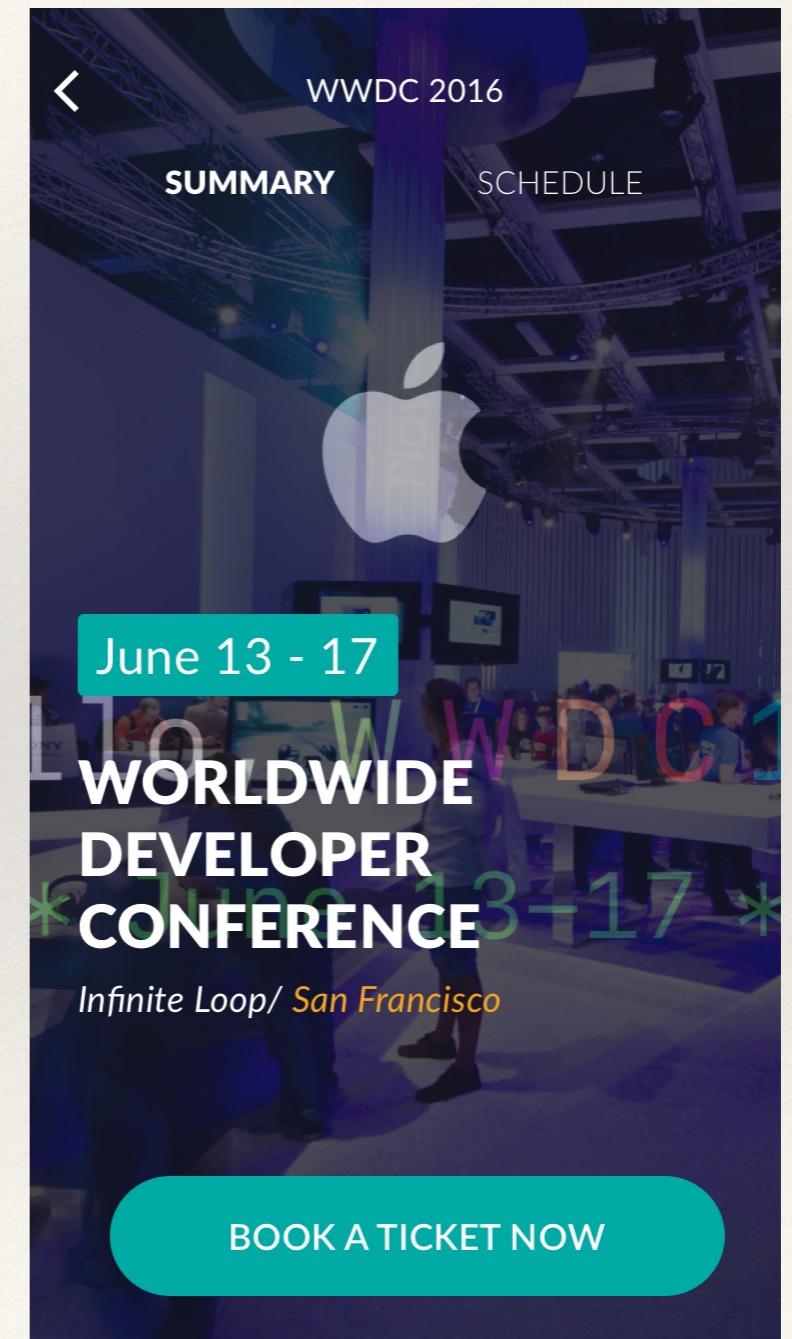
