 207

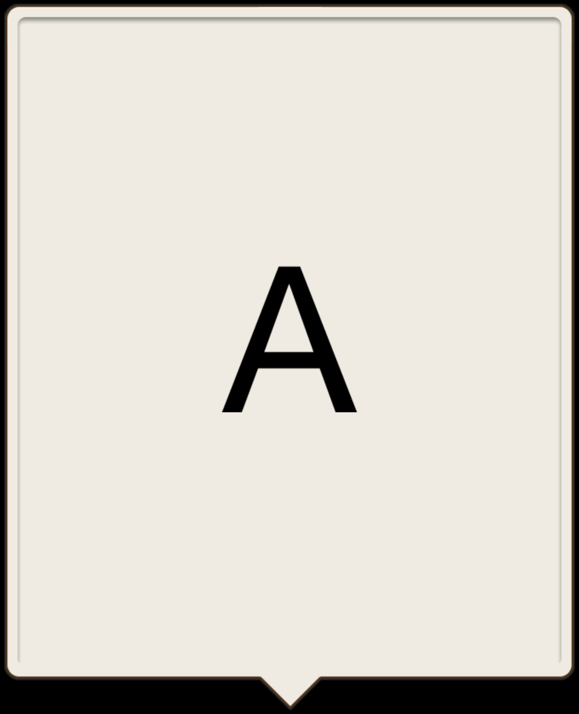
Chris Parker
UIKit Engineer

Appearance

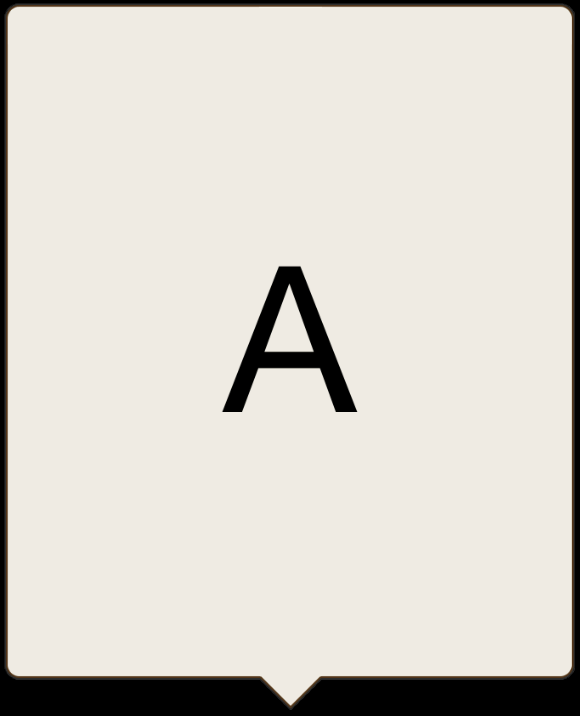
UIPopoverBackgroundView



UIPopoverBackgroundView

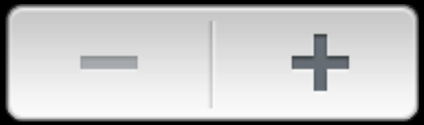


UIPopoverBackgroundView



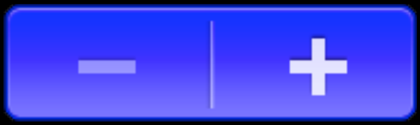
```
+ (BOOL)wantsDefaultContentAppearance;
```

UIStepper



UIStepper

Tint color



```
@property (nonatomic,retain) UIColor *tintColor;
```

UIStepper

Image customization

```
- (void)setBackgroundImage:(UIImage*)image
                  forState:(UIControlState)state;
```

UIStepper

Image customization

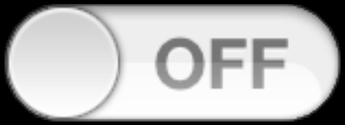
```
- (void)setBackgroundImage:(UIImage*)image
                  forState:(UIControlState)state;

- (void)setDividerImage:(UIImage*)image
    forLeftSegmentState:(UIControlState)left
      rightSegmentState:(UIControlState)right;
```

UIStepper

Image customization

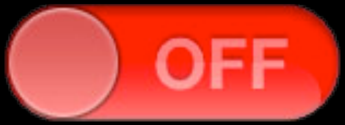
```
- (void)setBackgroundImage:(UIImage*)image
                  forState:(UIControlState)state;

- (void)setDividerImage:(UIImage*)image
    forLeftSegmentState:(UIControlState)left
      rightSegmentState:(UIControlState)right;

- (void)setIncrementImage:(UIImage *)image
                 forState:(UIControlState)state;
- (void)setDecrementImage:(UIImage *)image
                 forState:(UIControlState)state;
```

UISwitch



UISwitch

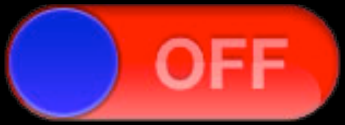
Tint color



```
@property (nonatomic, retain) UIColor *tintColor;
```

UISwitch

Thumb tint color



```
@property (nonatomic, retain) UIColor *thumbTintColor;
```

UISwitch

Image customization

```
@property (nonatomic, retain) UIImage *onImage;
@property (nonatomic, retain) UIImage *offImage;
```

UINavigationBar & UITabBar

Shadow images

```
@property (nonatomic,retain) UIImage *shadowImage;
```

UIToolbar

Shadow images

```
- (void)setShadowImage:(UIImage *)shadowImage
    forToolbarPosition:(UIToolbarPosition)topOrBottom;
```

UIBarButtonItem

Background images

```
- (void)setBackgroundImage:(UIImage *)bgImage
                  forState:(UIControlState)state
                     style:(UIBarButtonItemStyle)style
                barMetrics:(UIBarMetrics)barMetrics;
```

UIPageControl

Tint colors

```
@property (nonatomic,retain) UIColor *pageIndicatorTintColor;
@property (nonatomic,retain) UIColor *currentPageIndicatorTintColor;
```

UIImage API

Create images from raw data

```
+ (UIImage *)imageWithData:(NSData *)data
                     scale:(CGFloat)scale;

- (id)initWithData:(NSData *)data
             scale:(CGFloat)scale;
```

UIImage API

Create images from CIImages

```
+ (UIImage *)imageWithCIImage:(CIImage *)ciImage
            scale:(CGFloat)scale
      orientation:(UIImageOrientation)orientation;

- (id)initWithCIImage:(CIImage *)ciImage
            scale:(CGFloat)scale
      orientation:(UIImageOrientation)orientation;
```

UIPageViewController

Scrolling with view controllers

```
typedef NS_ENUM(NSInteger, UIPageViewControllerTransitionStyle) {
    UIPageViewControllerTransitionStylePageCurl = 0,
    UIPageViewControllerTransitionStyleScroll = 1
};

NSString *const UIPageViewControllerOptionInterPageSpacingKey;
```

UIPageViewController

UIPageViewControllerDelegate

```
- (void)pageViewController:(UIPageViewController *)pageViewController
willTransitionToViewControllers:(NSArray *)pendingViewControllers;

- (NSInteger)presentationCountForPageViewController:
                                                   (UIPageViewController *)pvc;
- (NSInteger)presentationIndexForPageViewController:
                                                   (UIPageViewController *)pvc;
```

Demo

UIPageViewController

UIPageViewControllerDelegate

```
- (void)pageViewController:(UIPageViewController *)pageViewController
willTransitionToViewControllers:(NSArray *)pendingViewControllers;

- (NSInteger)presentationCountForPageViewController:
                                                      (UIPageViewController *)pvc;
- (NSInteger)presentationIndexForPageViewController:
                                                      (UIPageViewController *)pvc;
```

Appearance Sessions

Advanced Appearance Customization on iOS	Mission Wednesday 10:15AM

UITableView

UITableViewCell Reuse

```
   - (void)registerNib:(UINib *)nib
forCellReuseIdentifier:(NSString *)identifier;

 - (void)registerClass:(Class)cellClass
forCellReuseIdentifier:(NSString *)identifier;

- (id)dequeueReusableCellWithIdentifier:(NSString *)identifier
                           forIndexPath:(NSIndexPath *)indexPath;
```

UITableViewCell Reuse

```
UITableViewCell *cell =
        [tableView dequeueReusableCellWithIdentifier:@”Cell”];

if (!cell) {
    cell = [[UITableViewCell alloc] init…];
}

[cell setTitle:[names objectAtIndex:[indexPath row]];
```

UITableViewCell Reuse

```
UITableViewCell *cell =
        [tableView dequeueReusableCellWithIdentifier:@”Cell”
                                        forIndexPath:indexPath];

[cell setTitle:[names objectAtIndex:[indexPath row]];
```

UITableViewHeaderFooterView

```
@property(nonatomic, retain) UIColor *tintColor;

@property(nonatomic, readonly, retain) UILabel* textLabel;
@property(nonatomic, readonly, retain) UILabel* detailTextLabel;

@property (nonatomic, readonly, retain) UIView *contentView;
@property (nonatomic, retain) UIView *backgroundView;

@property (nonatomic, readonly, copy) NSString *reuseIdentifier;

- (id)initWithReuseIdentifier:(NSString *)reuseIdentifier;
- (void)prepareForReuse;
```

UITableView

Sections

```
- (id)dequeueReusableHeaderFooterViewWithIdentifier:(NSString *)i;

- (UITableViewHeaderFooterView *)headerViewForSection:
(NSInteger)section;
- (UITableViewHeaderFooterView *)footerViewForSection:
(NSInteger)section;

@property(nonatomic, retain) UIColor *sectionIndexColor;
@property(nonatomic, retain) UIColor
*sectionIndexTrackingBackgroundColor;
```

UITableView

Delegate methods

```
    - (void)tableView:(UITableView *)tableView
      willDisplayCell:(UITableViewCell *)cell
    forRowAtIndexPath:(NSIndexPath *)indexPath;

    - (void)tableView:(UITableView *)tableView
willDisplayHeaderView:(UIView *)view
           forSection:(NSInteger)section;

    - (void)tableView:(UITableView *)tableView
willDisplayFooterView:(UIView *)view
           forSection:(NSInteger)section;
```

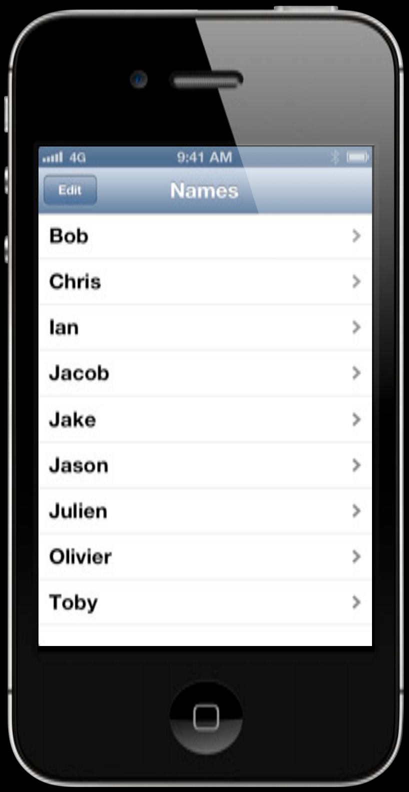
UITableView

Delegate methods

```
      - (void)tableView:(UITableView *)tableView
  didEndDisplayingCell:(UITableViewCell *)cell
     forRowAtIndexPath:(NSIndexPath *)indexPath;

      - (void)tableView:(UITableView *)tableView
didEndDisplayingHeaderView:(UIView *)view
            forSection:(NSInteger)section;

      - (void)tableView:(UITableView *)tableView
didEndDisplayingFooterView:(UIView *)view
            forSection:(NSInteger)section;
```

