

View Controllers

1DevDay, 2011

Priya Rajagopal

Jeff Kelley

What are View Controllers?

- Manage your application's user interface
- Typically, represents a screenfull of content
- Provide application flow logic, rotation, etc.
- Special view controllers manage other view controllers
 - Navigation controllers manage view hierarchies
 - Tab bar controllers swap between views

View Controller Basics

- `view` property provides screen-sized `UIView`
 - Usually provided in a nib, but you can override `loadView`
 - When memory is low, the device sends a low memory warning
 - If the view is not currently being displayed, it is released
 - So, clean up anything inside a view in `viewDidUnload`.

View Controller Lifecycle

- View Controller is created
 - -initWithNibName:bundle:
- View Controller loads its view the first time the view is accessed
 - Either from nib or using -loadView
- View Controller Lifecycle methods fire as appropriate

View Controller Lifecycle Methods

- In approximate order:
 - viewDidLoad
 - viewWillAppear:
 - viewDidAppear:
 - viewWillDisappear:
 - viewDidDisappear:
 - viewWillUnload
 - viewDidUnload
- Not guaranteed to get viewDidUnload if the view controller is deallocated first

Modal View Controllers

- One view controller presents another one modally, with optional animation
 - `[self presentViewController:myController animated:YES];`
- Can use `modalPresentationStyle` and `modalTransitionStyle` to modify behavior

Outlets

- Outlets are set up in the nib
- Properties or instance variables
- Your responsibility to disconnect them in `-viewDidLoad`
- Typical pattern:

```
@property (strong) IBOutlet UILabel *myLabel;  
  
- (void)viewDidLoad {  
    [super viewDidLoad];  
    [self setMyLabel:nil];  
}
```

IBOutlet Demo