User Manual for iOS Rich Text Editor

Version 1.0.0

This document will provide the full explanation of the functionality for the iOS Rich Text Editor and its integration with other iOS apps.

It is a powerful, unique featured and simple to use rich text editor wherein the users can perform all the formattings (such as bold, italics, font size, font family, font and background color, alignment etc.). It supports trackpad (a feature similar to mac's trackpad), native find and replace as well as web search options.

This editor supports the below features:

- Bold
- Italic
- Underline
- Align left
- Align center
- Align right
- Font color
- Background color
- Search on web
- Trackpad(move finger on keyboard to move cursor on textview i.e. similar to trackpad in mac's)
- Keyboard down
- Font size and family (In paid version)
- Native search and replace (In paid version)

Requirements: iOS 7 or later.

Below are some of the screen-shots of this app:



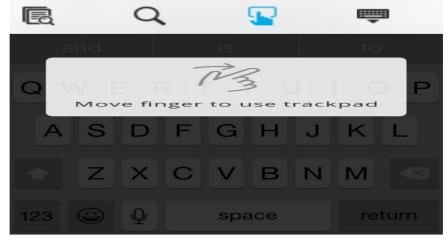
Find and Replace

●●●○○ IDEA 🕏

medma medma demo demo demo demo demo demo demo custom text editor demo medma medma demo demo demo demo

demo demo demo custom text editor demo custom text editor demo custom text editor demo custom textmo demo demo demo demo demo demo demo custom

text editor demo medma medma demo demo demo demo demo demo demo demo custom text editor demo custom text demo custom textmedma medma demo demo demo demo demo demo demo demo custom text



Trackpad control

●0000 IDEA 🕏 4:43 pm 93%

medma medma demo demo demo demo



Different Editor Controls

Integration with other iOS apps

This editor can be integrated with other iOS apps very easily. Below is the process for that:

####HOW TO ADD IN YOUR PROJECT----

- 1. Copy CustomTextView folder to your app.
- 2. Import #import "CustomTextView.h" in your view controller.
- $3. \ Call \ the \ constructor. \ eg. \ Custom TextView \ *custom TextView = [[Custom TextView alloc] in it With View: with View Controller: self];$

Here the view is in which CustomTextView is to be set.

- 4. Add CustomTextView to your view eg. [view addSubview:customTextView];
- 5. Set customTextView's delegates to self

Eg. customTextView.customDelegate=self;

####HOW TO SAVE WHEN EDITING IS DONE--

1. Call the function "saveWithFileName".

Eg. [customTextView saveWithFileName:@"user_1"];

// filename should be unique.

Function will give callback (below mentioned delegate calls).

- 2. Register for these delegate-
- -(void)attributedTextFileNotSavedWithError;
- -(void)fileSaved;

####HOW TO SHOW EDITED TEXT--

1. Call the function "setAttributedTextwithfileName" to set attributed text on CustomCustomTextView.

Eg. [customTextView setAttributedTextwithfileName@"user_1";] //filename same while saving

####DELEGATES--

1. -(void)attributedTextFileNotSavedWithError:(NSError *)error;

// if attributed text file is not saved to the local path.

- 2. -(void)errorWhileReadingAttributedTextFile:(NSError *)error;
- // error while converting content of attributed text file to nsdata.
- 3. -(void)fileDoesNotExistAtPath:(NSString *)path;
- // attributed text file not exist at path.
- 4. -(void)fileSaved;
- // attributed text file saved successfully.