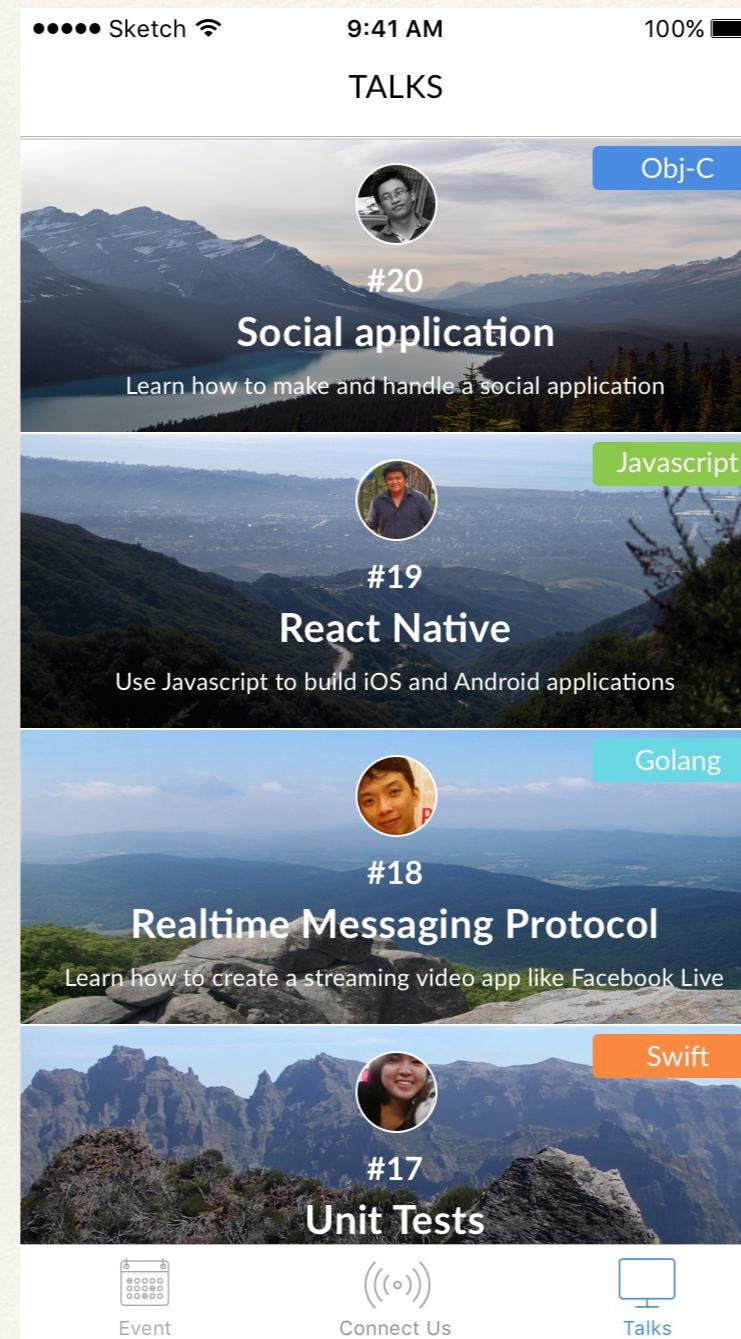
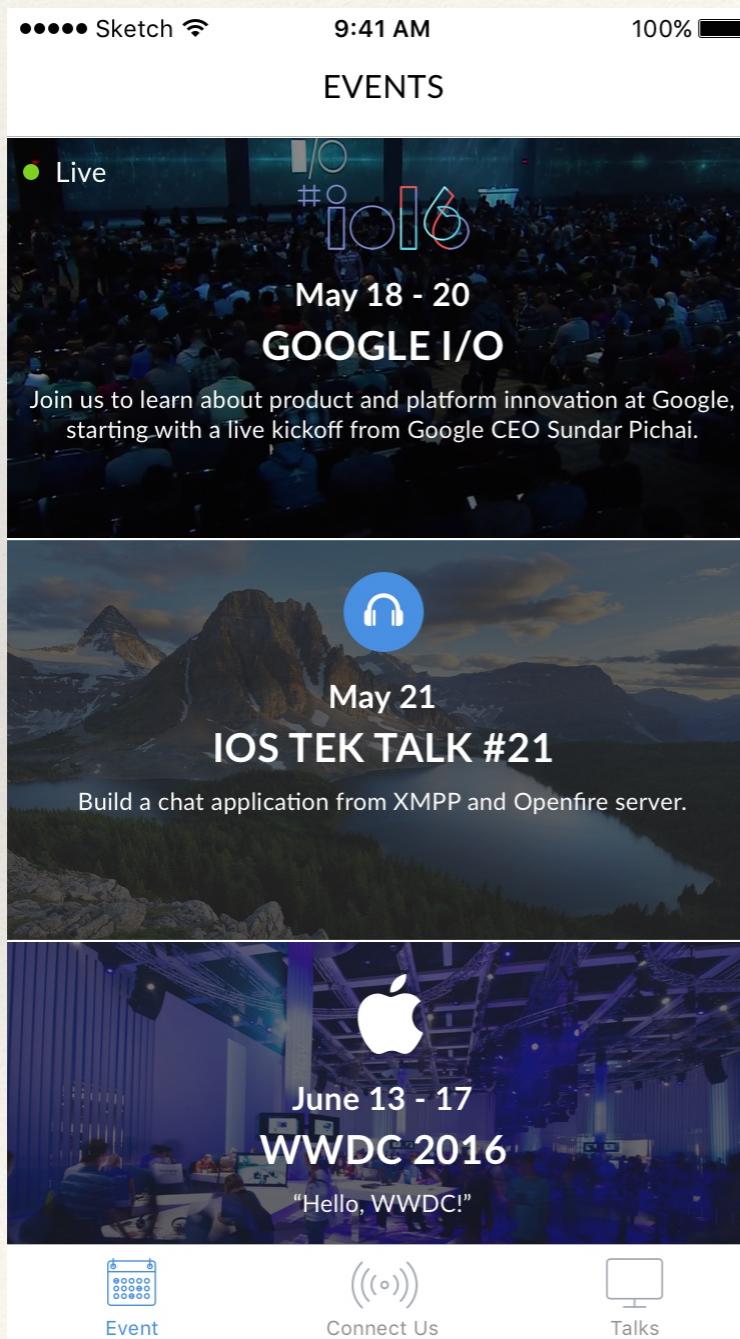