UIRefreshControl

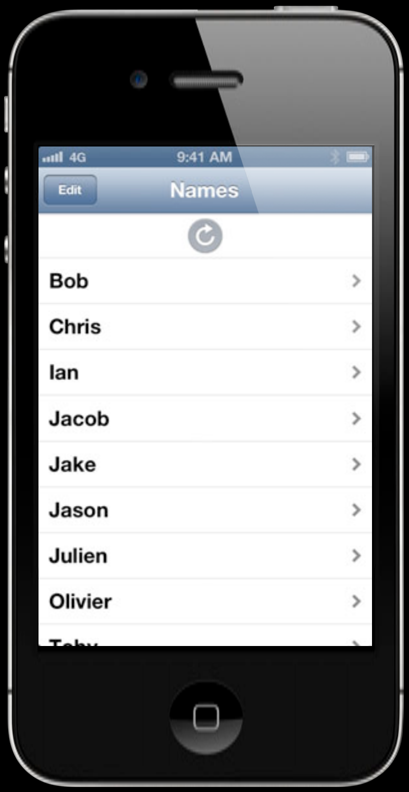
UITableViewController



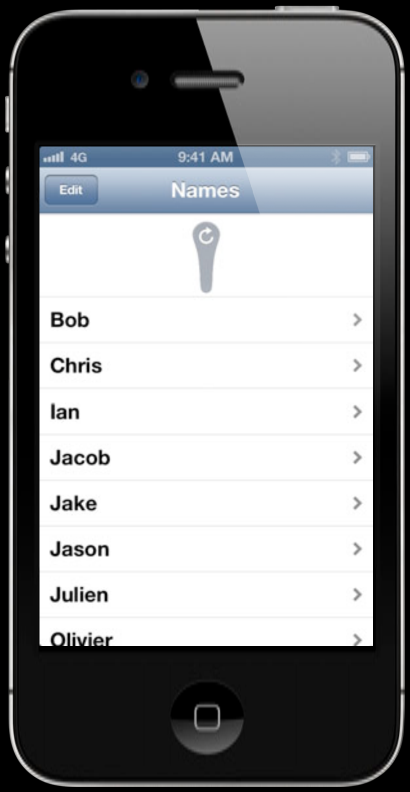
Bob
Chris
Ian
Jacob
Jake
Jason
Julien
Olivier
Toby

