iOS Development

Camera and Media Picker

Readings

Apple’s UIImagePickerController Class Reference

Framework Classes

UIImage

UIImageView

UIImagePickerController

UIPopoverController

The Image Picker

Instantiate and add to view hierarchy:

`UIImagePickerController`

Image Picker Types:

`UIImagePickerController.sourceType`**`Camera`**
`UIImagePickerController.sourceType`**`PhotoLibrary`**
`UIImagePickerController.sourceType`**`SavedPhotosAlbum`**

Instantiating the Image Picker

```
let imagePickerController =
UIImagePickerController();
```

UIImagePickerController

Demo

UIImagePickerController Source

```
imagePickerController.sourceType =
UIImagePickerControllerSourceTypeCamera;
```

Available as of iOS 7:

```
UIImagePickerControllerSourceTypeCamera
UIImagePickerControllerSourceTypePhotoLibrary
UIImagePickerControllerSourceTypeSavedPhotosAlbu
m
```

Checking if Camera Exists

```
if( UIImagePickerController.isSourceTypeAvailable(
      UIImagePickerControllerSourceTypeCamera) )
{
    imagePicker.sourceType =
        UIImagePickerControllerSourceTypeCamera
}
else{
    imagePicker.sourceType =
       UIImagePickerControllerSourceTypePhotoLibrary
}
```

UIImagePickerSourceType

Demo

UIImagePickerController Editing

```
imagePickerController.allowsEditing = true //
or false
```

UIImagePickerController Editing

Demo

UIImagePickerController Delegate

```
UIImagePickerControllerDelegate,
UINavigationControllerDelegate // delegates


imagePickerController.delegate = self as
UIImagePickerControllerDelegate &
UINavigationControllerDelegate
```

Working with the Image

```
func imagePickerController(picker: UIImagePickerController,
didFinishPickingMediaWithInfo info: [NSObject : AnyObject])
{
    let userChosenImage:UIImage =
info[UIImagePickerControllerOriginalImage] as! UIImage

    self.imageView.image = userChosenImage

    picker.dismiss(animated: true, completion: nil)
}
```

Image Info Constants

```
UIImagePickerControllerMediaType;
UIImagePickerControllerOriginalImage;
UIImagePickerControllerEditedImage;
UIImagePickerControllerCropRect;
UIImagePickerControllerMediaURL;
UIImagePickerControllerReferenceURL;
UIImagePickerControllerMediaMetadata;
```

Exif Data?

Check out:

AssetsLibrary.framework

ImageIO.framework

UIImagePickerController Captured Image

Demo

Smile Detection and UPC/QRCode Scanners

Smile Detection on an Image

```
@import CoreImage;
CIDetector *smileDetector = [CIDetector detectorOfType:CIDetectorTypeFace
                                    context:context
                                    options:@{CIDetectorTracking: @YES,
                                              CIDetectorAccuracy: CIDetectorAccuracyLow}];

NSArray *features = [smileDetector featuresInImage:image options:@{CIDetectorSmile:@YES}];

if (([features count] > 0) && (((CIFaceFeature *)features[0]).hasSmile)) {
    UIImageWriteToSavedPhotosAlbum(image, self, @selector(didFinishWritingImage), features);
} else {
    self.label.text = @"Say Cheese!"
}
```

Scan UPCs, QR codes, and barcodes

```
AVCaptureSession *session = [[AVCaptureSession alloc] init];
AVCaptureDevice *device = [AVCaptureDevice defaultDeviceWithMediaType:AVMediaTypeVideo];
NSError *error = nil;
AVCaptureDeviceInput *input = [AVCaptureDeviceInput deviceInputWithDevice:device error:&error];
if (input) {
    [session addInput:input];
} else {
    NSLog(@"Error: %@", error);
    return;
}
AVCaptureMetadataOutput *output = [[AVCaptureMetadataOutput alloc] init];
[output setMetadataObjectsDelegate:self queue:dispatch_get_main_queue()];
[session addOutput:output];
[output setMetadataObjectTypes:@[AVMetadataObjectTypeQRCode]];
[session startRunning];
```