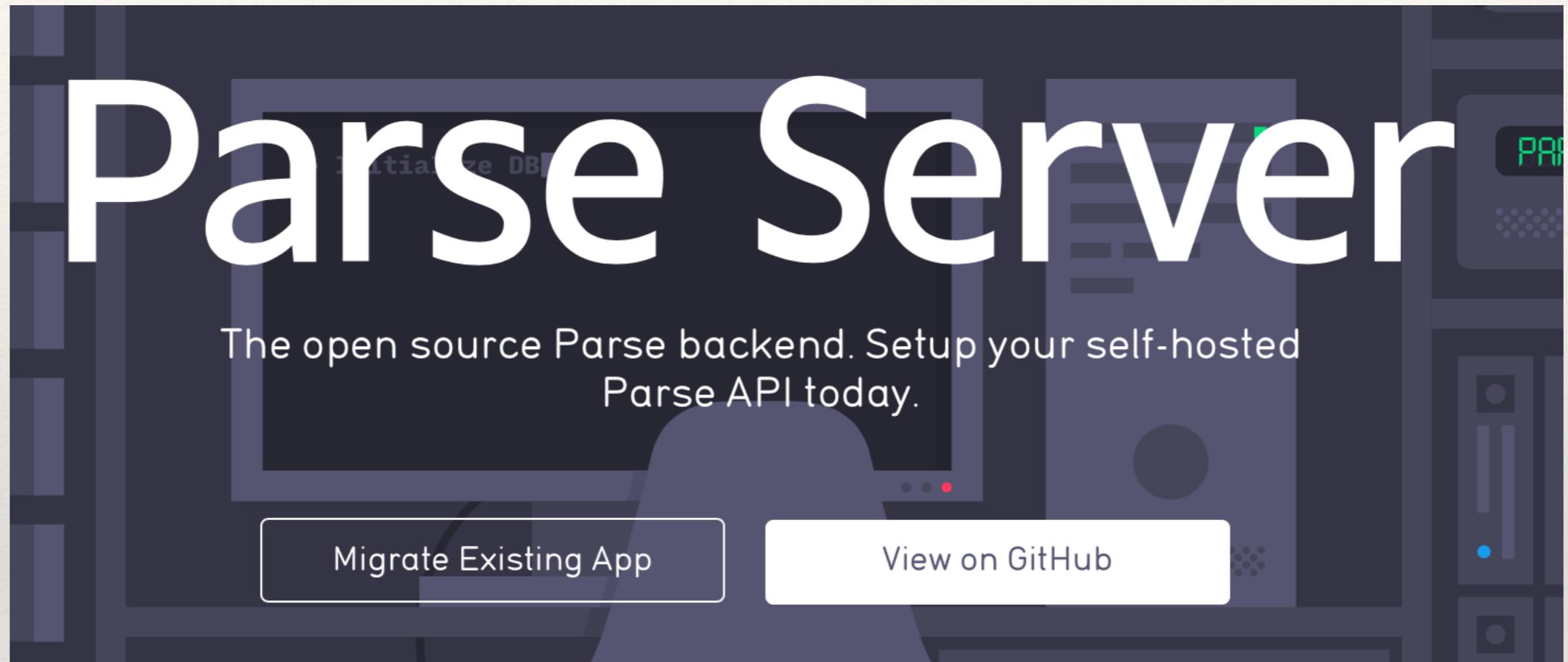