UIRefreshControl

UITableViewController

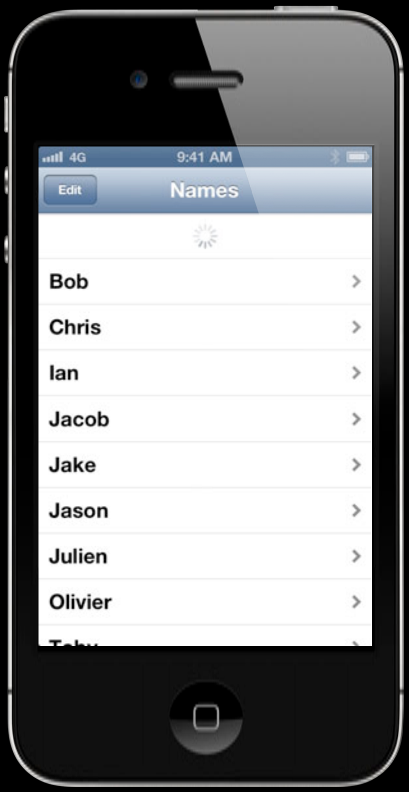


UIRefreshControl
UITableViewController



UIRefreshControl

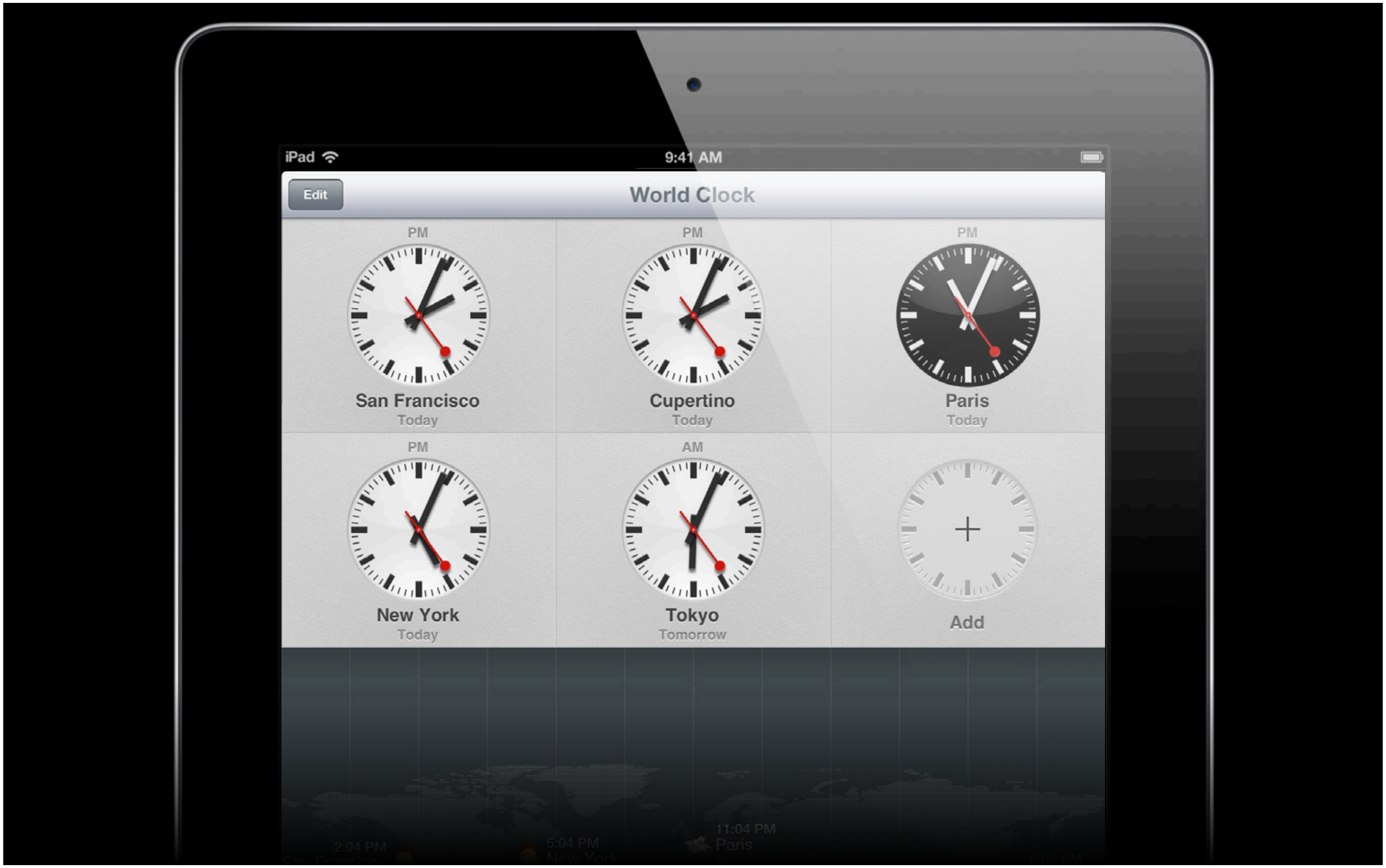
UITableViewController

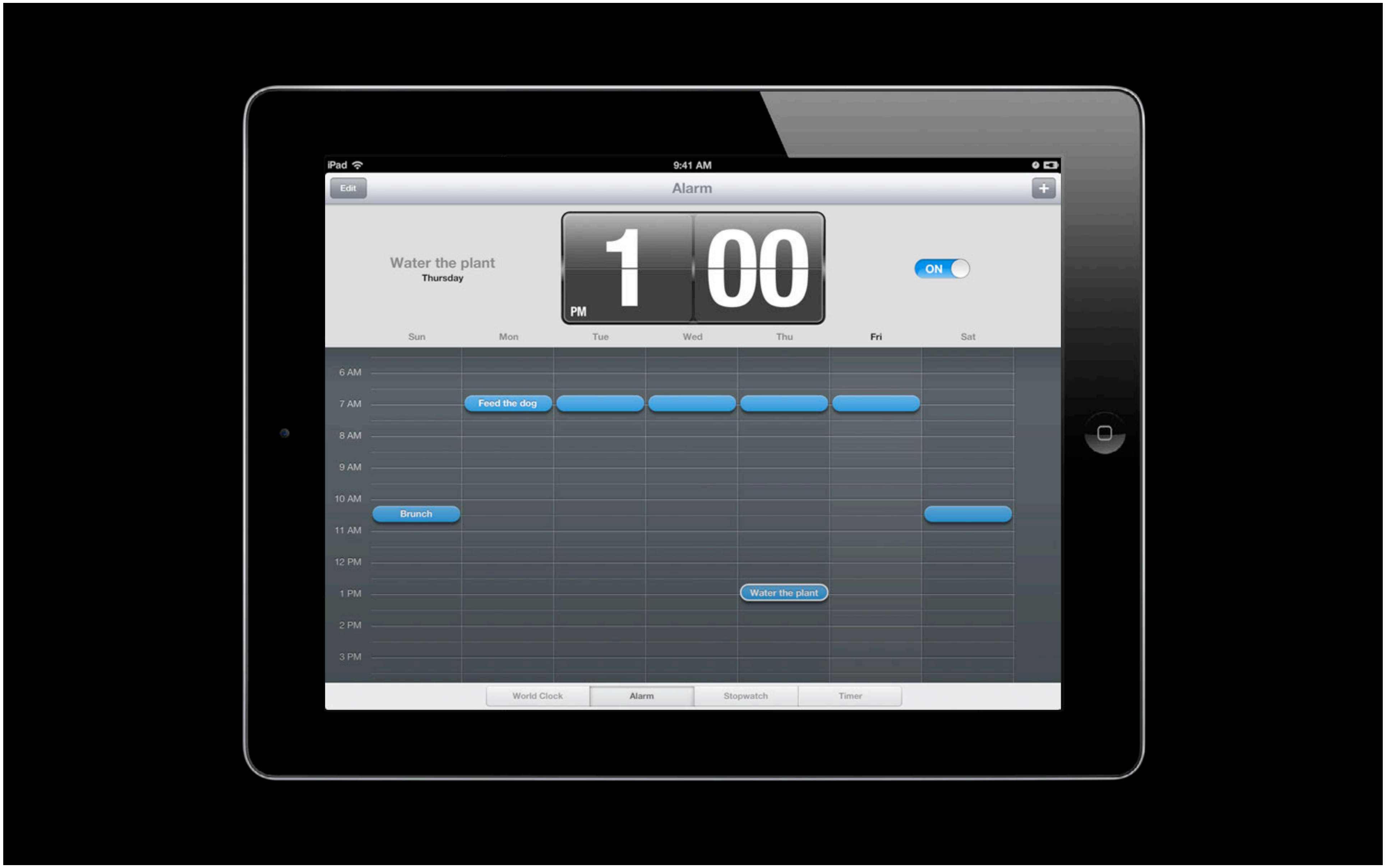


UIRefreshControl

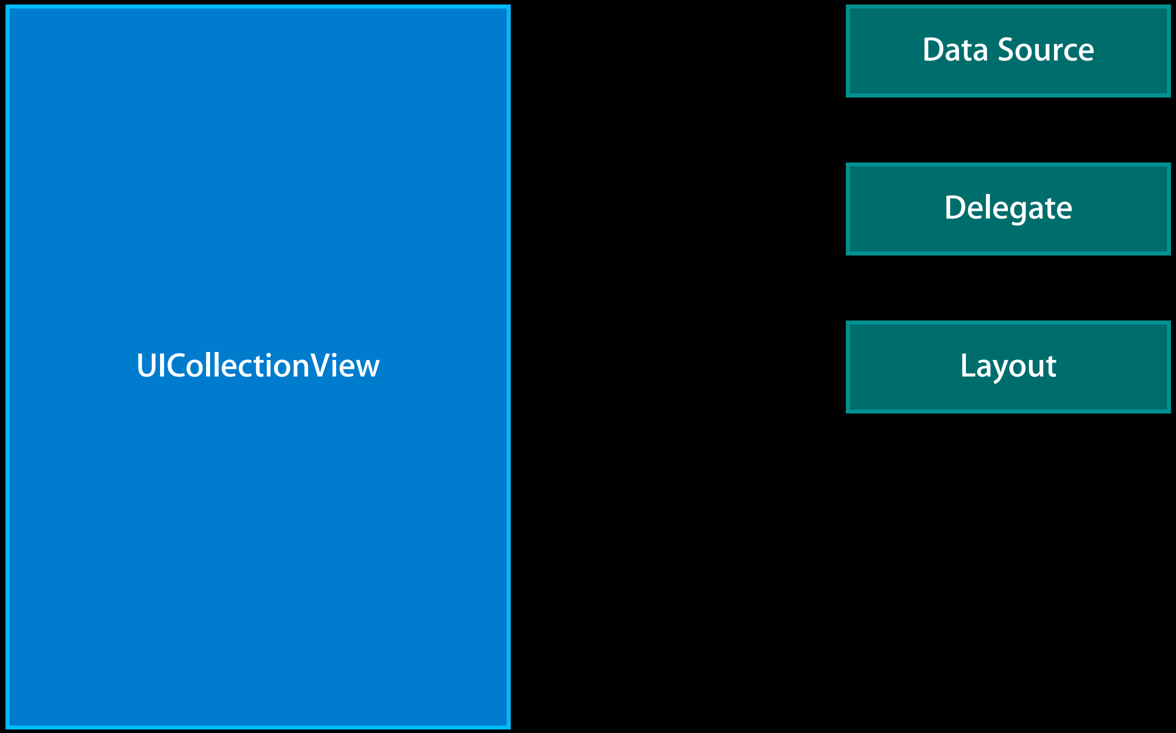
```
- (id)init;

@property (nonatomic, readonly, getter=isRefreshing) BOOL refreshing;

@property (nonatomic, retain) UIColor *tintColor;

- (void)beginRefreshing;
- (void)endRefreshing;
```

UICollectionView

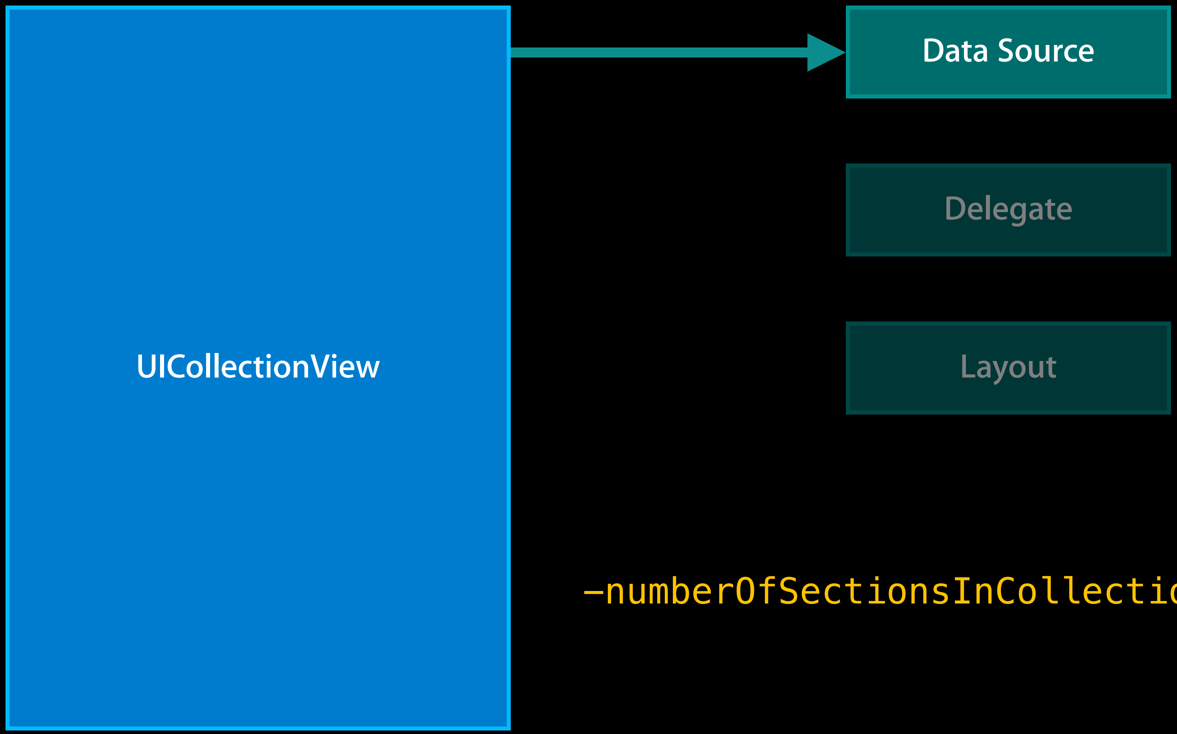

iPad
9:41 AM
Edit
World Clock
PM
San Francisco
Today
PM
Cupertino
Today
PM
Paris
Today
PM
New York
Today
AM
Tokyo
Tomorrow
Add
2:04 PM
5:04 PM
New York
11:04 PM
Paris


iPad
9:41 AM
Edit
Alarm
+
Water the plant
Thursday
1
PM
00
ON
Sun
Mon
Tue
Wed
Thu
Fri
Sat
6 AM
7 AM
8 AM
9 AM
10 AM
11 AM
12 PM
1 PM
2 PM
3 PM
Feed the dog
Brunch
Water the plant
World Clock
Alarm
Stopwatch
Timer

The Parts

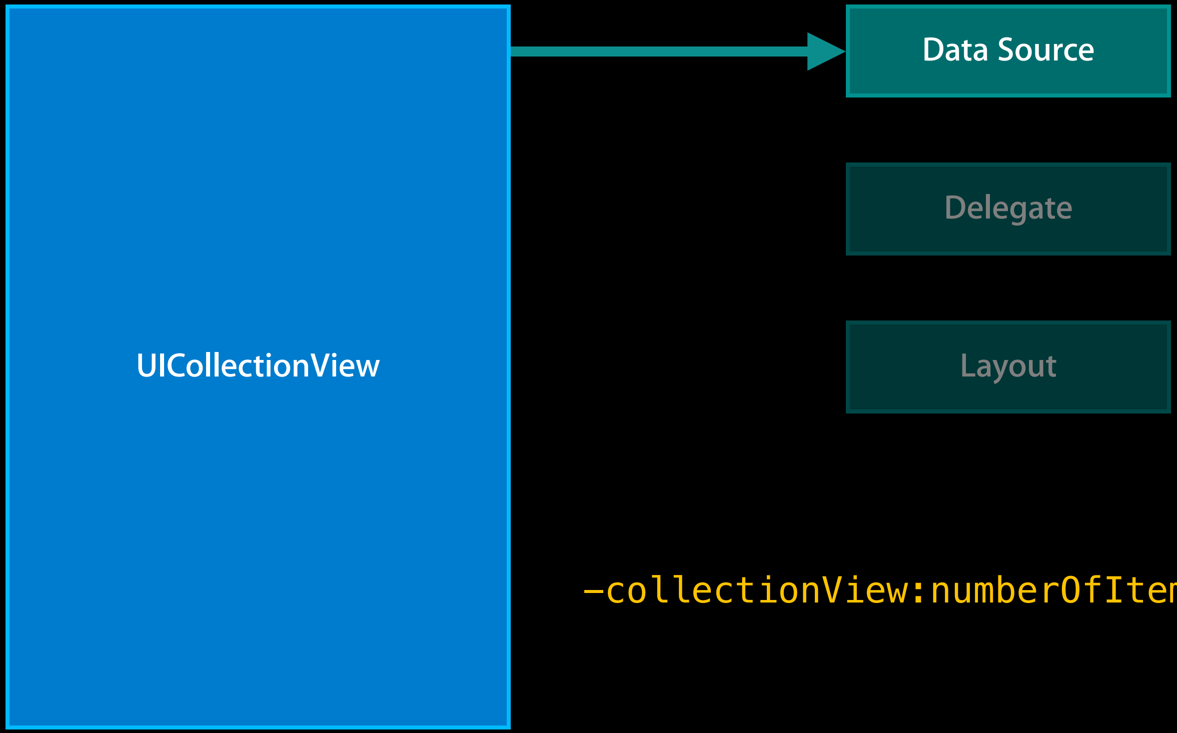


Laying Out the Collection



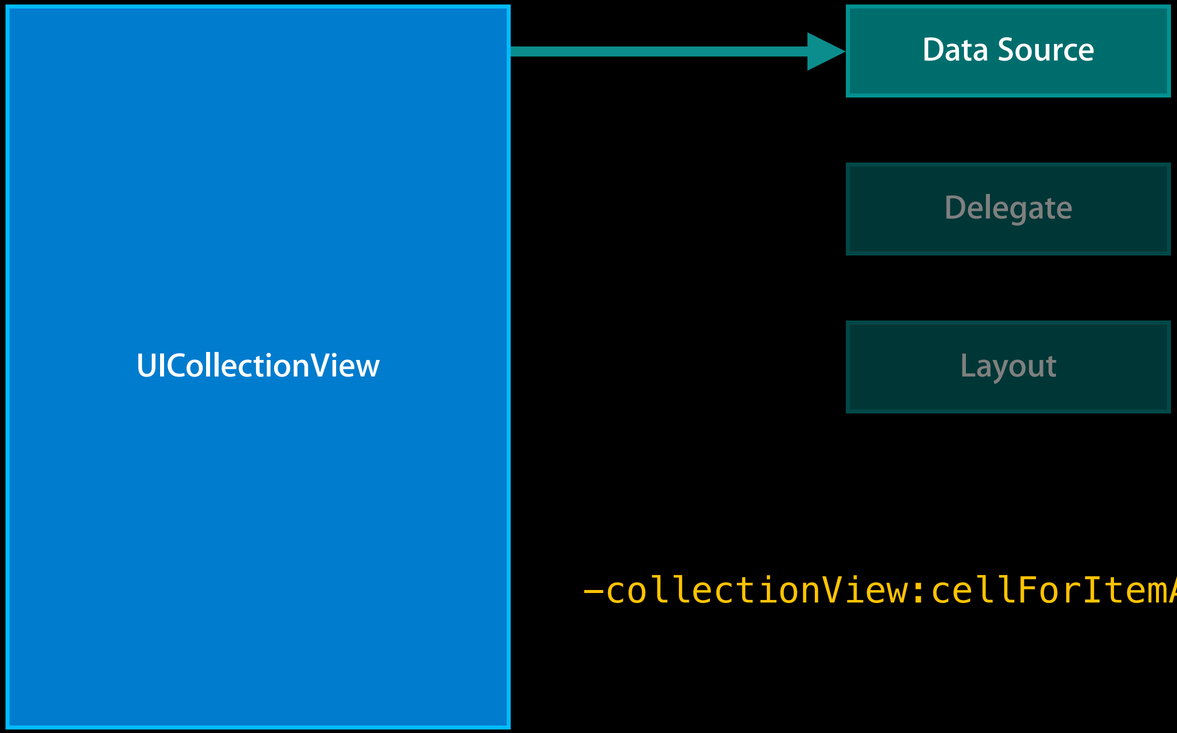
```
-numberOfSectionsInCollectionView:
```

