

Introducing UITableView

David Smith

Introducing UITableView

David Smith

Audience

- Targeted at Beginners
- Assumes basic familiarity with Objective-C / Cocoa

David Smith

- Independent iOS/Mac Developer for 4 years
- App Maker - Audiobooks & My Recipe Book
- Twitter: [_DavidSmith](#)
- Podcast: [DevelopingPerspective.com](#)
- Website: [david-smith.org](#)

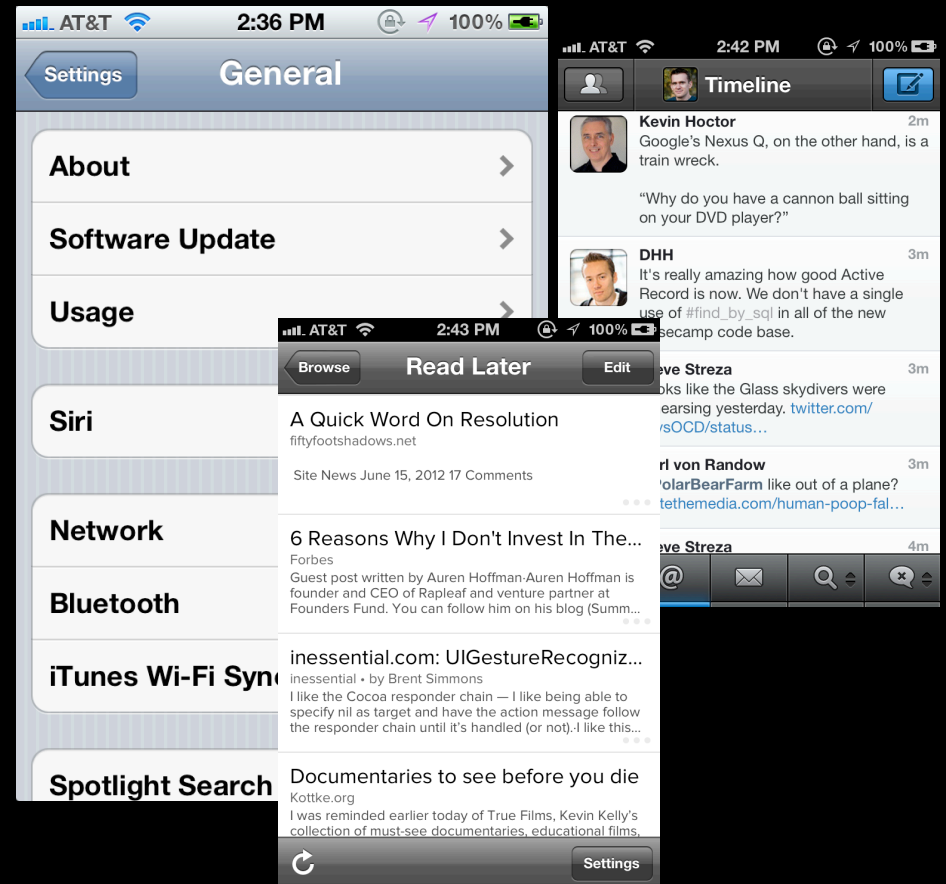


Format

- 40 mins
- Questions at the end

What is a UITableView?

