While browsing through some of the tips and tricks online about the iOS app development, I came through a couple of really cool tools.

How often has it happened that our App is bundled with loads of images and adding the images to Objects dynamically through programs turns out to be a bit of pain in our head? Although the Xcode’s internal Auto-Completion feature has been integrated really well, but it doesn’t have any trick through which we can get auto-complete for file names.

I came across a really interesting Plug-in which will enable Auto-completion of file names in any of your project and its really easy to integrate the tool in your system.

All you need to do is download the plug-in from GITHUB. The source code is available at:

<https://github.com/ksuther/KSImageNamed-Xcode>

1. Extract the contents of the downloaded ZIP file and run the project “KSImageNamed.xcodeproj” on your MAC. A plugin will be installed in XCode.
2. Restart your XCode and you are ready to use the feature.

To verify whether the plugin has been installed on your system or not, open terminal and go to ~/Library/Application Support/Developer/Shared/Xcode/Plug-ins. See the list of files and you should be able to find: “KSImageNamed.xcplugin”.

Have you ever wondered, while entering an email you could get an auto-completion of your domain name? Or if you want a text field to input certain values that are specific to your App. So here’s an interesting class which lets you manage those text fields and show Auto-complete of certain fields (like domain name while entering an email). Again damn easy to implement, all you need to do is download the project in your system from:

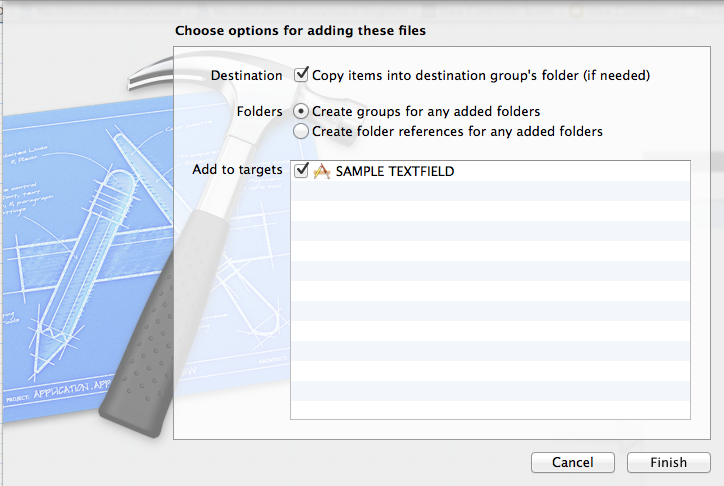
<https://github.com/hoteltonight/HTAutocompleteTextField?utm_source=iOS+Dev+Weekly&utm_campaign=49af20333c-iOS_Dev_Weekly_Issue_79&utm_medium=email>.

Although a quick-start guide is mentioned on the GITHUB page itself, here’s a brief guide to help you integrate the project fast:

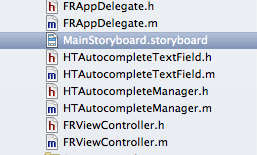
All you need to do is locate the following files in the recently downloaded folder

1. HTAutocompleteTextField.h
2. HTAutocompleteTextField.m
3. HTAutocompleteManager.h
4. HTAutocompleteManager.m

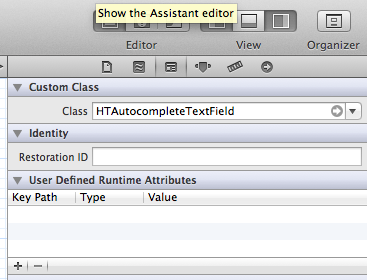
Make sure that you select the proper options from the copy menu:



Now your File Navigator on the left of your XCode should resemble this:



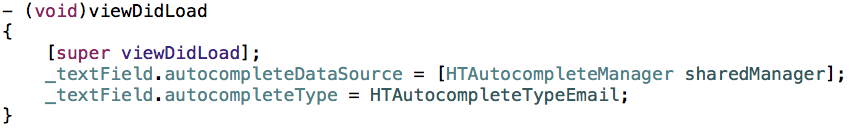
You can add the Text field programmatically too but also its very easy to simply drag and drop a UITextField from the attributes inspector onto your view. Since HTAutocompleteTextField is a subclass of UITextField, all the original features of a UITextField are still available. Now to make your dragged Textfield refer to this new class that you have added, you should select your TextField and from the Attributes Inspector set the class from default UITextField to HTAutocompleteTextField.



Also make sure to add the following import statements at the top of your interface file:



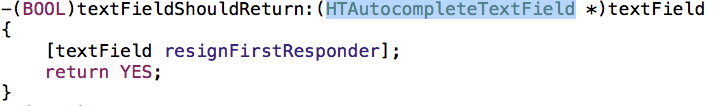
In your implementation file add the following two lines to your viewDidLoad method declaration:



Now after creating a property of your created Textfield you should be able to get the autocompletion of text on your textField.

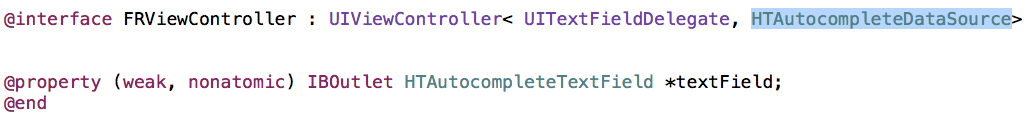
But hand on, you still don’t have a way to finally get the suggested text in your text field. Just conforms to the UITextFieldDelegate in your interface file.

In your implementation file override the UITextFieldShouldReturn method that you have been doing so far to dismiss the keyboard while the return key is pressed. But here, change the type of argument accepted by the method from UITextField \* to HTAutocompleteTextField \*. Now your method should look like this:



Go ahead and run your project, you should be able to get auto-complete of domain names while entering emails.

If you wish, for a certain textField in your App you wish to provide some specific auto-completion strings, all you need to do is conform to the protocol HTAutocompleteDataSource in your interface file like shown below:



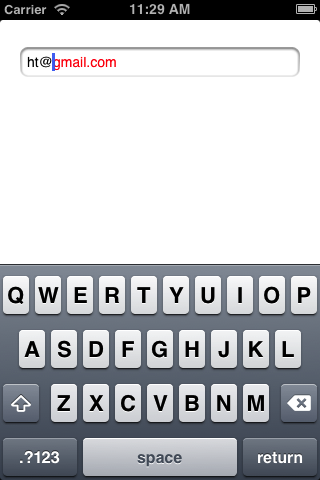
And then in your implementation file you need to over-ride a compulsory method for this data source which is:

- (NSString \*)textField:(HTAutocompleteTextField \*)textField completionForPrefix:(NSString \*)prefix ignoreCase:(BOOL)ignoreCase;

By default the font color of the auto-complete text is set to gray, you can even modify that color to suit your requrements like:

textField.autocompleteLabel.textColor = [UIColor redColor];

So now while auto-completing your textField’s text would appear as follows:



Hope everyone got something new and interesting to learn from this post.