Laying Out the Collection



```
-collectionView:numberOfItemsInSection:
```

Laying Out the Collection



```
-collectionView:cellForItemAtIndexPath:
```

UICollectionViewCell

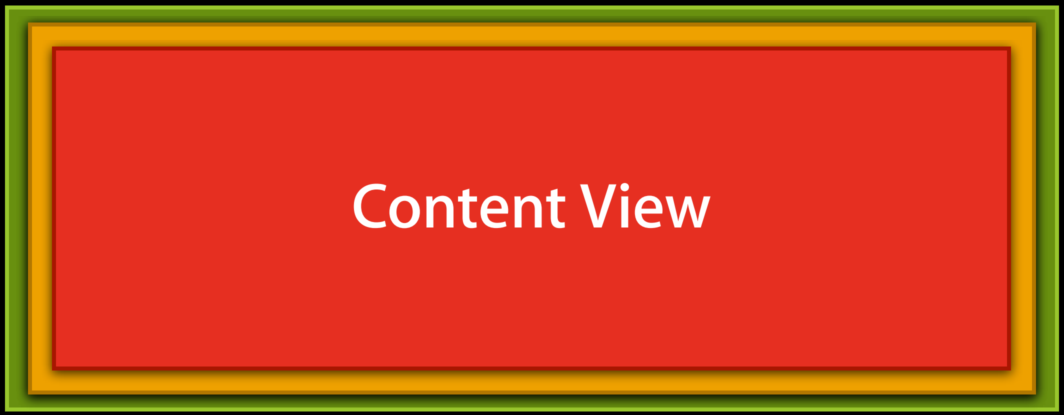
UICollectionViewCell

Background View

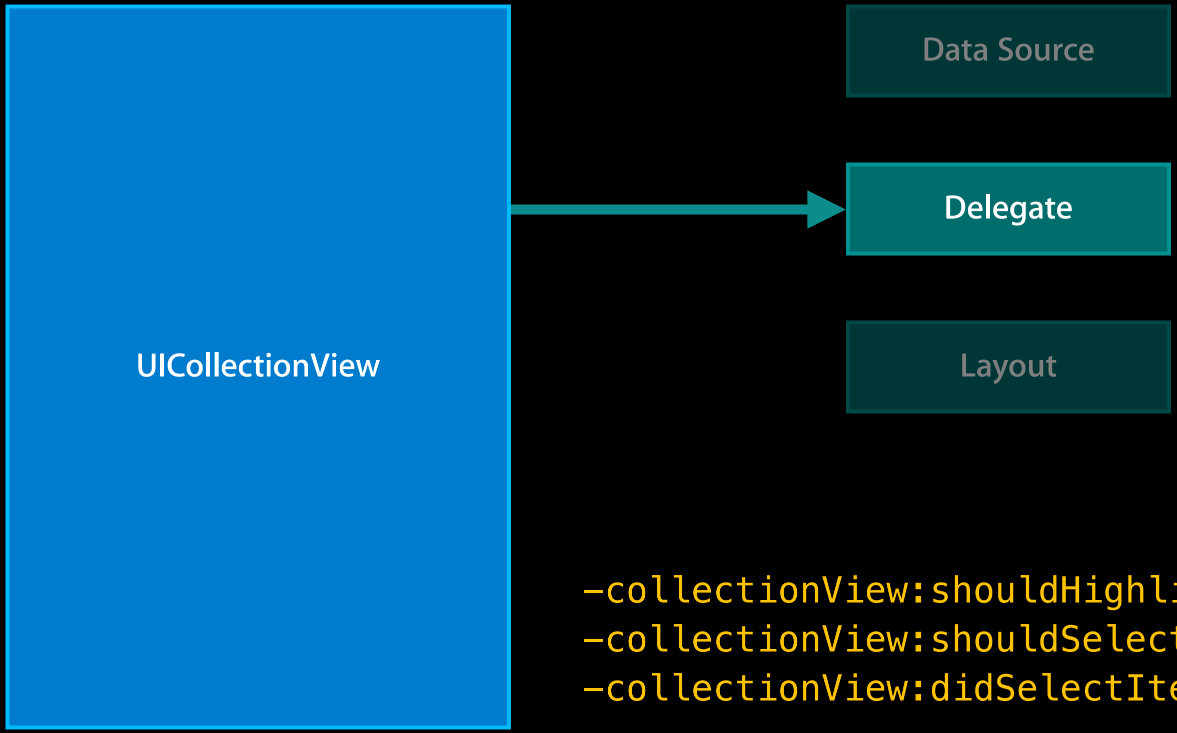
UICollectionViewCell

Selected Background View

UICollectionViewCell

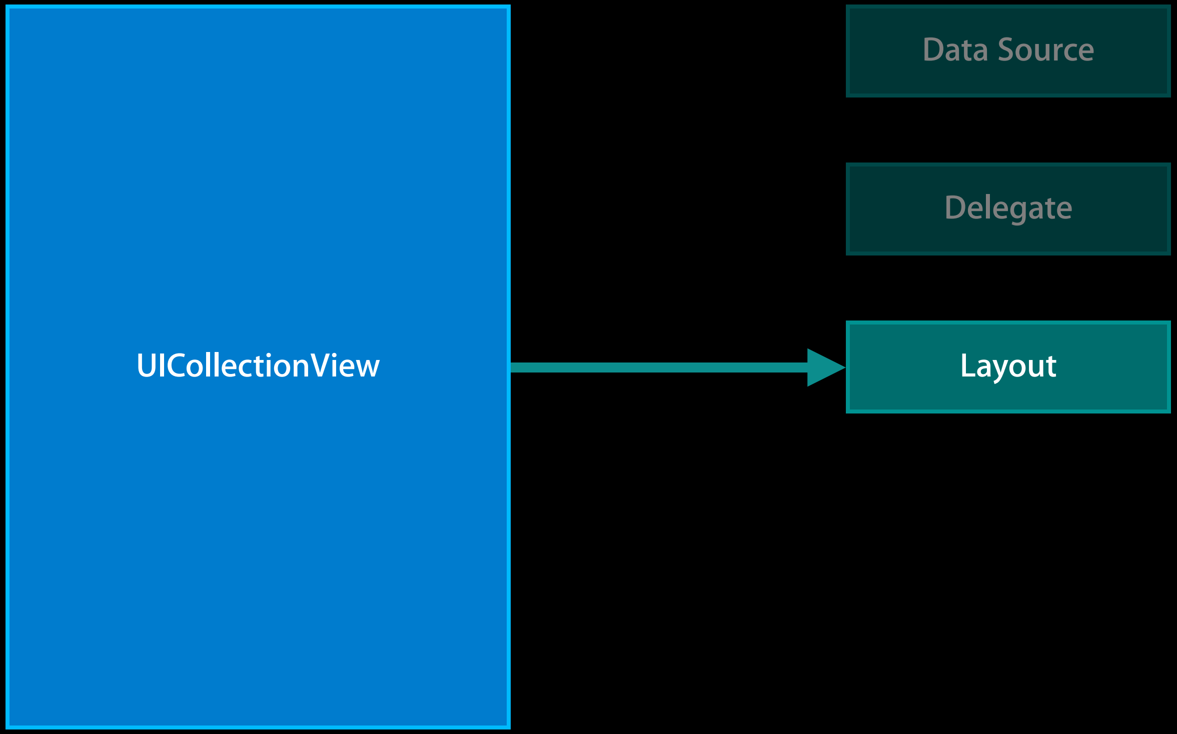


User Events

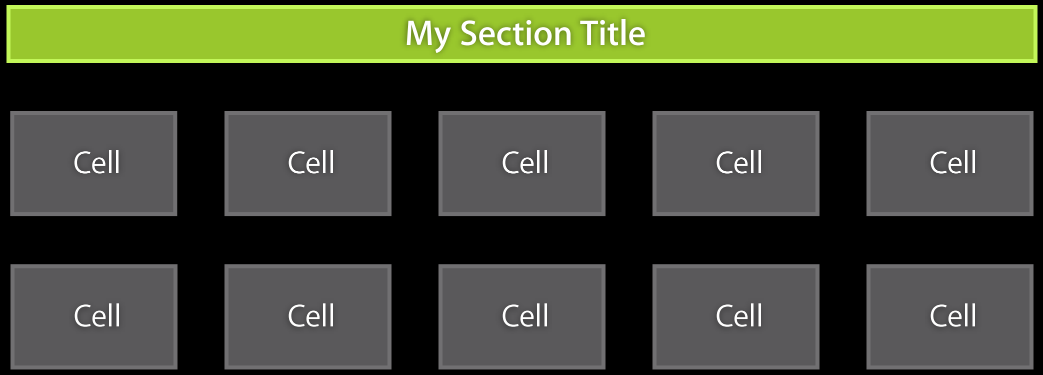


```
-collectionView:shouldHighlightItemAtIndexPath:
-collectionView:shouldSelectItemAtIndexPath:
-collectionView:didSelectItemAtIndexPath:
```