- Most widely used data control on iOS

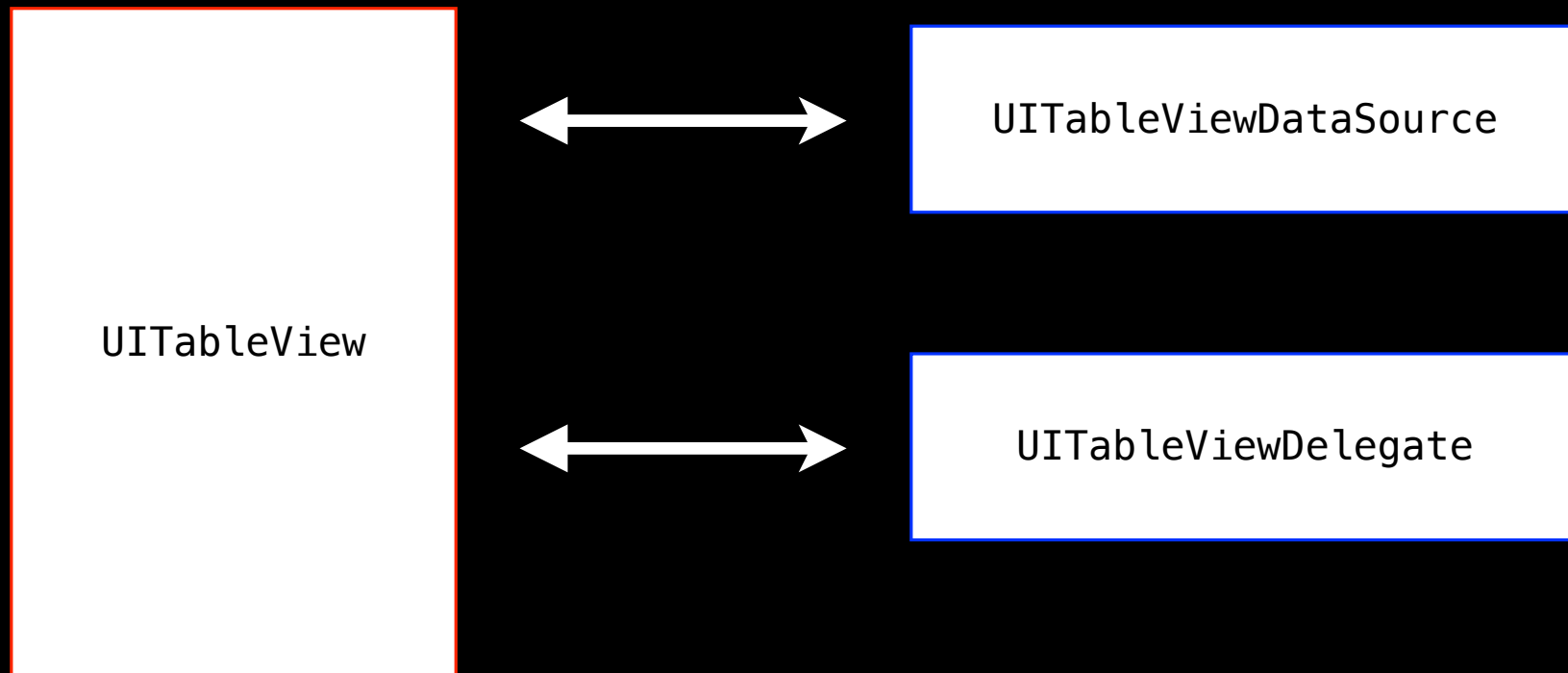


What is a UITableView?

- A vertically scrolling collection of data / views
- Designed for Performance & Simplicity
- Highly Flexible in terms of appearance and content



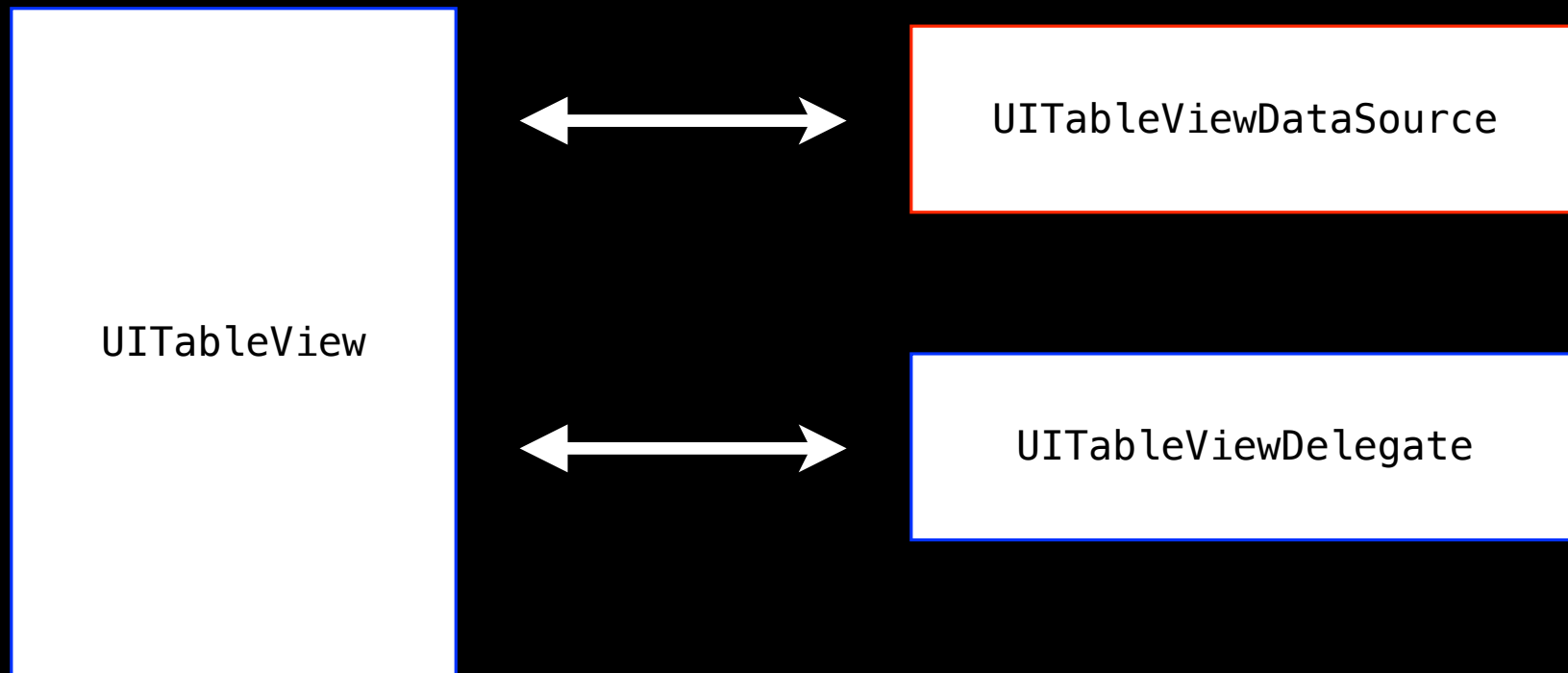
Setting Up a UITableView



UITableView

- The visual display
- Added to your application's view hierarchy
- Doesn't have to be fullscreen