Tek Talk app

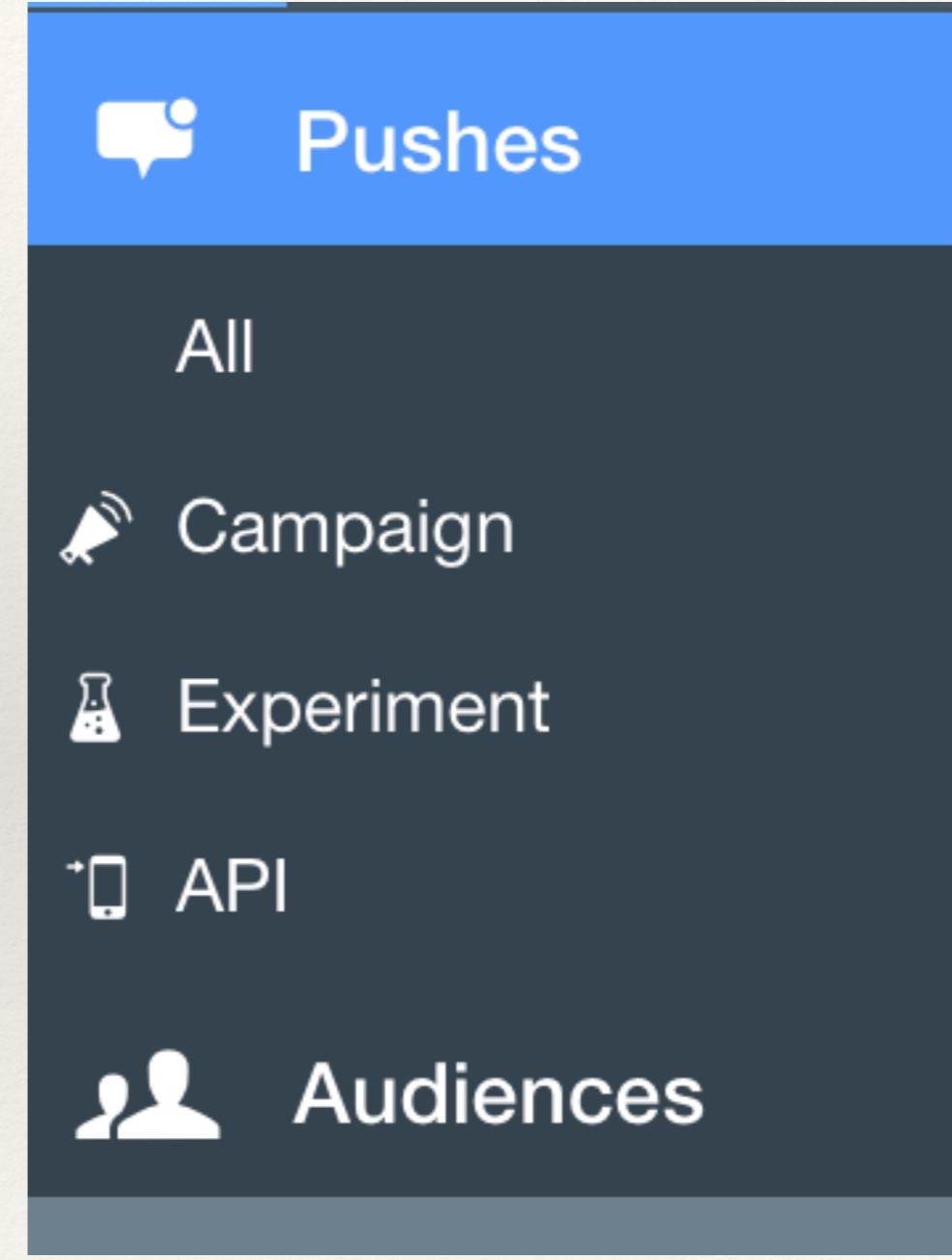
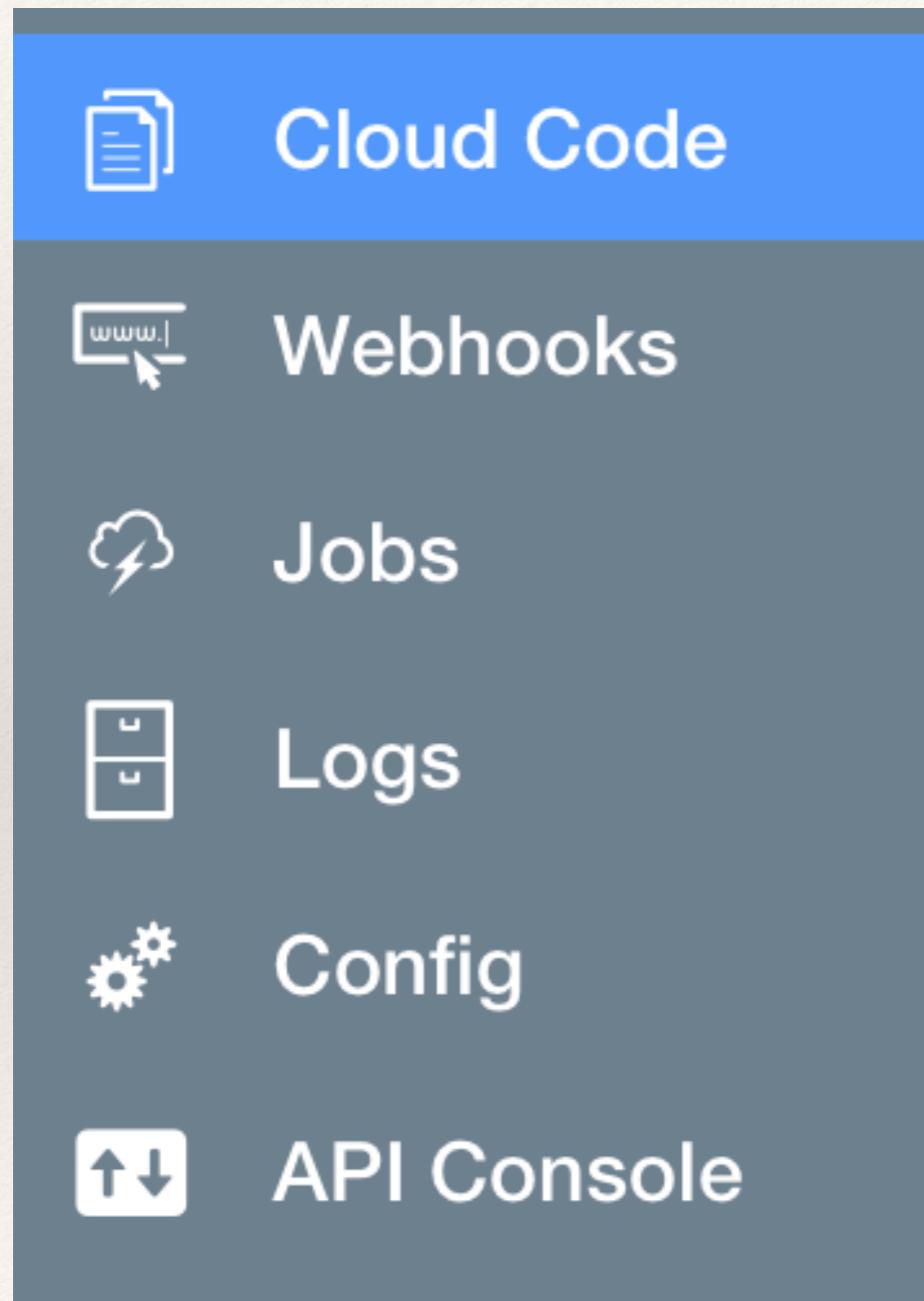