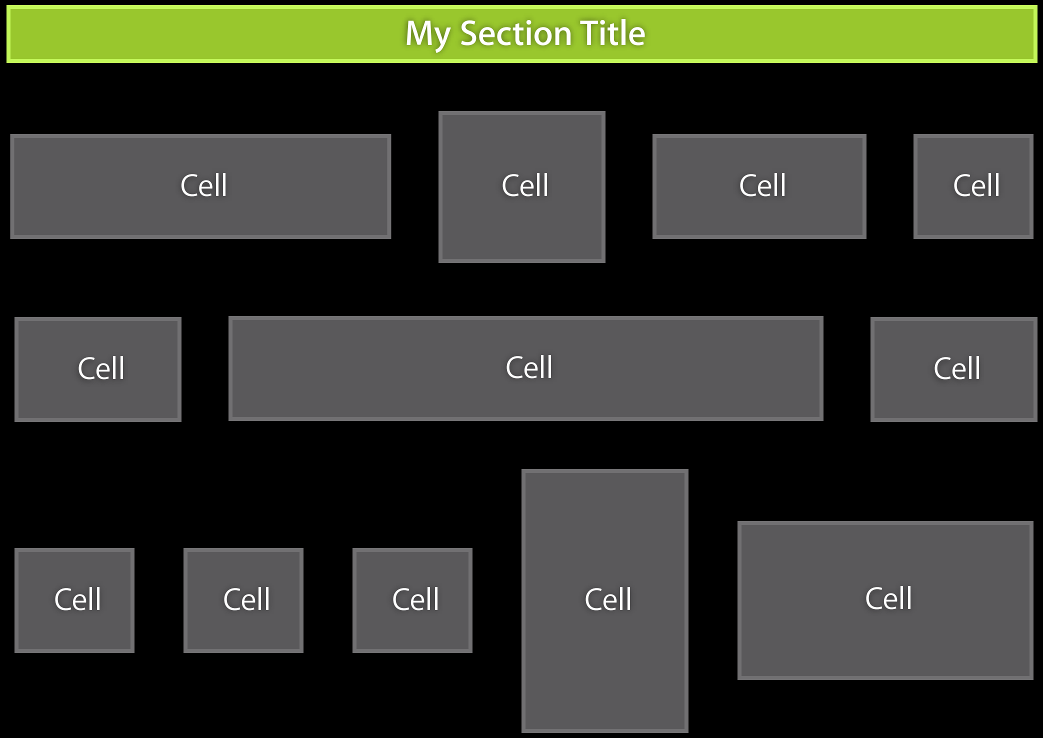
Laying Out the Collection



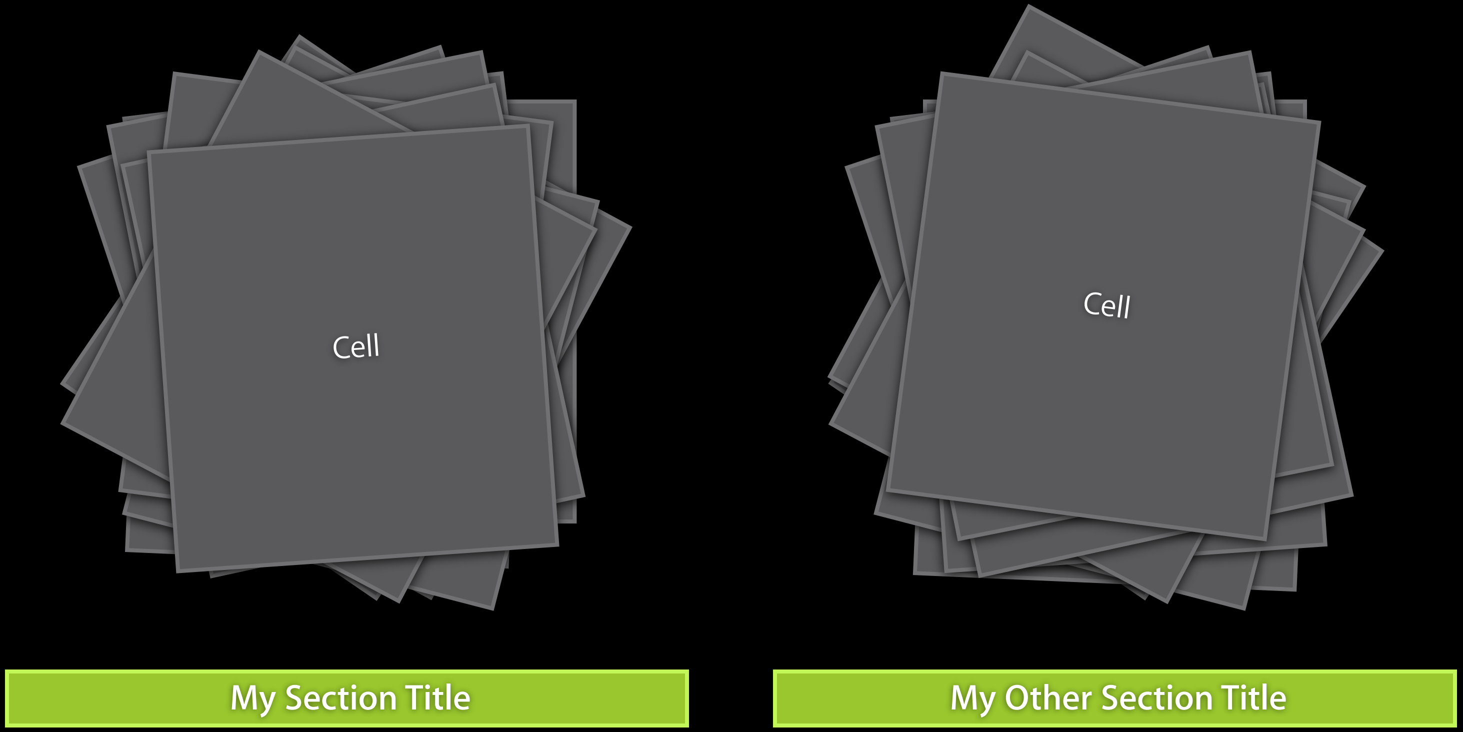
UICollectionViewFlowLayout

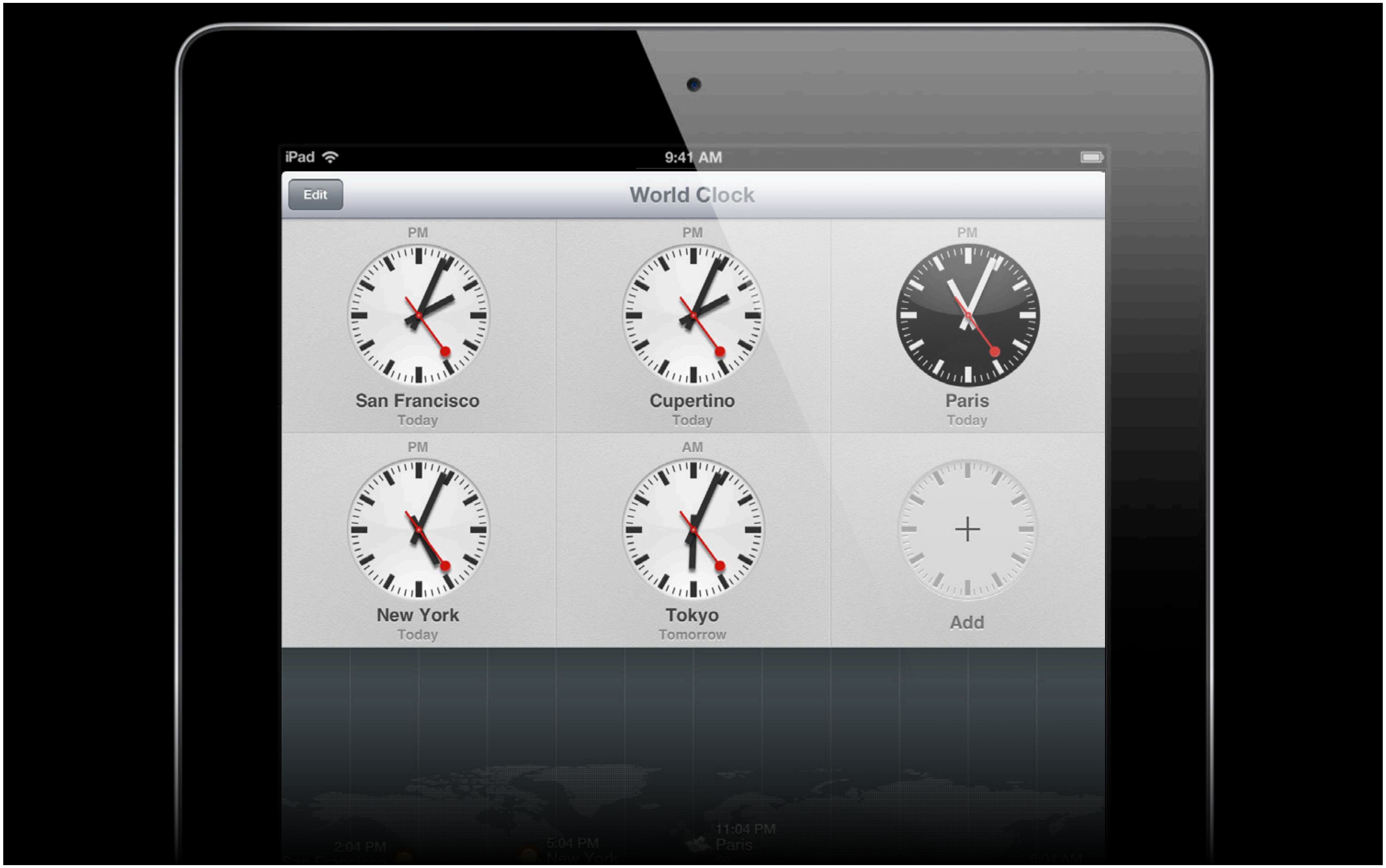


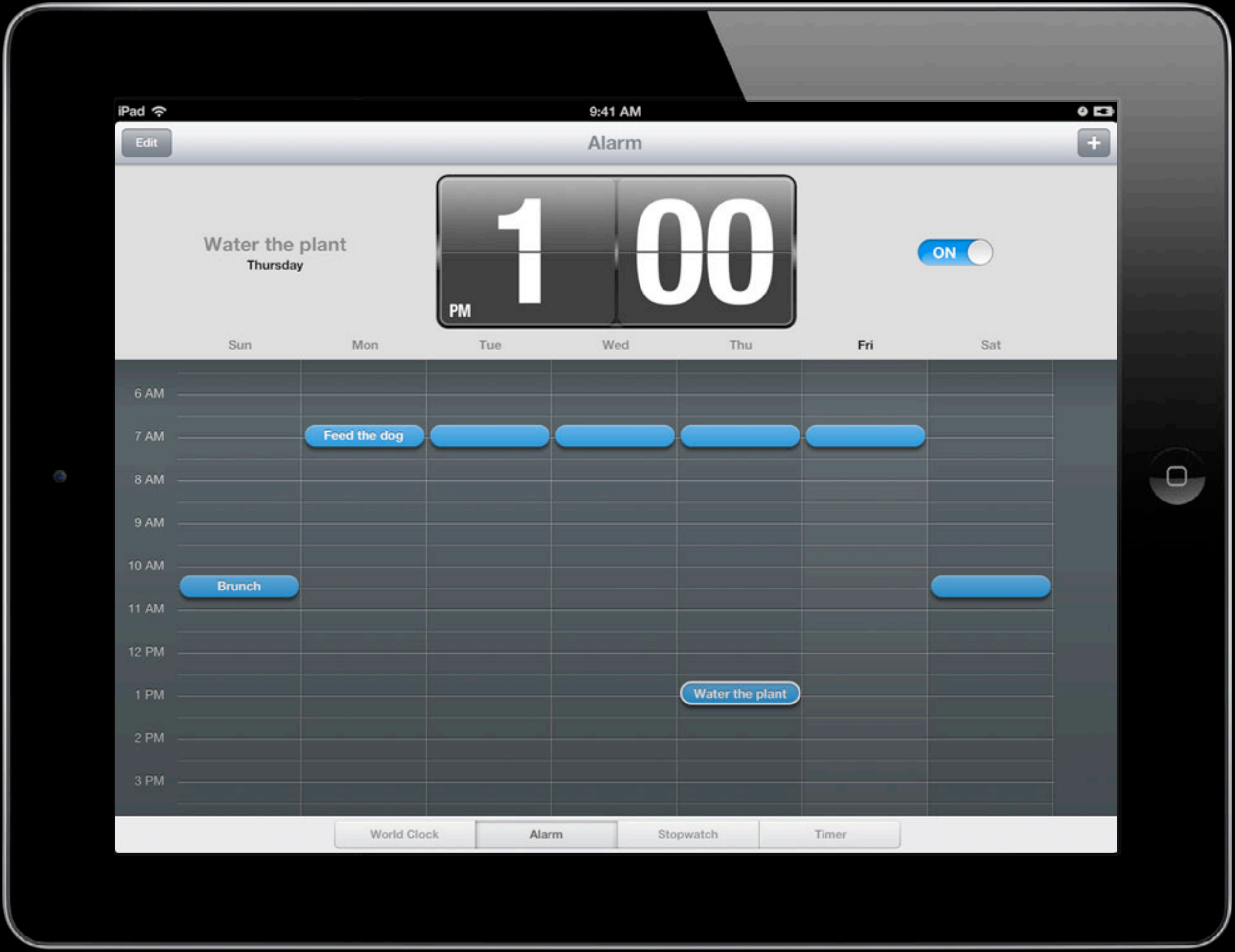
UICollectionViewFlowLayout



Custom UICollectionViewLayout




iPad
9:41 AM
Edit
World Clock
PM
San Francisco
Today
PM
Cupertino
Today
PM
Paris
Today
PM
New York
Today
AM
Tokyo
Tomorrow
Add
2:04 PM
5:04 PM
New York
11:04 PM
Paris