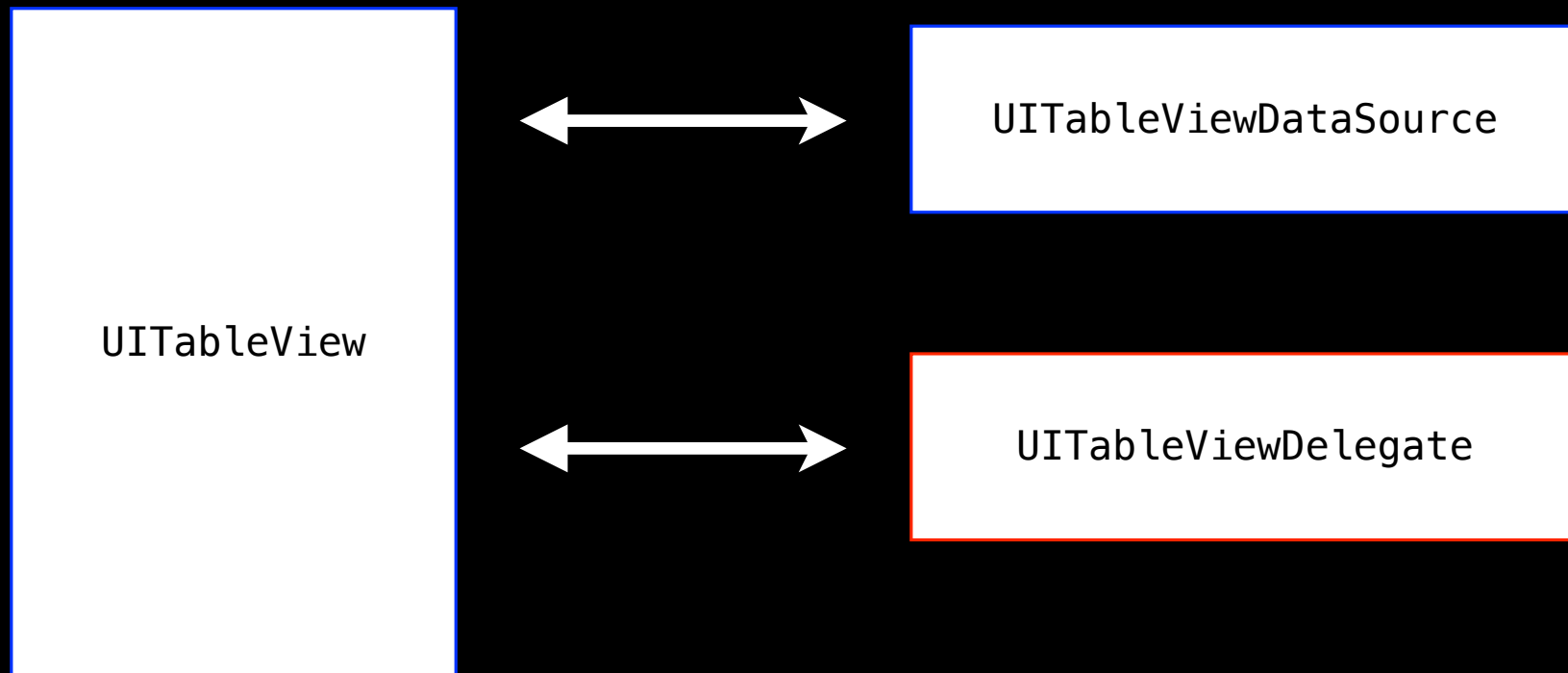
Setting Up a UITableView



UITableViewDataSource

- Configures the contents of the UITableView
- Requires:
 - `tableView:numberOfRowsInSection:`
 - `tableView:cellForRowAtIndexPath:`