Parse

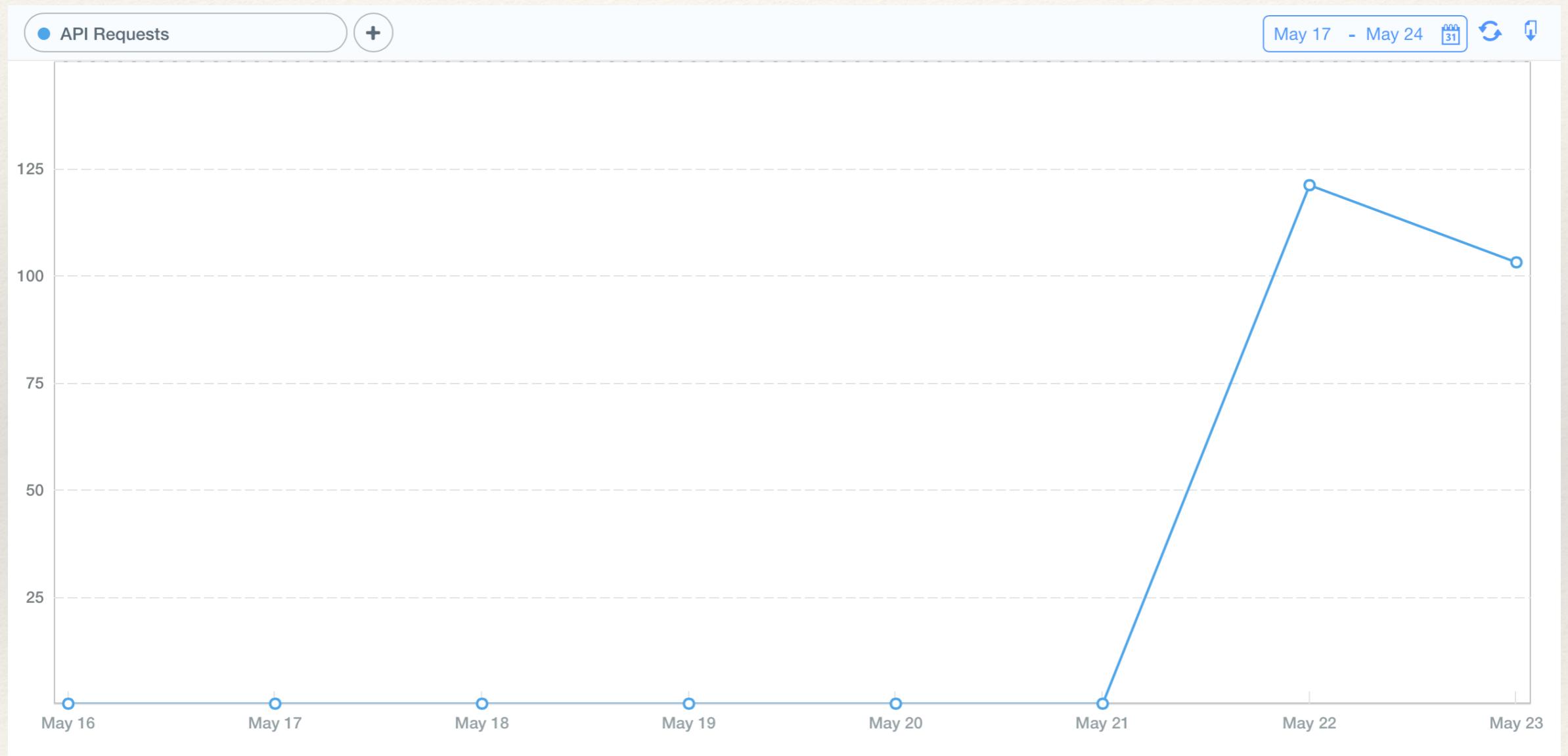


- ❖ Free
- ❖ Build backend-app within 5 mins
- ❖ Don't need to know backend side -> Leave it to Parse
- ❖ -> Focus to your product

Useful tool



Analytic



Database

Data	
🛡 Role	0
👤 User	0
Document	9
Event	23
Feedback	0
Schedule	36
Speaker	8
Tag	4
Video	13
+ Add Class	