iPad
9:41 AM
Edit
Alarm
+
Water the plant
Thursday
1 00
PM
ON
Sun
Mon
Tue
Wed
Thu
Fri
Sat
6 AM
7 AM
8 AM
9 AM
10 AM
11 AM
12 PM
1 PM
2 PM
3 PM
Feed the dog
Brunch
Water the plant
World Clock
Alarm
Stopwatch
Timer

UICollectionView Sessions

Introducing Collection Views	Presidio Tuesday 2:00PM
Advanced Collection Views and Building Custom Layouts	Mission Wednesday 11:30AM

UIViewController

View Unloading

```
- (void)viewWillUnload {
    // Unregister for notifications
    // Record some subview state
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Set some outlets to nil
}
```

View Unloading

```
- (void)viewWillUnload NS_DEPRECATED_IOS(5_0, 6_0);
```

```
- (void)viewDidUnload NS_DEPRECATED_IOS(3_0, 6_0);
```

Forwarding Callbacks

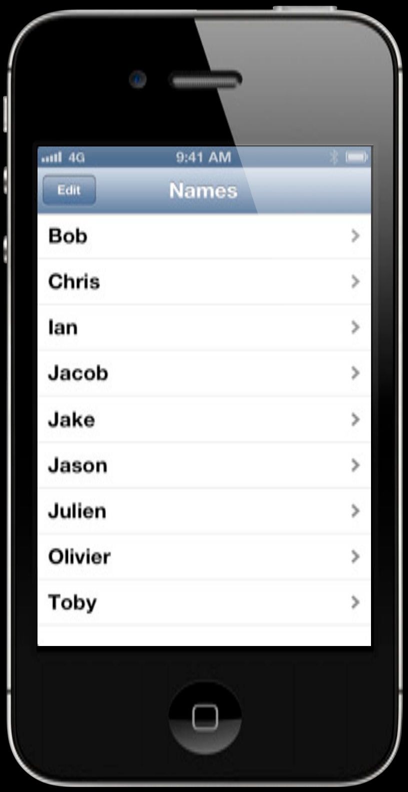
```
- (BOOL)automaticallyForwardAppearanceAndRotationMethodsToChildViewControllers;

- (BOOL)shouldAutomaticallyForwardRotationMethods;
- (BOOL)shouldAutomaticallyForwardAppearanceMethods;
```

Forwarding Callbacks

```
- (BOOL)automaticallyForwardAppearanceAndRotationMethodsToChildViewControllers;

- (BOOL)shouldAutomaticallyForwardRotationMethods;
- (BOOL)shouldAutomaticallyForwardAppearanceMethods;

- (void)beginAppearanceTransition:(BOOL)isAppearing
                         animated:(BOOL)animated;
- (void)endAppearanceTransition;
```

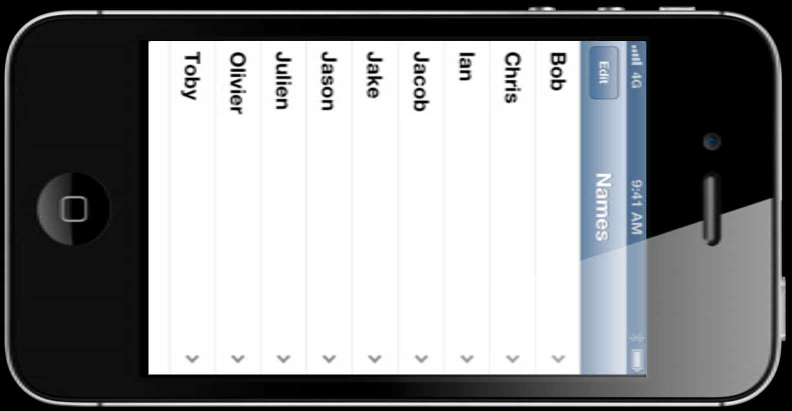
Rotation and Interface Orientation

shouldAutorotateToInterfaceOrientation:



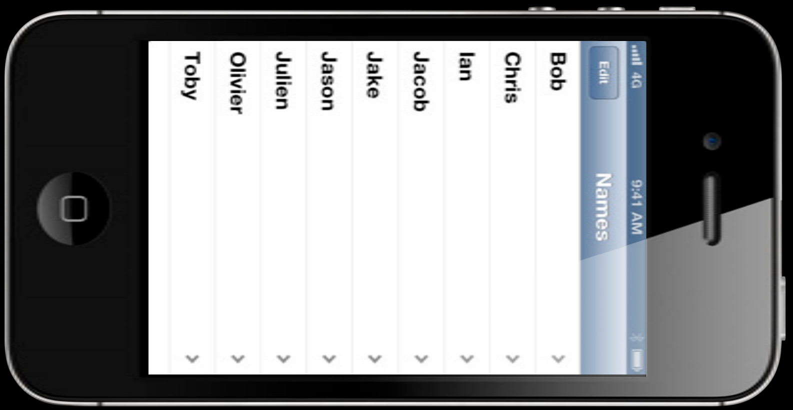
Rotation and Interface Orientation

shouldAutorotateToInterfaceOrientation:



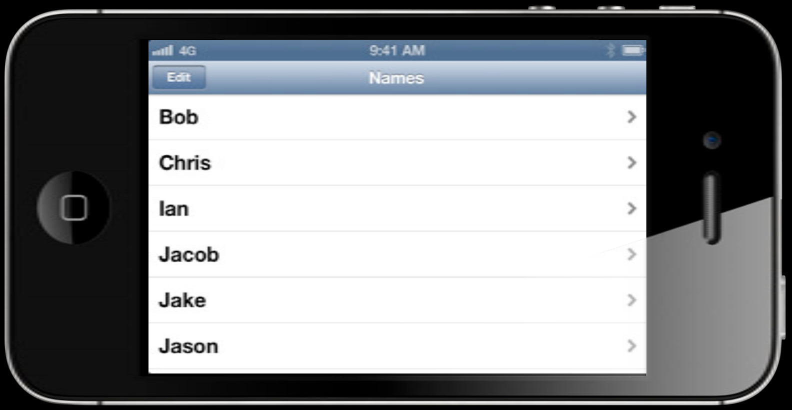
Rotation and Interface Orientation

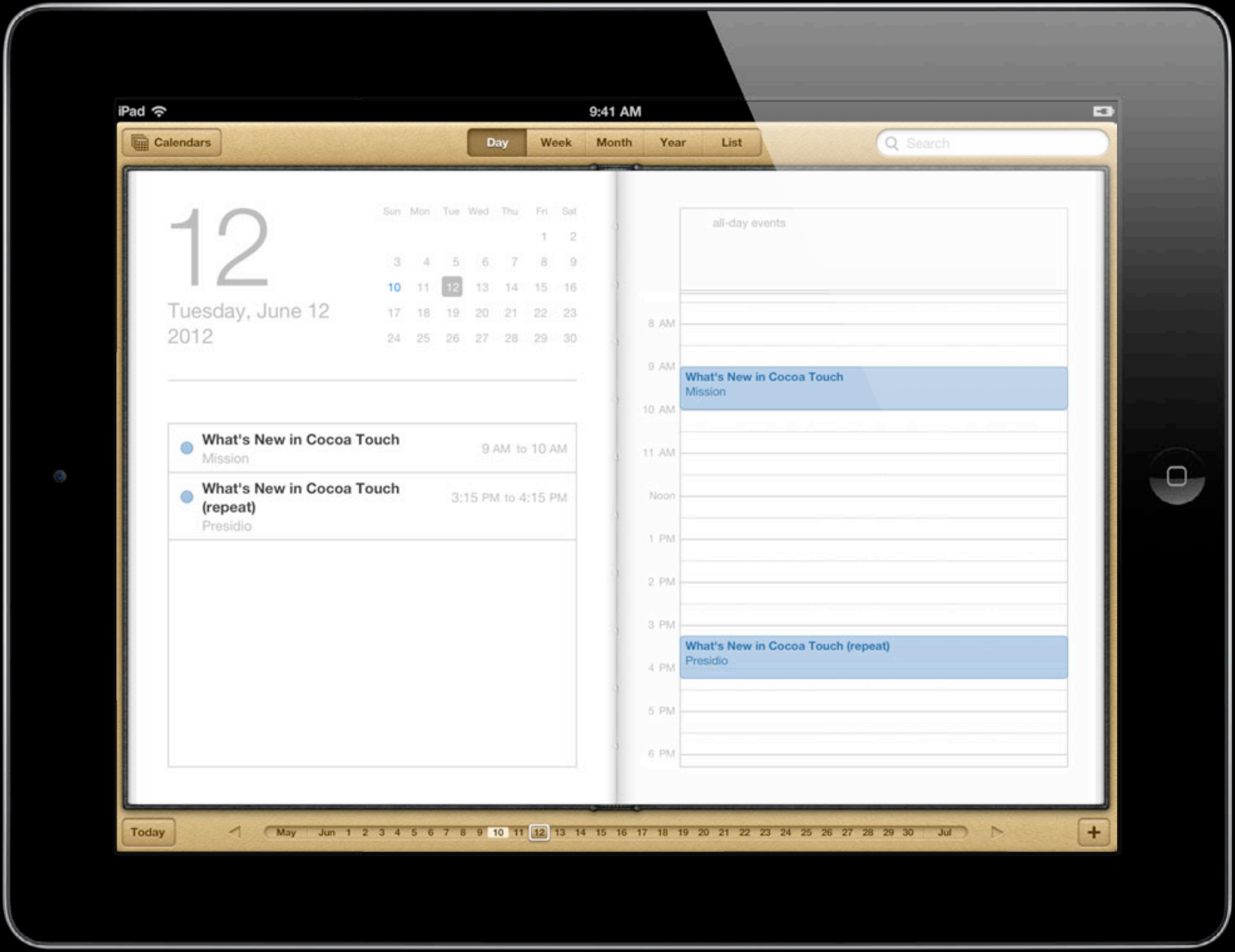
shouldAutorotateToInterfaceOrientation:



