

# mTouch-PDFReader

mTouch-PDFReader is a Xamarin iOS library to display PDF documents on iPad and iPhone. With this library you can to develop complete applications for displaying PDF documents, such as: e-book readers, e-magazines, document browsers, etc. The library is written on Xamarin iOS with use of Cocoa Touch classes for reading and displaying the PDF documents, without the use of third-party libraries. mTouch-PDFReader can be easily integrated to your project as a library or Xamarin component.

The library tested on the latest iOS version 8.1. The current library version is 3.0.0.

## Features

- iPad and iPhone support
- Two transition styles:
  - Curl
  - Scroll
- Two navigation orientations:
  - Horizontal
  - Vertical
- Autofit page width and height:
  - Based on the device's size
  - After rotating the device
- Page scaling
- Bookmarks
- Notes
- Pages thumbnails
- Navigation controls
- Options view
- No third-party code, only C# and Xamarin iOS

## How to integrate

To use the library in your project, you should to do a few simple steps:

1. Add a reference to mTouch-PDFReader library in your project. Or install mThouch-PDFReader component from Xamarin Component Store.
2. Register global managers in *AppDelegate* class:

```
public partial class AppDelegate : UIApplicationDelegate
{
    public AppDelegate()
    {
        var builder = new ContainerBuilder();
        builder.RegisterType().As().SingleInstance();
        builder.RegisterType().As().SingleInstance();
    }
}
```

```

        builder.RegisterType().As().SingleInstance();
        MgrAccessor.Initialize(builder);
    }
}

```

3. To open PDF document, use the following code:

```

// Set unique document id, document name (will be used as title) and document
full file path
var docId = 1;
var docName = "My book";
var docPath =
Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Personal),
"MyBook.pdf");

// Create Document View controller and push it in Navigation controller, for
example
var docVC = new DocumentVC();
NavigationController.PushViewController(docVC, true);

```

That's it! The PDF document will be opened with default settings:)

## How to use Bookmarks and Note manager

The library contains the simple versions of Bookmarks and Note manager. In demo application the all data stored in the memory, and after closing application is losing. But you can inherit these managers and store data in a file or Sqlite db. To do it, you should inherit

DocumentBookmarksManager (for bookmarks) and DocumentNoteManager (for notes) managers:

```

public class MyDocumentBookmarksManager : DocumentBookmarksManager
{
    public override List GetAllForDocument(int docId)
    {
        // This method must return all bookmarks objects for the document with id=docId
    }
    public override void Save(DocumentBookmark bookmark)
    {
        // This method for saving bookmark object
    }
    public override void Delete(int bookmarkId)
    {
        // This method for deleting bookmark by id
    }
}

public class MyDocumentNoteManager : DocumentNoteManager
{
    public override DocumentNote Load(int docId)

```

```

{
    // This method for loading note for document, by docId
}
public override void Save(DocumentNote note)
{
    // This method for saving note object
}
}

```

## How to use Settings

To use Settings View Controller, just create a SettingsVC object and add it as subview to your View Controller:

```

var settingsVC = new SettingsTableVC();
settingsVC.View.Frame = View.Bounds;
View.AddSubview(settingsVC.View);

```

## History of changes

- 2014-12-03 (v3.0.0)
  - Projects upgraded to support iOS 8.1.
  - Fixed displaying bookmarks, note, goto, thumbs views on iPhone. Updated icons for buttons in these views.
  - Refactored and improved Thumbs view.
  - Fixed settings view layout.
  - Changed toolbars icons.
  - Custom toolbar replaced to standard UIToolbar. Fixed bug with displaying toolbars on iPhones.
  - Fixed localization issues.
  - Handmade dependency resolver replaced to cool library - Autofac.
  - Fixed toolbar and statusbar sizes and positions.
- 2013-04-20 (v2.1.0)
  - Added autofit page width and height.
  - Updated toolbar icons.
  - Refactored buttons creation logic.
  - Deleted a few settings.
- 2012-11-10 (v2.0.0)
  - Major refactoring of the library core logic.
  - Added UIPageViewController, instead self implementation.
  - Added Resource manager.
  - Refactored Note manager.
  - Refactored Bookmarks manager.
  - Changed the navigation buttons logic.

- Fixed some major bugs.
- 2012-07-15 (v1.0.4)
  - Fixed bug with shadow copy of the page (reproduced on iPad 3).
- 2012-04-27 (v1.0.3)
  - Fixed some bugs.
- 2012-03-09 (v1.0.2)
  - Added page thumbnails.
  - Added new settings for the library customization.
- 2012-02-22 (v1.0.1)
  - Added localization support for the library string resources.
  - Translated the library interface to Russian and English.
- 2012-02-12 (v1.0.0)
  - The first public version of the library!
- 2012-01-28 (v0.9.2)
  - Added Settings view.
  - Added Options manager.
  - Implemented detailed page view.
  - Finished Note and Bookmarks managers.
  - Optimized the library core.
  - Fixed some bugs.
- 2012-01-15 (v0.9.1)
  - Finished the library core.

## Questions and issues

If you have any questions or issues, please write me at: [support@mtouch-pdfreader.com](mailto:support@mtouch-pdfreader.com) or create issue on github.

## License

This code available under the MIT License. See License.md for details.

## Authors

Alexander Mac ([amatsibarov@gmail.com](mailto:amatsibarov@gmail.com))