- ❖ No-SQL database
- ❖ Easy to setup
- ❖ Easy to query

Function - Function

```
//MARK : - Table View Delegate
func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return self.talksData.count
}

func numberOfSectionsInTableView(tableView: UITableView) -> Int {
    return 1
}
func tableView(tableView: UITableView, heightForRowAtIndexPath indexPath: NSIndexPath) ->
    CGFloat {
    return heightCell
}
func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) ->
    TalkCell {
    let cell = tableView.dequeueReusableCellWithIdentifier("TalkCell") as? TalkCell
    let talkObj = self.talksData[indexPath.row]

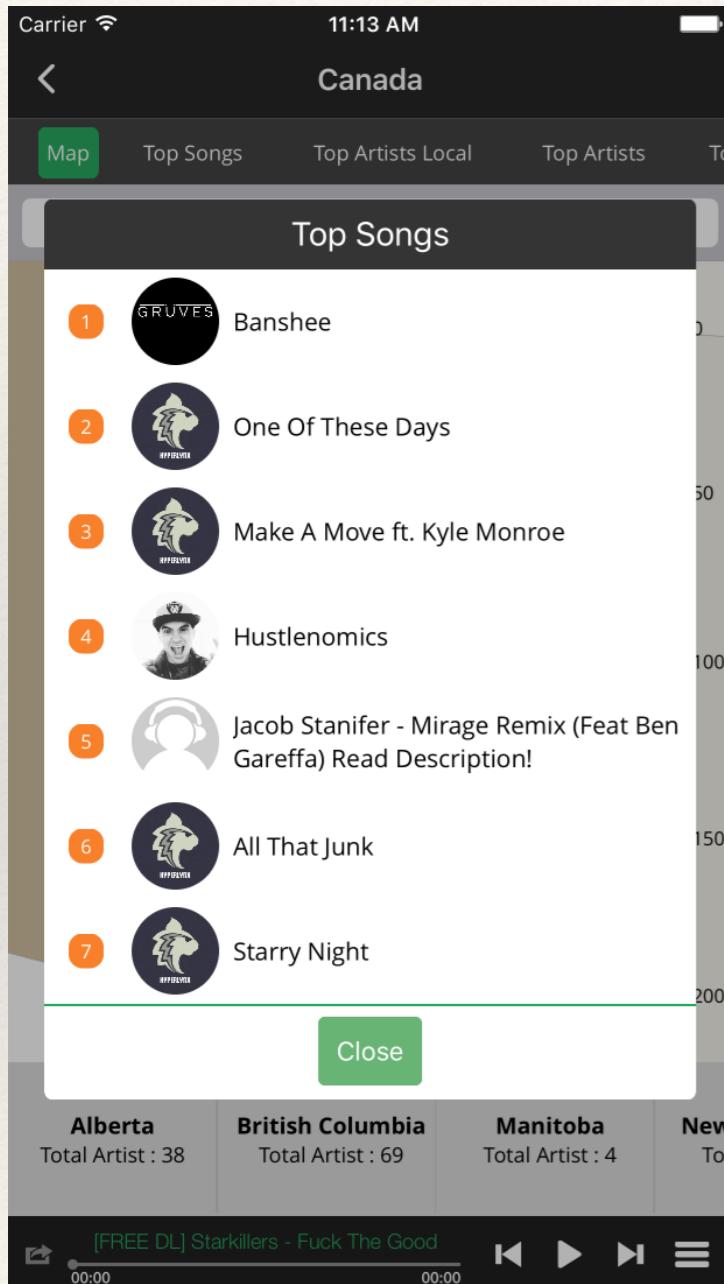
    cell?.imgAvatar?.nk_cancelLoading()
    cell?.imgAvatar?.image = nil;
    cell?.lbDescriptionTekTalk?.text = talkObj["shortDescription"] as? String
    cell?.lbNameTekTalk?.text = talkObj["subject"] as? String
    cell?.bgAvatar?.nk_setImageWith(NSURL(string:talkObj["background"] as! String)!)

    // Speaker
    if let speaker = talkObj["speaker"] as? PFObject {
        cell?.imgAvatar?.nk_setImageWith(NSURL(string:speaker["avatar"] as! String)!)
    }
}

// Video
if let video = talkObj["videos"] as? PFObject {
    print(video)
}

return cell!
}
func tableView(tableView: UITableView, didSelectRowAtIndexPath indexPath: NSIndexPath) {
    tableView.deselectRowAtIndexPath(indexPath, animated: true)
    if let obj = self.talksData[indexPath.row] as? PFObject {
        RoutesManager.pushToTalkDetailWithEventObj(obj)
    }
}
```

Controller - Function



```
    }
-(NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section{
    return self.arrData.count;
}
-(UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath{
    ArtistCell *cell = (ArtistCell*)[self.tableView dequeueReusableCellWithIdentifier:@"ArtistCell" forIndexPath:indexPath];
    cell.delegate = nil;
    switch (self.topDataType) {
        case TopDataTypeSongs:
        {
            cell.feed = self.arrData[indexPath.row];
            UITapGestureRecognizer *tap = [[UITapGestureRecognizer alloc] initWithTarget:self
                                                                           action:@selector(imageTapped:)];
            cell.image.userInteractionEnabled = YES;
            [cell.image addGestureRecognizer:tap];
            cell.image.feed = self.arrData[indexPath.row];
            cell.delegate = self;
        }
        break;
        case TopDataTypeArtist:
        case TopDataTypeArtistLocal:{
            cell.topFanAndArtist = self.arrData[indexPath.row];
        }
        break;
        case TopDataTypeFan:
        {
            cell.topFanAndArtist = self.arrData[indexPath.row];
            cell.lbContent.text = cell.topFanAndArtist.fullName;
        }
        break;
        default:
        break;
    }
    cell.lbCount.text = [NSString stringWithFormat:@"%@",(long)indexPath.row + 1];
    return cell;
}
-(void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath{
    [self gotoDetailContent:self.arrData[indexPath.row]];
    [self.tableView deselectRowAtIndexPath:indexPath animated:YES];
}
-(void)imageTapped:(UITapGestureRecognizer*)gesture{
    [self didTouchedImage:(ImageView*)gesture.view];
}
-(void)didTouchedImage:(ImageView *)image {
    [self showHUDOnWindow];
    [[DataCenter sharedInstance] getProfileById:[NSString stringWithFormat:@"%@",image.feed.user.ID.stringValue] callback:^(ECUser *user, id
```