Setting Up a UITableView



UITableViewDelegate

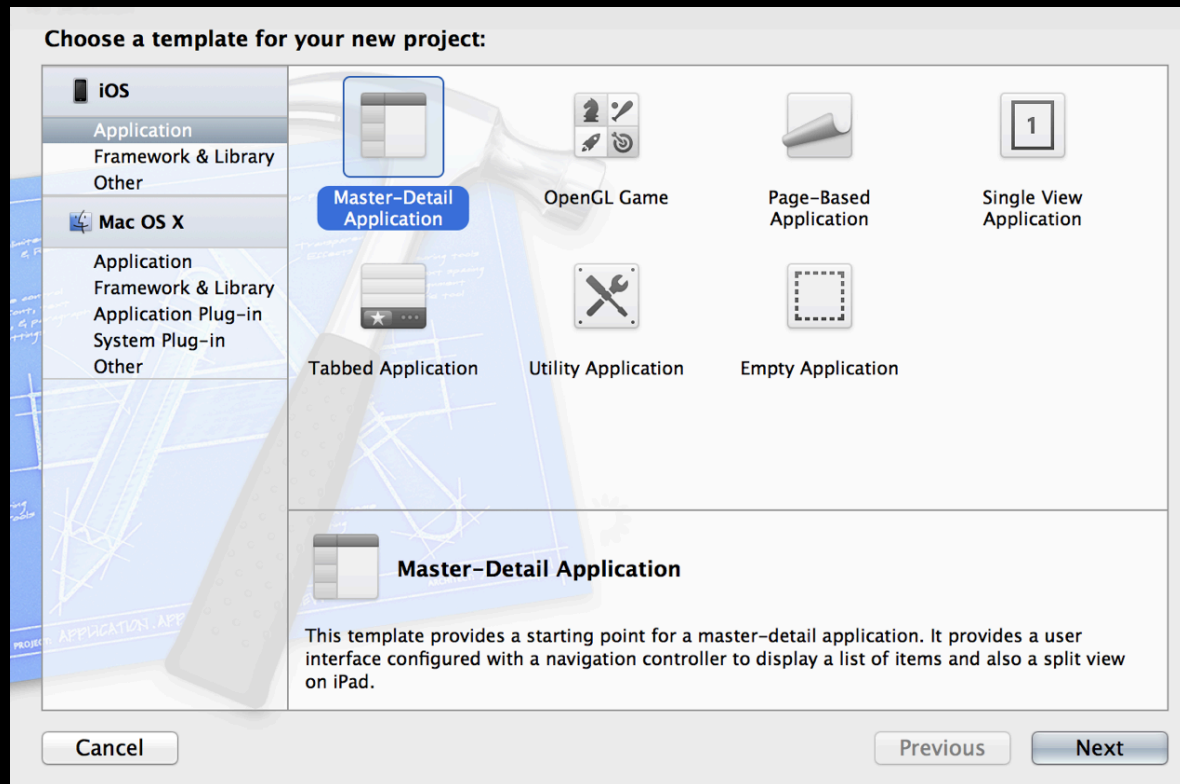
- Configures the appearance and interactions of the UITableView
- No required methods, some commonly used ones:
 - `tableView:heightForRowAtIndexPath:`
 - `tableView:didSelectRowAtIndexPath:`

UITableViewController

- Base implementation
- Conforms to both UITableViewDelegate and UITableViewDataSource
- Typically used within a UINavigationController
- Handles full life-cycle of the UITableView

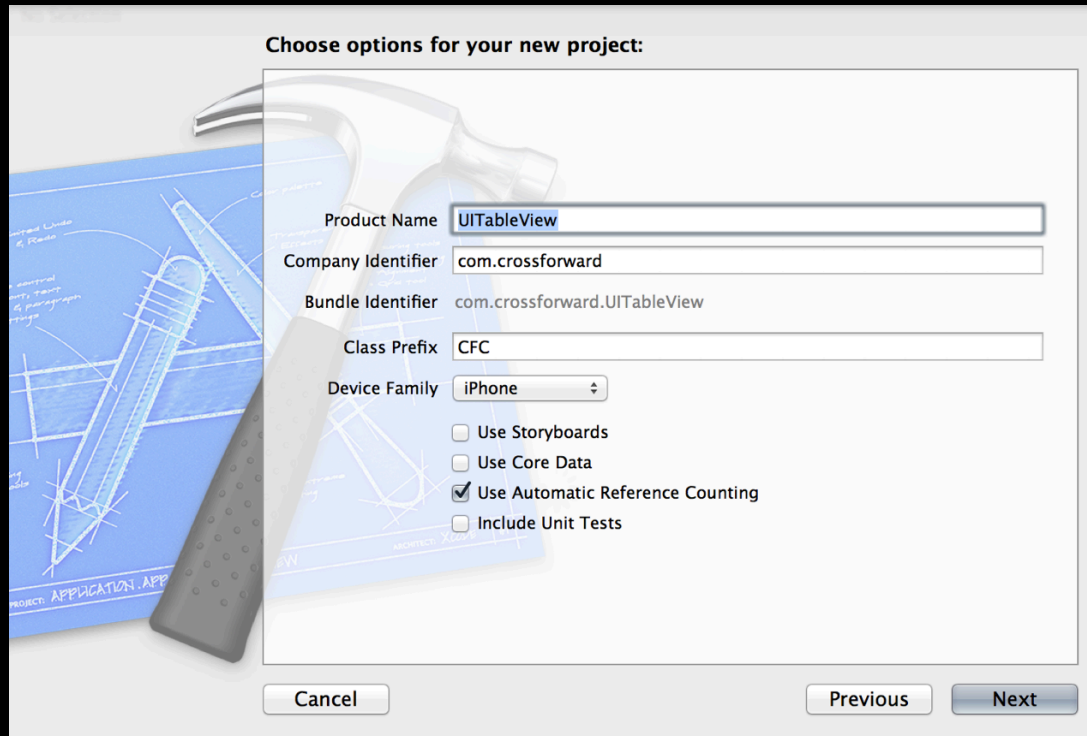
Super Simple Demo

Super Simple Demo



Super Simple Demo

Choose options for your new project:



Product Name

Company Identifier

Bundle Identifier

Class Prefix

Device Family

☐ Use Storyboards

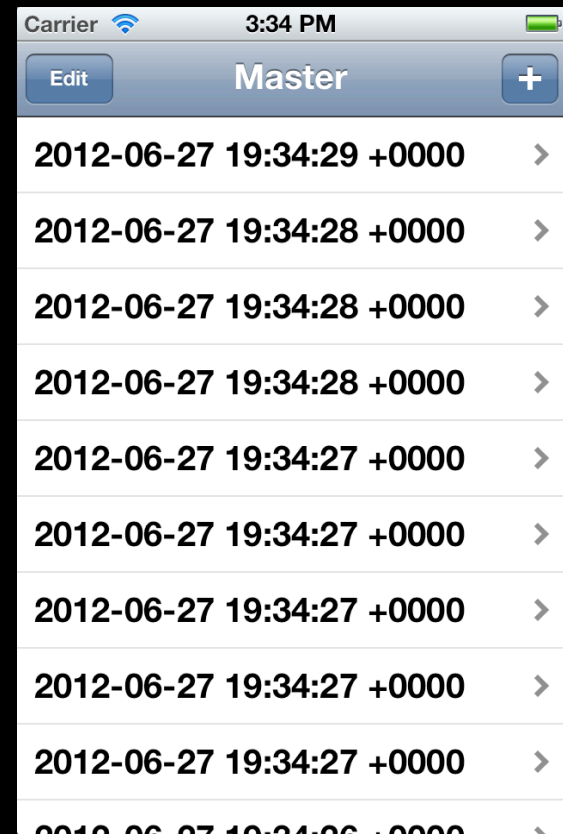
☐ Use Core Data

☒ Use Automatic Reference Counting

☐ Include Unit Tests

Super Simple Demo

- That's It.
- We have a UITableViewController

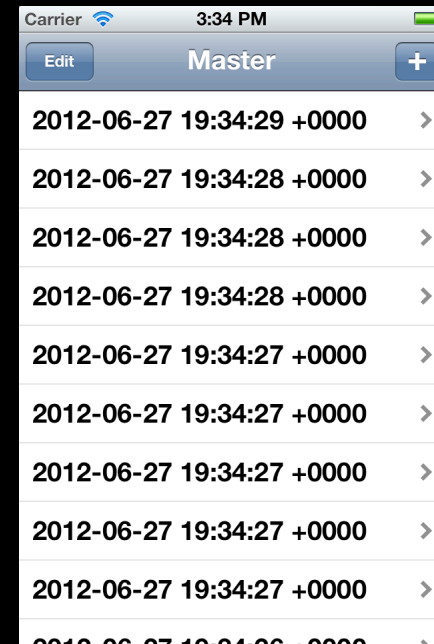


Styles

UITableViewStyleGrouped



UITableViewStylePlain



Super Simple Demo

```
- (NSInteger)tableView:(UITableView *)tableView  
numberOfRowsInSection(NSInteger)section  
{  
    return _objects.count;  
}
```

Super Simple Demo

```
- (UITableViewCell *)tableView:(UITableView *)tableView
    cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];

    if (cell == nil) {
        cell = [[UITableViewCell alloc]
            initWithStyle:UITableViewCellStyleDefault
            reuseIdentifier:CellIdentifier];
        cell.accessoryType = UITableViewCellAccessoryDisclosureIndicator;
    }

    NSDate *object = [_objects objectAtIndex:indexPath.row];
    cell.textLabel.text = [object description];
    return cell;
}
```

Super Simple Demo

```
- (UITableViewCell *)tableView:(UITableView *)tableView
    cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];

    if (cell == nil) {
        cell = [[UITableViewCell alloc]
            initWithStyle:UITableViewCellStyleDefault
            reuseIdentifier:CellIdentifier];
        cell.accessoryType = UITableViewCellAccessoryDisclosureIndicator;
    }

    NSDate *object = [_objects objectAtIndex:indexPath.row];
    cell.textLabel.text = [object description];
    return cell;
}
```

Reuse

Active Cells



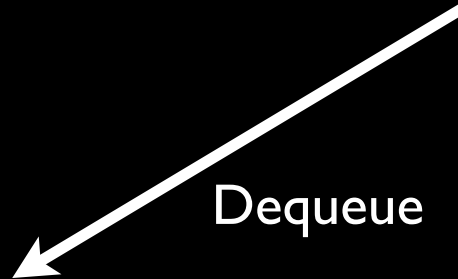
Scroll
Offscreen



Previously Used Cells



Dequeue



Super Simple Demo

```
- (UITableViewCell *)tableView:(UITableView *)tableView
    cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];

    if (cell == nil) {
        cell = [[UITableViewCell alloc]
            initWithStyle:UITableViewCellStyleDefault
            reuseIdentifier:CellIdentifier];
        cell.accessoryType = UITableViewCellAccessoryDisclosureIndicator;
    }

    NSDate *object = [_objects objectAtIndex:indexPath.row];
    cell.textLabel.text = [object description];
    return cell;
}
```


UITableViewCellStyle

- UITableViewCellStyleDefault
- UITableViewCellStyleSubtitle
- UITableViewCellStyleValue1
- UITableViewCellStyleValue2

Main Text



Main Text

Detail Text



Main Text

Detail Text



Main Text **Detail Text**



Super Simple Demo

```
- (UITableViewCell *)tableView:(UITableView *)tableView
    cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];

    if (cell == nil) {
        cell = [[UITableViewCell alloc]
            initWithStyle:UITableViewCellStyleDefault
            reuseIdentifier:CellIdentifier];
        cell.accessoryType = UITableViewCellAccessoryDisclosureIndicator;
    }

    NSDate *object = [_objects objectAtIndex:indexPath.row];
    cell.textLabel.text = [object description];
    return cell;
}
```

