## Explanation for the project

1. Requirement

a) Load data from API.

Solution : Use AFNetWorking library to do *Get* request from server and parse the Json data locally for displaying.

b) Expand cell height with longer text.

Solution : Calculate the text height given an fixed frame (the UILabel size). To customized the height of the cell, I need to return the calculated height in the delegate function of UITableViewDelegate.

c) Swipe each cell to show an edit button and click to edit.

Solution : Create an edit action in "*editActionsForRow*" delegate function. To refresh the table view, I need to invoke a callback function when the updating is done.

d) Swipe each cell to show delete button and delete the item from the list.

Solution : Create an delete action in "*editActionsForRow*" delegate function. If clicked, delete the data from the list and reload the table view.

e) Date in local timezone.

Solution : Create an DateFormatter object and configure it with TimeZone.current to convert it to local time zone.

f) Design Pattern.

Solution: MVC (Model-View-Controller). The structure of the project is shown in the folder. Model folder contains data models. NetworkService folder contains request service. UI folder contains two sub-folder : view and controller. Extension folder contains helper class for convenience.

g) Keyboard Avoiding Problem.

Solution: When editing, the text view will be potentially covered by the keyboard. To handle this, I import the "TPKeyboardAvoiding" package which has a nice solution for this problem.

h) Adaptation for different size screen.

Solution : The editing page has many subviews. To avoid that the content is covered, I use scroll to display the elements to ensure the app can be used in full-size devices properly.

i) Unit Test Case.

Solution : Because of the time constraint, I just fulfill the unit test cases for “FeedItem” class.