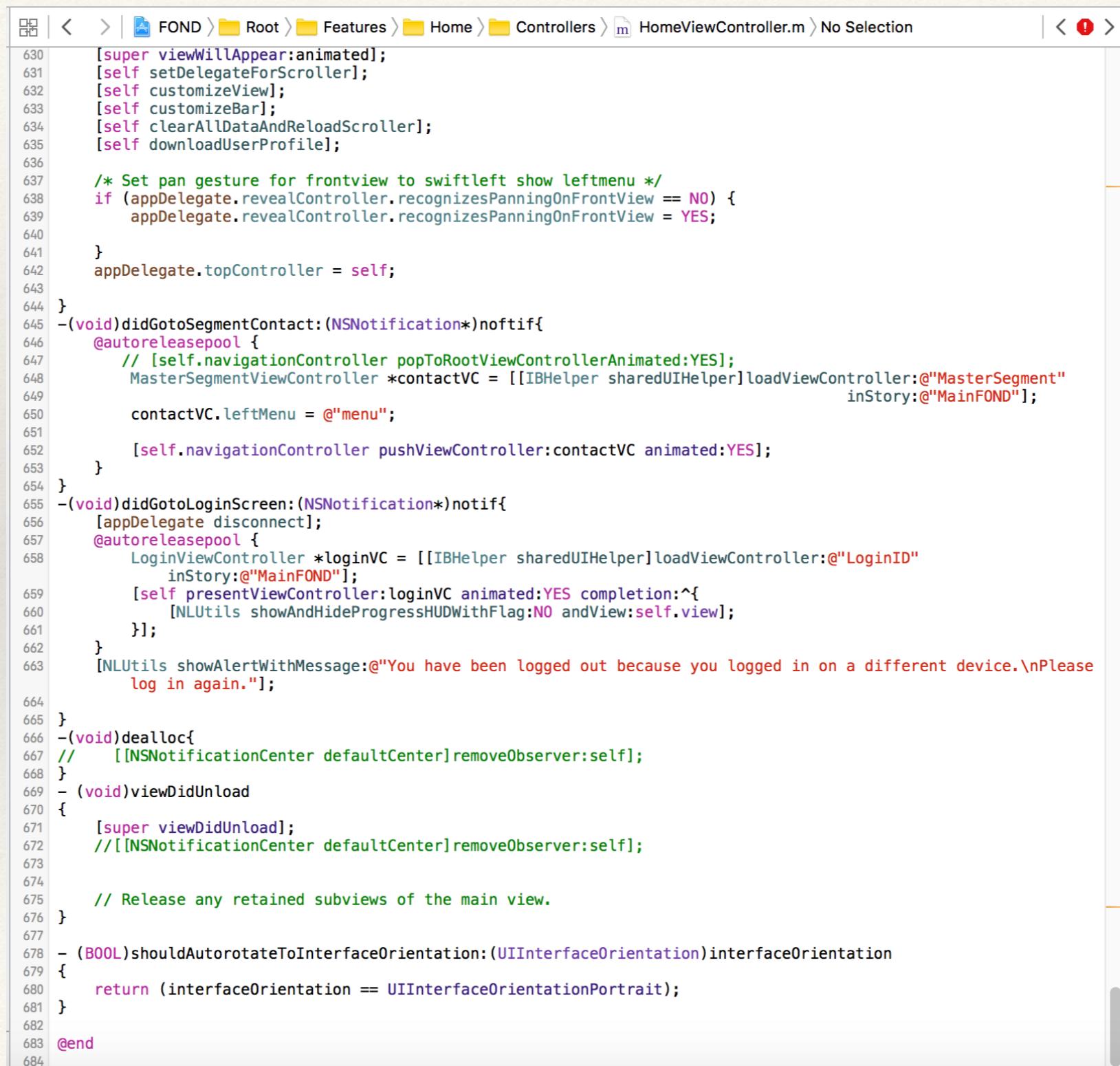
Controller - Manage

```
747     }
748     completion:NUL];
749 }
750 -(void)hideFilterTypeView{
751     [_currentlyActiveSlidingCell slideInContentView];
752     _currentlyActiveSlidingCell = nil;
753     self.btnFilter.selected = NO;
754     [UIView animateWithDuration:0.2
755         delay:0
756         options:UIViewAnimationOptionCurveEaseOut
757         animations:^{
758             self.filTypeView.frame = ViewSetOriginY(self.filTypeView, self.view.bounds.size.height+20);
759         }
760         completion:NUL];
761 }
762 -(void)showSearchBar{
763     [UIView animateWithDuration:0.2
764         delay:0
765         options:UIViewAnimationOptionCurveEaseOut
766         animations:^{
767             self.searchBar.frame = ViewSetOriginY(self.searchBar, 0);
768             self.viewHeader.frame = ViewSetOriginY(self.viewHeader, self.searchBar.frame.origin.y + self.searchBar.frame.size.height);
769             self.tableView.frame = ViewSetOriginY(self.tableView, self.viewHeader.frame.origin.y + self.viewHeader.frame.size.height);
770             self.tableView.frame = ViewSetHeight(self.tableView, self.view.bounds.size.height - self.searchBar.frame.size.height-self.
771                 viewHeader.frame.size.height);
772             self.viewGallery.frame = self.tableView.frame;
773             if (self.btnFilterGallery.selected == YES) {
774                 self.filterImageGallery.frame = ViewSetOriginY(self.filterImageGallery, self.viewHeader.frame.size.height + self.viewHeader
775                     .frame.origin.y-2);
776             }
777         }
778         completion:NUL];
779 }
780 -(void)hideSearchBar{
781     [self.searchBar resignFirstResponder];
782     [UIView animateWithDuration:0.2
783         delay:0
784         options:UIViewAnimationOptionCurveEaseOut
785         animations:^{
786             self.searchBar.frame = ViewSetOriginY(self.searchBar, -44);
787             self.viewHeader.frame = ViewSetOriginY(self.viewHeader, self.searchBar.frame.origin.y + self.searchBar.frame.size.height);
788             self.tableView.frame = ViewSetOriginY(self.tableView, self.viewHeader.frame.origin.y + self.viewHeader.frame.size.height);
789             self.tableView.frame = ViewSetHeight(self.tableView, self.view.bounds.size.height +self.searchButon.frame.size.height-80);
790             self.viewGallery.frame = self.tableView.frame;
791             if (self.btnFilterGallery.selected == YES) {
792                 self.filterImageGallery.frame = ViewSetOriginY(self.filterImageGallery, self.viewHeader.frame.size.height + self.viewHeader
793                     .frame.origin.y-2);
794             }
795         }
796         completion:NUL];
797 }
798 -(void)hideGalleryMenu{
799     [UIView beginAnimations:@"PartialPageCurlEffect" context:nil];
```