Rotation and Interface Orientation

shouldAutorotateToInterfaceOrientation:





12
Tuesday, June 12
2012

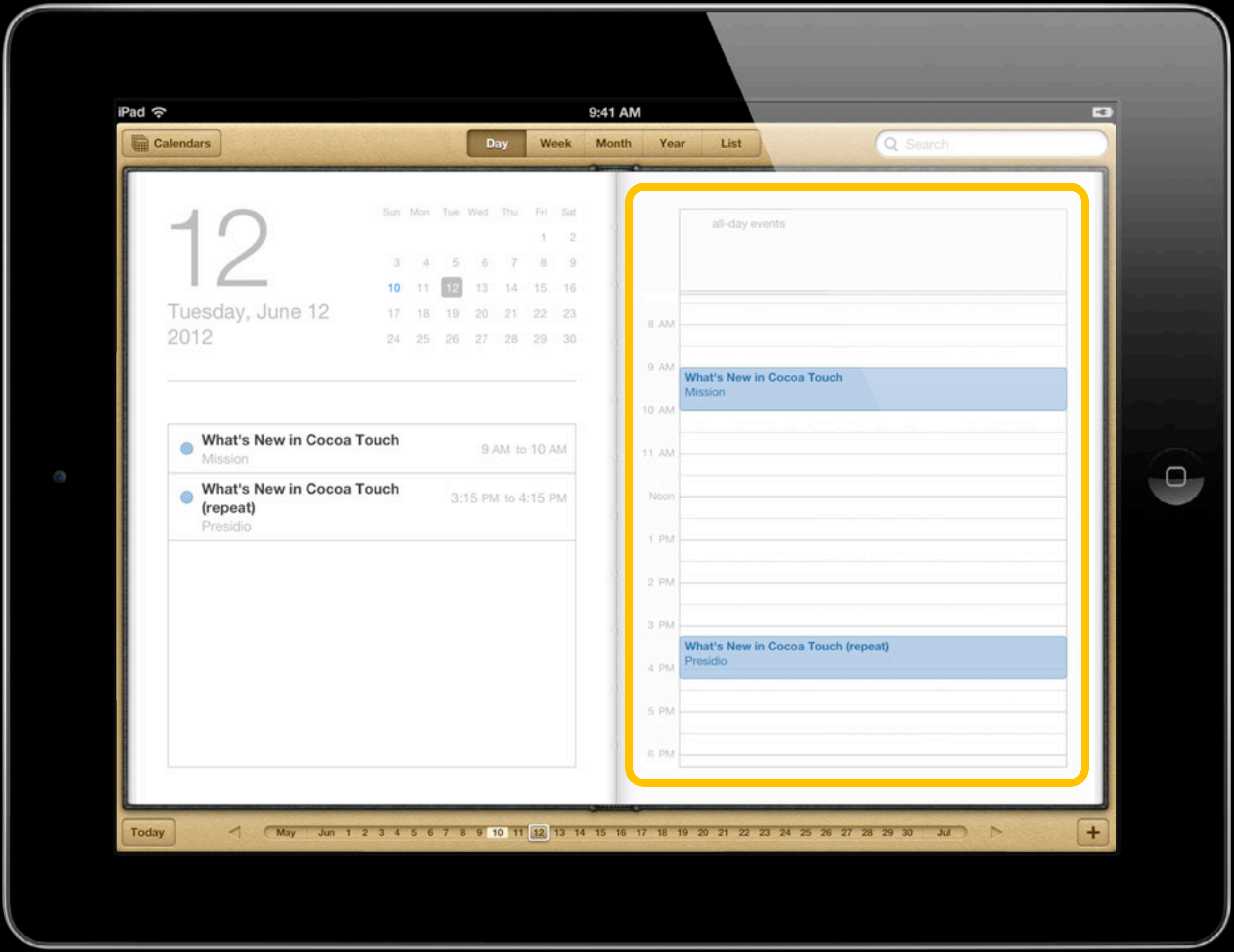
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

What's New in Cocoa Touch Mission	9 AM to 10 AM
What's New in Cocoa Touch (repeat) Presidio	3:15 PM to 4:15 PM

all-day events

8 AM
9 AM What's New in Cocoa Touch — Mission
10 AM
11 AM
Noon
1 PM
2 PM
3 PM What's New in Cocoa Touch (repeat) — Presidio
4 PM
5 PM
6 PM

May Jun 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 Jul



12
Tuesday, June 12
2012

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

What's New in Cocoa Touch — 9 AM to 10 AM
Mission

What's New in Cocoa Touch (repeat) — 3:15 PM to 4:15 PM
Presidio

all-day events

8 AM
9 AM
What's New in Cocoa Touch
Mission
10 AM
11 AM
Noon
1 PM
2 PM
3 PM
What's New in Cocoa Touch (repeat)
Presidio
4 PM
5 PM
6 PM

May Jun 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 Jul

Rotation and Interface Orientation

shouldAutorotateToInterfaceOrientation:

- Conflates interface orientation with rotation
- Conflates interface orientation with layout
- Interface orientation in many cases is meaningless
    - Child view controllers
    - Form sheets

Rotation and Interface Orientation

```
- (NSUInteger)supportedInterfaceOrientations;
- (UIInterfaceOrientation)preferredInterfaceOrientationForPresentation;
```

Rotation and Interface Orientation

```
- (NSUInteger)supportedInterfaceOrientations;
- (UIInterfaceOrientation)preferredInterfaceOrientationForPresentation;


- (NSUInteger)supportedInterfaceOrientationsForWindow:(UIWindow *)window;
```

Rotation and Interface Orientation

```
- (NSUInteger)supportedInterfaceOrientations;
- (UIInterfaceOrientation)preferredInterfaceOrientationForPresentation;
```

```
- (NSUInteger)supportedInterfaceOrientationsForWindow:(UIWindow *)window;
```

```
UIApplicationSupportedInterfaceOrientationsIsEnabled
```

UIViewController Sessions

The Evolution of View Controllers on iOS	Mission Thursday 2:00PM

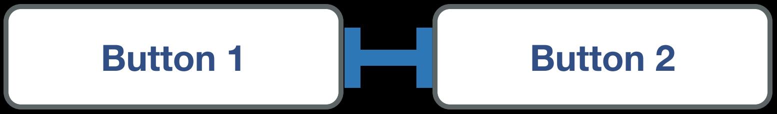
Auto Layout

View Layout

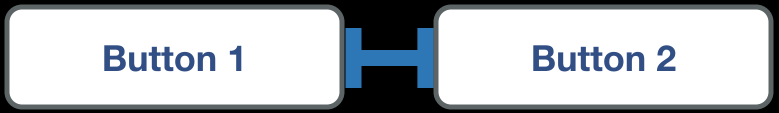
- Explicit layout
  - `- (void)setFrame:(CGRect)newFrame;`
- Autoresizing Masks
  - `UIViewAutoresizingFlexibleRightMargin`
  - `UIViewAutoresizingFlexibleTopMargin`

View Layout

- Explicit layout
  - `- (void)setFrame:(CGRect)newFrame;`
- Autoresizing Masks
  - `UIViewAutoresizingFlexibleRightMargin`
  - `UIViewAutoresizingFlexibleTopMargin`



Constraints



- A way to describe the relationships between objects
- The frames are calculated automatically when layout changes

Describing Constraints

NSLayoutConstraint

```
+ (id)constraintWithItem:(id)item1
               attribute:(NSLayoutAttribute)attribute1
               relatedBy:(NSLayoutRelation)relation
                  toItem:(id)item2
               attribute:(NSLayoutAttribute)attribute2
              multiplier:(CGFloat)multiplier
                constant:(CGFloat)constant;
```

Describing Constraints

NSLayoutConstraint

```
+ (id)constraintWithItem:(id)item1
               attribute:(NSLayoutAttribute)attribute1
               relatedBy:(NSLayoutRelation)relation
                  toItem:(id)item2
               attribute:(NSLayoutAttribute)attribute2
              multiplier:(CGFloat)multiplier
                constant:(CGFloat)constant;
```

item1.attribute1 = multiplier × item2.attribute2 + constant

Describing Constraints

NSLayoutConstraint

```
+ (id)constraintWithItem:(id)item1
               attribute:(NSLayoutAttribute)attribute1
               relatedBy:(NSLayoutRelation)relation
                  toItem:(id)item2
               attribute:(NSLayoutAttribute)attribute2
              multiplier:(CGFloat)multiplier
                constant:(CGFloat)constant;
```

item1.attribute1 ≥ multiplier × item2.attribute2 + constant

Describing Constraints

NSLayoutConstraint

```
+ (id)constraintWithItem:(id)item1
               attribute:(NSLayoutAttribute)attribute1
               relatedBy:(NSLayoutRelation)relation
                  toItem:(id)item2
               attribute:(NSLayoutAttribute)attribute2
              multiplier:(CGFloat)multiplier
                constant:(CGFloat)constant;
```

$$\text{item1.attribute1} \leq \text{multiplier} \times \text{item2.attribute2} + \text{constant}$$

Describing Constraints

NSLayoutConstraint

```
+ (id)constraintWithItem:(id)item1
               attribute:(NSLayoutAttribute)attribute1
               relatedBy:(NSLayoutRelation)relation
                  toItem:(id)item2
               attribute:(NSLayoutAttribute)attribute2
              multiplier:(CGFloat)multiplier
                constant:(CGFloat)constant;
```

$$\text{item1.attribute1} = \text{multiplier} \times \text{item2.attribute2} + \text{constant}$$

Constraints for Buttons

```
NSLayoutConstraint *constraint =
[NSLayoutConstraint constraintWithItem:button1
                             attribute:NSLayoutAttributeRight
                             relatedBy:NSLayoutRelationEqual
                                toItem:button2
                             attribute:NSLayoutAttributeLeft
                            multiplier:1.0
                              constant:-20.0];


[closestCommonAncestor addConstraint:constraint];
```

Describing Constraints

NSLayoutConstraint.h

- Relations

```
typedef NS_ENUM(NSInteger, NSLayoutRelation) {
    NSLayoutRelationLessThanOrEqual = -1,
    NSLayoutRelationEqual = 0,
    NSLayoutRelationGreaterThanOrEqual = 1,
};
```

Describing Constraints

NSLayoutConstraint.h

- Attributes

```
typedef NS_ENUM(NSInteger, NSLayoutAttribute) {
    NSLayoutAttributeLeft = 1,
    NSLayoutAttributeRight,
    NSLayoutAttributeTop,
    NSLayoutAttributeBottom,
    NSLayoutAttributeLeading,
    NSLayoutAttributeTrailing,
    NSLayoutAttributeWidth,
    NSLayoutAttributeHeight,
    NSLayoutAttributeCenterX,
    NSLayoutAttributeCenterY,
    NSLayoutAttributeBaseline,

    NSLayoutAttributeNotAnAttribute = 0
 };
```

Constraints

Constraints

- Visual format

  -
    ```
    + (NSArray *)constraintsWithVisualFormat:(NSString *)format
                 options:(NSLayoutFormatOptions)opts
                 metrics:(NSDictionary *)metrics
                 views:(NSDictionary *)views;
    ```

Constraints

- Visual format

    -
      ```
      + (NSArray *)constraintsWithVisualFormat:(NSString *)format
                   options:(NSLayoutFormatOptions)opts
                   metrics:(NSDictionary *)metrics
                   views:(NSDictionary *)views;
      ```
- Optional

Constraints

- Visual format