Accessory Styles

- UITableViewCellAccessoryDisclosureIndicator
- UITableViewCellAccessoryDetailDisclosureButton
- UITableViewCellAccessoryCheckmark



Super Simple Demo

```
- (UITableViewCell *)tableView:(UITableView *)tableView
    cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];

    if (cell == nil) {
        cell = [[UITableViewCell alloc]
            initWithStyle:UITableViewCellStyleDefault
            reuseIdentifier:CellIdentifier];
        cell.accessoryType = UITableViewCellAccessoryDisclosureIndicator;
    }

    NSDate *object = [_objects objectAtIndex:indexPath.row];
    cell.textLabel.text = [object description];
    return cell;
}
```

Responding to the User

UITableViewDelegate

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:
(NSIndexPath *)indexPath
{
    if (!self.detailViewController) {
        self.detailViewController = [[CFCDetailViewController alloc]
initWithNibName:@"CFCDetailViewController" bundle:nil];
    }
    NSDate *object = [_objects objectAtIndex:indexPath.row];
    self.detailViewController.detailItem = object;
    [self.navigationController
pushViewController:self.detailViewController animated:YES];
}
```

Responding to the User

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:
(NSIndexPath *)indexPath
{
    if (!self.detailViewController) {
        self.detailViewController = [[CFCDetailViewController alloc]
initWithNibName:@"CFCDetailViewController" bundle:nil];
    }
    NSDate *object = [_objects objectAtIndex:indexPath.row];
    self.detailViewController.detailItem = object;
    [self.navigationController
pushViewController:self.detailViewController animated:YES];
}
```

Responding to the User

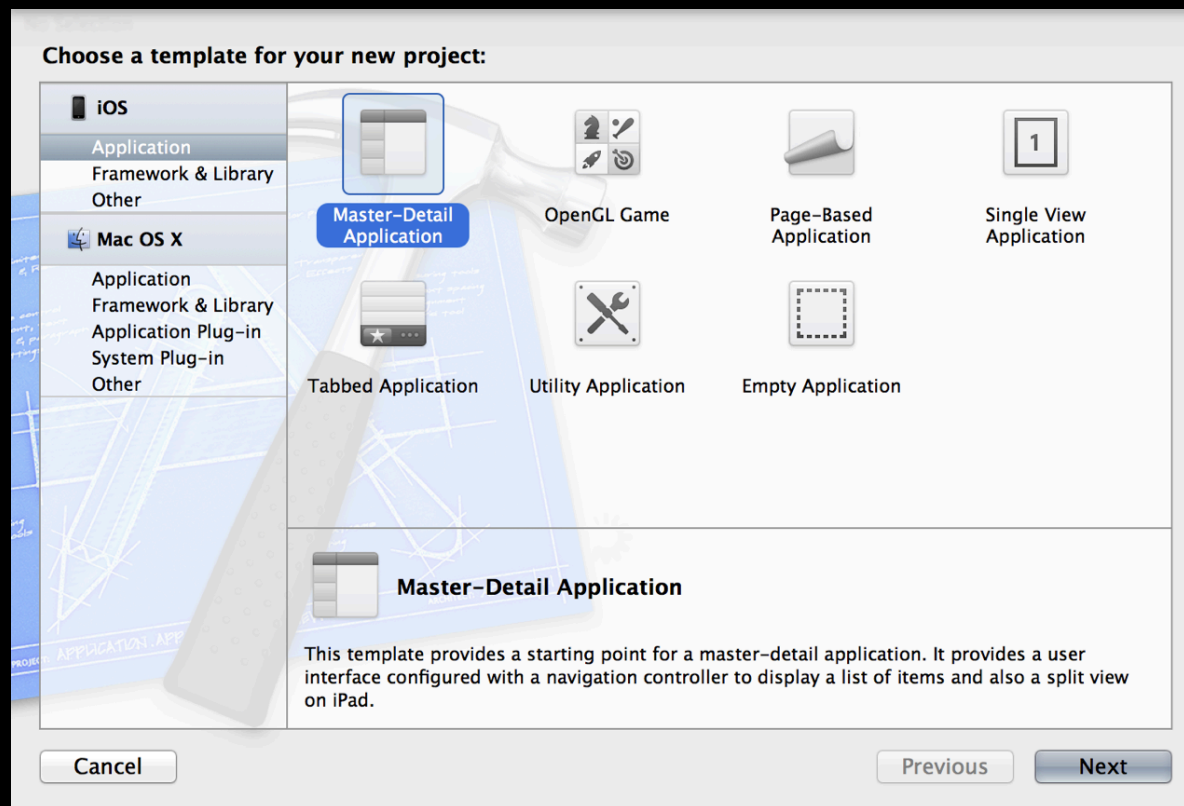
```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:
(NSIndexPath *)indexPath
{
    if (!self.detailViewController) {
        self.detailViewController = [[CFCDetailViewController alloc]
initWithNibName:@"CFCDetailViewController" bundle:nil];
    }
    NSDate *object = [_objects objectAtIndex:indexPath.row];
    self.detailViewController.detailItem = object;
    [self.navigationController
pushViewController:self.detailViewController animated:YES];
}
```