AVCaptureMetadataOutputObjectsDelegate

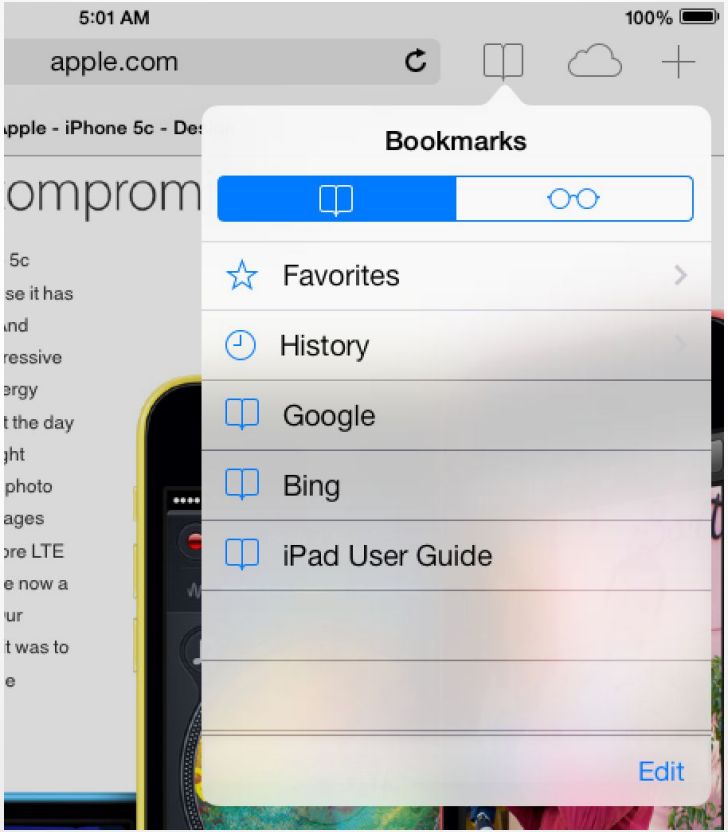
```
- (void)captureOutput:(AVCaptureOutput *)captureOutput
didOutputMetadataObjects:(NSArray *)metadataObjects
       fromConnection:(AVCaptureConnection *)connection
{
    NSString *QRCode = nil;
    for (AVMetadataObject *metadata in metadataObjects) {
        if ([metadata.type isEqualToString:AVMetadataObjectTypeQRCode]) {
            QRCode = [(AVMetadataMachineReadableCodeObject *)metadata stringValue];
            break;
        }
    }

    NSLog(@"QR Code: %@", QRCode);
}
```

AVMetadataObject Types

```
AVMetadataObjectTypeUPCECode
AVMetadataObjectTypeCode39Code
AVMetadataObjectTypeCode39Mod43Code
AVMetadataObjectTypeEAN13Code
AVMetadataObjectTypeEAN8Code
AVMetadataObjectTypeCode93Code
AVMetadataObjectTypeCode128Code
AVMetadataObjectTypePDF417Code
AVMetadataObjectTypeQRCode
AVMetadataObjectTypeAztecCode
```

Popovers

iOS Popovers



Instantiating the UIPopoverController

```
UIPopoverController *popoverController = [[UIPopoverController alloc] initWithContentViewController: imagePickerController];
CGRect popoverFrame = currentButton.frame;
popoverFrame.origin.x += photoThumbnailsView.frame.origin.x;
popoverFrame.origin.y += photoThumbnailsView.frame.origin.y;
CGRect translatedRect = [self.view convertRect: popoverFrame fromView: scrollView ];
[popoverController presentPopoverFromRect: translatedRect inView: self.view permittedArrowDirections:UIPopoverArrowDirectionRight animated: YES];
[popoverController setDelegate: self];
```

UIPopoverController Delegate

```
<UIPopoverControllerDelegate>

-(void)popoverControllerDidDismissPopover:
(UIPopoverController *)popoverController {...}
```

UIPopoverController

Demo