

IOS TIPS: VIEWING PDFS IN LESS THAN 30 SECONDS

By using the Quick Look Framework

If you need in your app to present a document like a PDF, a bunch of images, or even an Microsoft Office or iWork document, you can use the [Quick Look Framework](#) that ships with iOS.

You start by importing the `QuickLook` framework into your project file. You then create a `QLPreviewController`, create a `QLPreviewControllerDataSource`, hand in as many `QLPreviewItem` instances to the data source as you need, and then present the controller. The only wrinkle here is that the `QLPreviewItem` and `QLPreviewControllerDataSource` objects are just protocols, you have to create concrete types in order to use them.

Before we start the clock, create a new single view application in Xcode, and have a PDF document to hand that you would like to open. In this app we're going open a PDF from the application bundle, but note that this approach works just as well with a loose file or an internet based URL. With your project and PDF resource in place, we're ready to start the clock:

- In Xcode, in **ViewController.m** file, below the first `#import` statement, add the following import statement:

```
@import QuickLook;
```

- Below the `@import QuickLook;` declaration, add the following code to create a `QLPreviewItem` implementation:

```
/*
 * Quicklook Preview Item
 */
@interface PreviewItem : NSObject <QLPreviewItem>
@property(readonly, nullable, nonatomic) NSURL * previewItem-
URL;
@property(readonly, nullable, nonatomic) NSString * previewItem-
Title;
```

```

@end
@implementation PreviewItem
- (instancetype)initWithPreviewURL:(NSURL *)docURL WithTitle:(NSString
*)title {
    self = [super init];
    if (self) {
        _previewItemURL = [docURL copy];
        _previewItemTitle = [title copy];
    }
    return self;
}
@end
/***/

```

This `QLPreviewItem` implementation provides the mandatory `previewItemURL` property, but also includes the optional `previewItemTitle` property, with a convenience *init* method to set those properties.

- Next, below the `PreviewItem` implementation add the following code that implements the `QLPreviewControllerDataSource` protocol:

```

/*
 * QuickLook Datasource for rendering PDF docs
 */
@interface PDFDataSource : NSObject <QLPreviewControllerDataSource>
@property (strong, nonatomic) PreviewItem *item;
@end
@implementation PDFDataSource
- (instancetype)initWithPreviewItem:(PreviewItem *)item {
    self = [super init];
    if (self) {
        _item = item;
    }
    return self;
}
- (NSInteger)numberOfPreviewItemsInPreviewController:(QLPreviewCon-
troller *)controller {
    return 1;
}

```

```

}
- (id<QLPreviewItem>)previewController:(QLPreviewController *)con-
troller previewItemAtIndex:(NSInteger)index {
    return self.item;
}
@end
/**/

```

Only the two methods are defined in the `QLPreviewControllerDataSource` protocol, the remaining implementation provide storage for the `PreviewItem` instance and a convenience init method to create the data source.

- In the interface declaration for the `ViewController`, add the following property declaration:

```

@property (strong, nonatomic) PDFDataSource *pdfDatasource;

```

You'll need to retain an instance of the `QLPreviewControllerDataSource` for when you present the `QLPreviewController`.

- Below the `viewDidLoad` method, add the following code, adjusting the name of the file you wish to load from within the application bundle, here mine is called *article.pdf*:

```

- (void)viewDidAppear:(BOOL)animated {
    [super viewDidAppear:animated];

    /*
     * get the path to the pdf resource.
     */

    NSString *path = [[NSBundle mainBundle] pathForResource:@"art-
icle" ofType:@"pdf"];
    NSURL *docURL = [NSURL fileURLWithPath:path];

    /*

```

```

    *   create the Quicklook controller.
    */

    QLPreviewController *qlController = [[QLPreviewController alloc]
init];

    PreviewItem *item = [[PreviewItem alloc] initWithPreviewURL:docURL
WithTitle:@"Article"];
    self.pdfDatasource = [[PDFDataSource alloc] initWithPreview-
Item:item];
    qlController.dataSource = self.pdfDatasource;

    /*
    *   present the document.
    */

    [self presentViewController:qlController animated:YES comple-
tion:nil];
}

```

With this code you get the path to the file from the default bundle, create a `NSURL` pointing to the file in the bundle; you then create a `QLPreviewController`, a new `PreviewItem` with the URL and title "Article". Next you create a new data source, handing it the preview item, and then finally, present the QuickLook controller.

- Stop the clock!

I have created a [sample project](#) that you can use if you want to see this approach in action.