Super Simple Demo

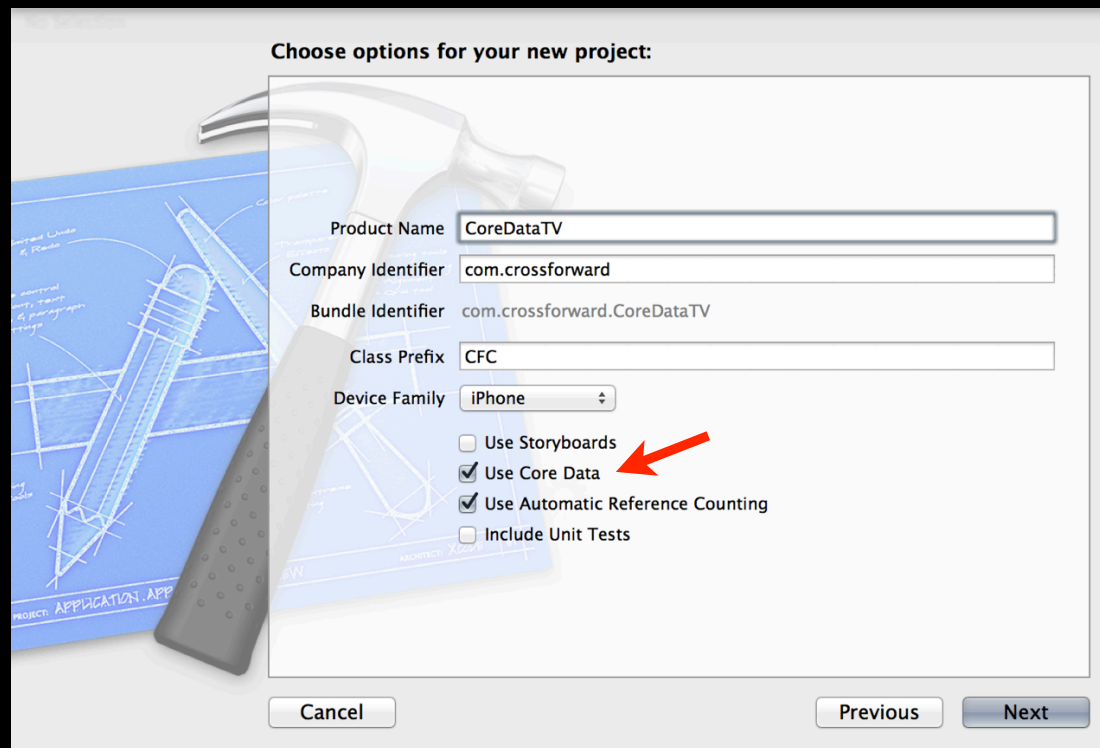
- Basic working UITableView
- Now...actual data.

Mostly Simple Demo

Mostly Simple Demo



Mostly Simple Demo



Mostly Simple Demo

```
@property (strong, nonatomic) NSFetchResultsController *fetchResultsController;  
@property (strong, nonatomic) NSManagedObjectContext *managedObjectContext;
```

More on Core Data:
Getting to Know Core Data
Whitney Young

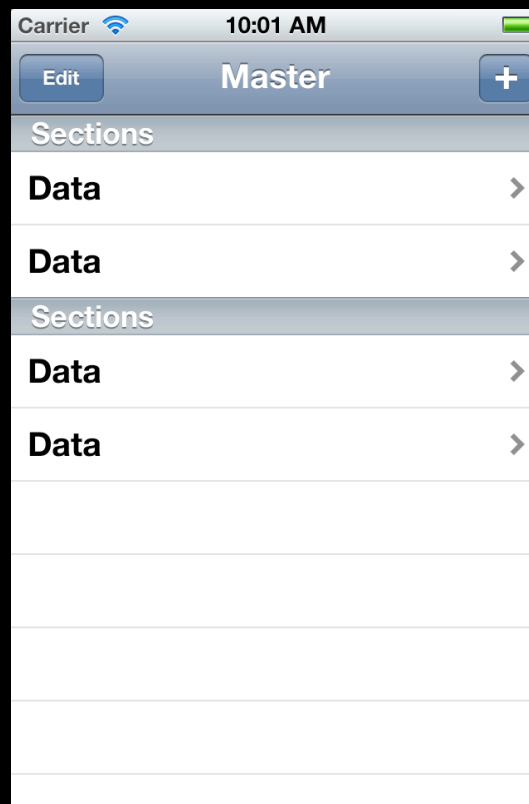
Saturday, at 2:45

How much data

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return [[self.fetchedResultsController sections] count];
}

- (NSInteger)tableView:(UITableView *)tableView
  numberOfRowsInSection:(NSInteger)section
{
    id <NSFetchedResultsControllerSectionInfo> sectionInfo =
    [[self.fetchedResultsController sections] objectAtIndex:section];
    return [sectionInfo numberOfObjects];
}
```

How much data



How much data

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return [[self.fetchedResultsController sections] count];
}

- (NSInteger)tableView:(UITableView *)tableView
  numberOfRowsInSection:(NSInteger)section
{
    id <NSFetchedResultsControllerSectionInfo> sectionInfo =
        [[self.fetchedResultsController sections] objectAtIndex:section];

    return [sectionInfo numberOfObjects];
}
```