  -
    ```
    + (NSArray *)constraintsWithVisualFormat:(NSString *)format
                                     options:(NSLayoutFormatOptions)opts
                                     metrics:(NSDictionary *)metrics
                                       views:(NSDictionary *)views;
    ```
- Optional
- Prioritized

Constraints

- Visual format

  -
    ```
    + (NSArray *)constraintsWithVisualFormat:(NSString *)format
                 options:(NSLayoutFormatOptions)opts
                 metrics:(NSDictionary *)metrics
                 views:(NSDictionary *)views;
    ```
- Optional
- Prioritized
- Maximums and minimums

Constraints

- Visual format

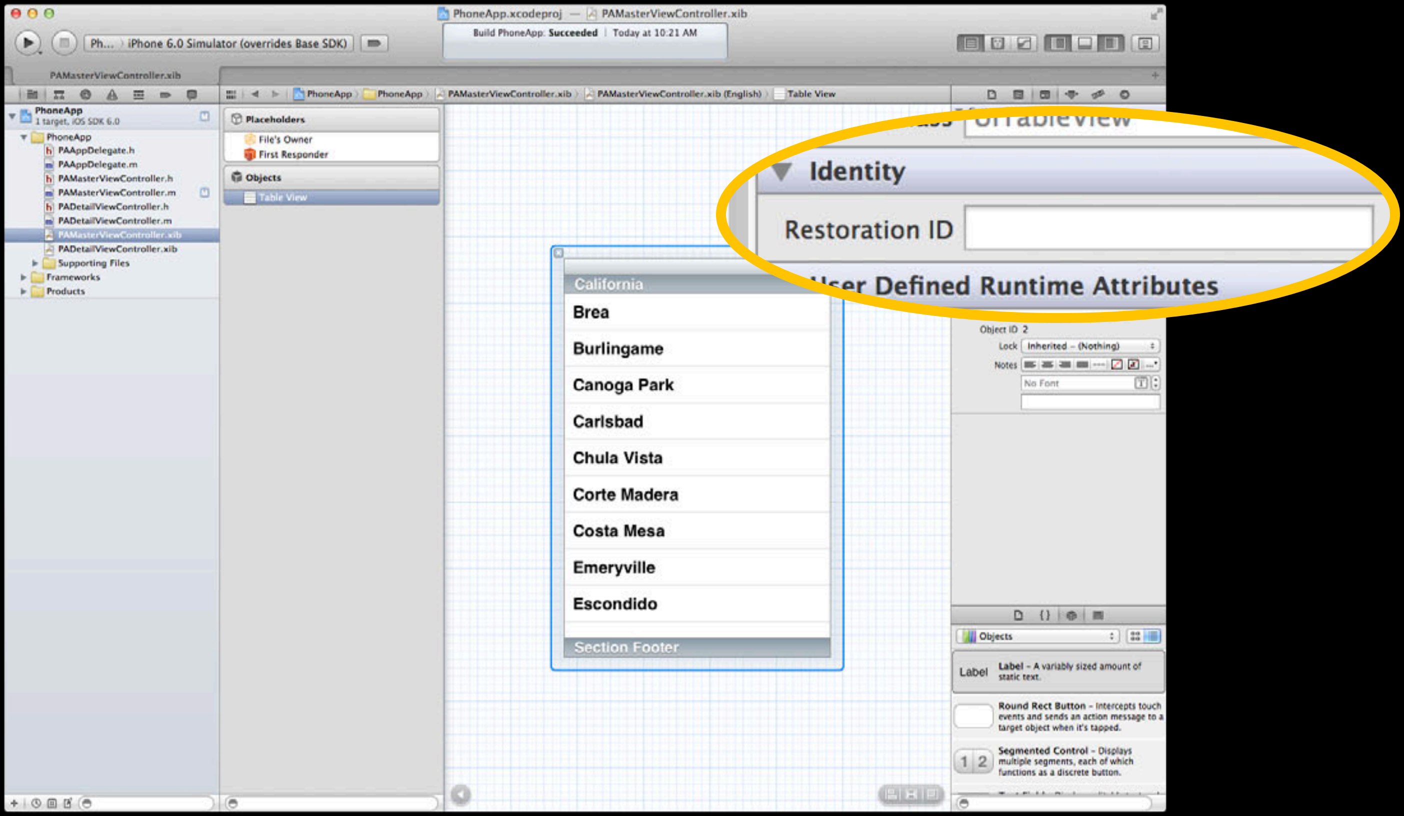
  -
    ```
    + (NSArray *)constraintsWithVisualFormat:(NSString *)format
                          options:(NSLayoutFormatOptions)opts
                          metrics:(NSDictionary *)metrics
                          views:(NSDictionary *)views;
    ```
- Optional
- Prioritized
- Maximums and minimums
- Apply to any two views

Auto Layout Sessions

Introduction to Auto Layout for iOS and OS X	Mission Tuesday 10:15AM

State
Restoration

Restoration Identifiers


PhoneApp.xcodeproj — PAMasterViewController.xib
Build PhoneApp: Succeeded | Today at 10:21 AM
iPhone 6.0 Simulator (overrides Base SDK)
PhoneApp
PAAppDelegate.h
PAAppDelegate.m
PAMasterViewController.h
PAMasterViewController.m
PADetailViewController.h
PADetailViewController.m
PAMasterViewController.xib
PADetailViewController.xib
Supporting Files
Frameworks
Products
Placeholders
File's Owner
First Responder
Objects
Table View
California
Brea
Burlingame
Canoga Park
Carlsbad
Chula Vista
Corte Madera
Costa Mesa
Emeryville
Escondido
Section Footer
Identity
Restoration ID
User Defined Runtime Attributes
Object ID 2
Lock Inherited - (Nothing)
Notes
No Font
Objects
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.

Restoration Identifiers

```
@property (nonatomic, copy) NSString *restorationIdentifier;
```

State Preservation

- User hits the home button
  - `- (void)encodeRestorableStateWithCoder:(NSCoder *)coder;`

- (call super!)

State Restoration

State Restoration

- Delegate method gets called

```
– (BOOL)application:(UIApplication *)app
        willFinishLaunchingWithOptions:(NSDictionary *)options;
```

State Restoration

- Delegate method gets called

```
- (BOOL)application:(UIApplication *)app
        willFinishLaunchingWithOptions:(NSDictionary *)options;
```

- State restoration process begins

```
+ (UIViewController *)viewControllerWithRestorationIdentifierPath:
                                                   (NSArray *)identifierComponents
                      coder:(NSCoder *)coder;
- (void)decodeRestorableStateWithCoder:(NSCoder *)coder;
```

State Preservation & Restoration Sessions

Saving and Restoring Application State on iOS	Presidio Tuesday 4:30PM
Saving and Restoring Application State on iOS	Russian Hill Thursday 3:15PM

NSAttributedString

Creating Attributed Strings

```
NSAttributedString *s = [[NSAttributedString alloc]
     initWithString:@”Welcome to WWDC!”
         attributes:@{ NSFontAttributeName : [UIFont systemFontOfSize:36.0f],
                       NSUnderlineStyleAttributeName : @1 }];
```

Welcome to WWDC!

NSMutableAttributedString

```
- (void)addAttributes:(NSDictionary *)attrs range:(NSRange)range;
- (void)addAttribute:(NSString *)key value:(id)value range:(NSRange)range;
- (void)removeAttribute:(NSString *)key range:(NSRange)range;
```

Attributes

#import <UIKit/NSAttributedString.h>

```
NSString *const NSFontAttributeName;
NSString *const NSParagraphStyleAttributeName;
NSString *const NSForegroundColorAttributeName;
NSString *const NSBackgroundColorAttributeName;
NSString *const NSLigatureAttributeName;
NSString *const NSBaselineOffsetAttributeName;
NSString *const NSStrikethroughStyleAttributeName;
NSString *const NSUnderlineStyleAttributeName;
NSString *const NSStrokeColorAttributeName;
NSString *const NSStrokeWidthAttributeName;
NSString *const NSShadowAttributeName;
```

String Drawing

#import <UIKit/NSStringDrawing.h>

```
@interface NSAttributedString (NSStringDrawing)

- (CGSize)size;
- (void)drawAtPoint:(CGPoint)point;
- (void)drawInRect:(CGRect)rect;

@end
```

String Drawing

#import <UIKit/NSStringDrawing.h>

```
@interface NSAttributedString (NSExtendedStringDrawing)

- (void)drawWithRect:(CGRect)rect
             options:(NSStringDrawingOptions)options
             context:(NSStringDrawingContext *)context;

- (CGRect)boundingRectWithSize:(CGSize)size
             options:(NSStringDrawingOptions)options
             context:(NSStringDrawingContext *)context;

@end
```

String Drawing

#import <UIKit/NSStringDrawing.h>

```
typedef NS_ENUM(NSInteger, NSStringDrawingOptions) {
    NSStringDrawingTruncatesLastVisibleLine = 1 << 5,
    NSStringDrawingUsesLineFragmentOrigin = 1 << 0,
    NSStringDrawingUsesFontLeading = 1 << 1,
    NSStringDrawingUsesDeviceMetrics = 1 << 3,
};
```

String Drawing

#import <UIKit/NSStringDrawing.h>

```
@interface NSStringDrawingContext : NSObject

@property(nonatomic) CGFloat minimumScaleFactor;
@property(nonatomic) CGFloat minimumTrackingAdjustment;

@property(nonatomic, readonly) CGFloat actualScaleFactor;
@property(nonatomic, readonly) CGFloat actualTrackingAdjustment;

@property(nonatomic, readonly) CGRect totalBounds;

@end
```

Attributed Strings Sessions

Introduction to Attributed Strings for iOS	Mission Wednesday 3:15PM
Advanced Attributed Strings for iOS	Mission Thursday 10:15AM

Other Sessions

Social

- New social services
  - Facebook
  - Sina Weibo
- Works with Accounts

Game Center

- Challenges
- Consolidated view controller
- Control over authentication UI

Maps

- Launch Maps
- Indicate points of interest
- Transit apps

PassKit

- Downloadable cards
  - Coupons
  - Boarding passes
  - Event tickets
- Pushed to device
- Signed

In-App Purchase

- SKDownload
- SKPaymentTransaction
- Content purchase from within the app
- Apple can host your content

EventKit

- Proximity
- Time-based alarms

Data Privacy

- Messages for your users
- Presenting (and not presenting) UI

Related Sessions

Building Concurrent User Interfaces on iOS	Pacific Heights Wednesday 9:00AM
Keyboard Input in iOS	Russian Hill Wednesday 2:00PM
Enhancing User Experience with Scroll Views	Presidio Wednesday 3:15PM
Up and Running: Making a Great Impression with Every Launch	Nob Hill Wednesday 4:30PM
Polishing Your Interface Rotations	Mission Thursday 4:30PM
Building Advanced Gesture Recognizers	Marina Thursday 11:30AM
Internationalization Tips and Tricks	Marina Friday 10:15AM

More Information

Jake Behrens
UI Frameworks Evangelist
behrens@apple.com

Documentation
http://developer.apple.com/

Apple Developer Forums
http://devforums.apple.com/

WWDC2012

[redacted]