

Social Searcher iOS Application

1. Project Name: SocialSearcher

2. About Project

- a. You need to open 'SocialSearcher.xcworkspace' file to compile and run the application
- b. All open source projects are imported using 'cocoa pods' commands.
- c. I created git repository from the very first time I started write the application
- d. I have own coding convention and style. This project is follows my coding convention and style. I manage a document that describing those coding convention and style at the github (<https://github.com/op1000/CodingStyleGuide>). Although this document is written in Korean, I will prepare English version later soon.
- e. This project uses prefix 'NC' on all classes name which is followed from my initial name 'NCJung'.
- f. I used new tech-knowledges introduced at WWDC 2015 such as StackView using autolayout in storyboard, UI automation using xCode not instrument.
- g. Some features are not supported under iOS 9.0. Therefore I set the target deploy setting to iOS 9.0. So this project produces compile error on under iOS9.0.
- h. By using autolayout feature I can manage all iOS mobile devices to fit adequately, including portrait mode and landscape mode.
- i. Memory warning detecting and adequate codes is also implemented at all `didReceiveMemoryWarning` function.

3. Model Classes

- a. `NCImageCachingManager`: Image caching class that saves image data to mobile disk.
- b. `NCURLManager`: Manages URL manipulation, create/delete folders.
- c. `NCYoutubeDataManager`: Communicate with Youtube API and save results to `NSDictionary`. Uses delegate to communicate with user classes like UI.
- d. `NCYoutubeDataContainer`: Saves Youtube API search results. This class is Singleton class.
- e. `NCUtilManager`: Aggregated simple functions as static.

4. Youtube API calls

- a. Recommended channel list
 - i. Calls `https://www.googleapis.com/youtube/v3/guideCategories`: fetches 'ChannelID' only with the call.
 - ii. Then call `https://www.googleapis.com/youtube/v3/playlists`: fetches channel list from the results. For fetching next page, I use 'nextPageToken' field. The 'nextPageToken' field value stored in `NSDictionary` defined at `NCYoutubeDataContainer` class.
- b. Video list
 - i. Calls `https://www.googleapis.com/youtube/v3/playlists`: fetches channel's video list. For fetching next page, I use 'nextPageToken' field. The 'nextPageToken' field value stored in `NSDictionary` defined at `NCYoutubeDataContainer` class.
 - ii. Each cell calls `https://www.googleapis.com/youtube/v3/videos`: fetches duration info from the result.
- c. Youtube search
 - i. Calls `https://www.googleapis.com/youtube/v3/search`: fetches searched list from the results.

5. Unit testing classes

- a. `NCYoutubeDataManagerTests`: Tests `NCYoutubeDataManager`

- d. NCMainviewControllerTests: Tests main UI, landscape, portrait, play video. This UI test mechanism is pretty new and also experimental.
- c. NCSearchViewControllerTests: Tests search UI, landscape, portrait, play video. This UI test mechanism is pretty new and also experimental.
- d. NCVideoListViewControllerTests: Tests video list UI, landscape, portrait, play video. This UI test mechanism is pretty new and also experimental.

6. UI flow

- a. You can check UI flow at 'Main.storyboard'.