Where the data comes from

```
- (NSFetchedResultsController *)fetchedResultsController
{
    if (__fetchedResultsController != nil) {
        return __fetchedResultsController;
    }

    NSFetchedRequest *fetchRequest = [[NSFetchedRequest alloc] init];

    // Edit the entity name as appropriate.
    NSEntityDescription *entity = [NSEntityDescription
        entityForName:@"Event"
        inManagedObjectContext:self.managedObjectContext];
    [fetchRequest setEntity:entity];

    // Set the batch size to a suitable number.
    [fetchRequest setFetchBatchSize:20];

    // Edit the sort key as appropriate.
    NSSortDescriptor *sortDescriptor = [[NSSortDescriptor alloc] initWithKey:@"timeStamp" ascending:NO];
    NSArray *sortDescriptors = [NSArray arrayWithObjects:sortDescriptor, nil];

    [fetchRequest setSortDescriptors:sortDescriptors];
}
```


Where the data comes from

```
- (NSFetchedResultsController *)fetchedResultsController
{
    if (__fetchedResultsController != nil) {
        return __fetchedResultsController;
    }

    NSFetchRequest *fetchRequest = [[NSFetchRequest alloc] init];

    // Edit the entity name as appropriate.
    NSEntityDescription *entity = [NSEntityDescription
        entityForName:@"Event"
        inManagedObjectContext:self.managedObjectContext];
    [fetchRequest setEntity:entity];

    // Set the batch size to a suitable number.
    [fetchRequest setFetchBatchSize:20];

    // Edit the sort key as appropriate.
    NSSortDescriptor *sortDescriptor = [[NSSortDescriptor alloc] initWithKey:@"timeStamp" ascending:NO];
    NSArray *sortDescriptors = [NSArray arrayWithObjects:sortDescriptor, nil];

    [fetchRequest setSortDescriptors:sortDescriptors];
}
```

Where the data comes from

```
- (NSFetchedResultsController *)fetchedResultsController
{
    if (__fetchedResultsController != nil) {
        return __fetchedResultsController;
    }

    NSFetchedRequest *fetchRequest = [[NSFetchedRequest alloc] init];

    // Edit the entity name as appropriate.
    NSEntityDescription *entity = [NSEntityDescription
        entityForName:@"Event"
        inManagedObjectContext:self.managedObjectContext];
    [fetchRequest setEntity:entity];

    // Set the batch size to a suitable number.
    [fetchRequest setFetchBatchSize:20];

    // Edit the sort key as appropriate.
    NSSortDescriptor *sortDescriptor = [[NSSortDescriptor alloc] initWithKey:@"timeStamp" ascending:NO];
    NSArray *sortDescriptors = [NSArray arrayWithObjects:sortDescriptor, nil];

    [fetchRequest setSortDescriptors:sortDescriptors];
}
```

Where the data comes from

```
- (NSFetchedResultsController *)fetchedResultsController
{
    if (__fetchedResultsController != nil) {
        return __fetchedResultsController;
    }

    NSFetchRequest *fetchRequest = [[NSFetchRequest alloc] init];

    // Edit the entity name as appropriate.
    NSEntityDescription *entity = [NSEntityDescription
        entityForName:@"Event"
        inManagedObjectContext:self.managedObjectContext];
    [fetchRequest setEntity:entity];

    // Set the batch size to a suitable number.
    [fetchRequest setFetchBatchSize:20];

    // Edit the sort key as appropriate.
    NSSortDescriptor *sortDescriptor = [[NSSortDescriptor alloc] initWithKey:@"timeStamp" ascending:NO];
    NSArray *sortDescriptors = [NSArray arrayWithObjects:sortDescriptor, nil];

    [fetchRequest setSortDescriptors:sortDescriptors];
}
```

Where the data comes from

```
// Edit the section name key path and cache name if appropriate.
// nil for section name key path means "no sections".
NSFetchedResultsController *aFetchedResultsController =
    [[NSFetchedResultsController alloc]
     initWithFetchRequest:fetchRequest
     managedObjectContext:self.managedObjectContext
     sectionNameKeyPath:nil
     cacheName:@"Master"];

aFetchedResultsController.delegate = self;
self.fetchedResultsController = aFetchedResultsController;

NSError *error = nil;
if (![self.fetchedResultsController performFetch:&error]) {
    NSLog(@"Unresolved error %@, %@", error, [error userInfo]);
    abort();
}

return __fetchedResultsController;
}
```

Getting the Data

```
// Customize the appearance of table view cells.
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";

    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:CellIdentifier];
        cell.accessoryType = UITableViewCellAccessoryDisclosureIndicator;
    }

    [self configureCell:cell atIndexPath:indexPath];
    return cell;
}
```

Getting the Data

```
- (void)configureCell:(UITableViewCell *)cell atIndexPath:(NSIndexPath *)indexPath
{
    NSManagedObject *object =
        [self.fetchedResultsController objectAtIndexPath:indexPath];

    cell.textLabel.text = [[object valueForKey:@"timeStamp"] description];
}
```

That is UITableViews

Other Abilities:

- Animatable Deletion / Insertion
- Managing Selection
- Re-ordering of data (automatically or via user input)

Further Reading

- [TableView Programming Guide for iOS](#)

Questions.