Controller - Manage

```
978 - (void)xmppMUC:(XMPPMUC *)sender roomJID:(XMPPJID *)roomJID didReceiveInvitation:(XMPPMessage *)message{
979 //}
980 }
981 - (void)xmppRoom:(XMPPRoom *)sender occupantDidJoin:(XMPPJID *)occupantJID withPresence:(XMPPPresence *)presence{
982 /* handle new user in room and reload message tab*/
983 }
984 - (void)xmppRoster:(XMPPRoster *)sender didReceivePresenceSubscriptionRequest:(XMPPPresence *)presence
985 {
986     XMPPUserCoreDataStorageObject *user = [self.xmppRosterStorage userForJID:[presence from]
987                                         xmppStream:self.xmppStream
988                                         managedObjectContext:[self managedObjectContext_roster]];
989     DDLogVerbose(@"didReceivePresenceSubscriptionRequest from user %@", user.jidStr);
990     [self.xmppRoster acceptPresenceSubscriptionRequestFrom:[presence from] andAddToRoster:YES];
991 }
992 - (void)xmppStream:(XMPPStream *)sender didReceivePresence:(XMPPPresence *)presence
993 {
994     NSXMLElement *x = [presence elementForName:@"x" xmlns:@"http://jabber.org/protocol/muc#user"];
995     for (NSXMLElement *status in [x elementsForName:@"status"])
996     {
997         switch ([status attributeIntValueForName:@"code"])
998         {
999             case 201:{
1000                 NSLog(@"%@", @"OK---");
1001                 NSLog(@"%@", @"False---");
1002             };
1003         }
1004     }
1005 }
1006 }
1007
1008 - (void)xmppStream:(XMPPStream *)sender didReceiveError:(NSError *)error
1009 {
1010     DDLogVerbose(@"%@", THIS_FILE, THIS_METHOD);
1011 }
1012
1013 - (void)xmppStreamDidDisconnect:(XMPPStream *)sender withError:(NSError *)error
1014 {
1015     DDLogVerbose(@"%@", THIS_FILE, THIS_METHOD);
1016
1017     if (!isXmppConnect)
1018     {
1019
1020         blockLoginXMPP(error);
1021         //        blockLoginXMPP = nil;
1022         DDLogError(@"Unable to connect to server. Check xmppStream.hostName");
1023     }
1024 }
1025
1026 //////////////////////////////////////////////////////////////////
1027 #pragma mark XMPPRosterDelegate
1028 //////////////////////////////////////////////////////////////////
1029
1030 - (void)xmppRoster:(XMPPRoster *)sender didReceiveBuddyRequest:(XMPPPresence *)presence
1031 {
1032     DDLogVerbose(@"%@", THIS_FILE, THIS_METHOD);
1033 }
```

Controller - Manage



The screenshot shows the Xcode interface with the file `HomeViewController.m` open. The file path is `FOND > Root > Features > Home > Controllers > HomeViewController.m`. The code is written in Objective-C and handles various notifications and view controller transitions.

```
630 [super viewWillAppear:animated];
631 [self setDelegateForScroller];
632 [self customizeView];
633 [self customizeBar];
634 [self clearAllDataAndReloadScroller];
635 [self downloadUserProfile];
636
637 /* Set pan gesture for frontview to swiftleft show leftmenu */
638 if (appDelegate.revealController.recognizesPanningOnFrontView == NO) {
639     appDelegate.revealController.recognizesPanningOnFrontView = YES;
640 }
641 appDelegate.topController = self;
642
643 }
644 -(void)didGoToSegmentContact:(NSNotification*)notif{
645     @autoreleasepool {
646         // [self.navigationController popToRootViewControllerAnimated:YES];
647         MasterSegmentViewController *contactVC = [[IBHelper sharedUIHelper]loadViewController:@"MasterSegment"
648                                                 inStory:@"MainFOND"];
649         contactVC.leftMenu = @"menu";
650
651         [self.navigationController pushViewController:contactVC animated:YES];
652     }
653 }
654 -(void)didGoToLoginScreen:(NSNotification*)notif{
655     [appDelegate disconnect];
656     @autoreleasepool {
657         LoginViewController *loginVC = [[IBHelper sharedUIHelper]loadViewController:@"LoginID"
658                                         inStory:@"MainFOND"];
659         [self presentViewController:loginVC animated:YES completion:^{
660             [NLUtils showAndHideProgressHUD:NO andView:self.view];
661         }];
662         [NLUtils showAlertWithMessage:@"You have been logged out because you logged in on a different device.\nPlease
663 log in again."];
664     }
665 }
666 -(void)dealloc{
667     // [[NSNotificationCenter defaultCenter]removeObserver:self];
668 }
669 - (void)viewDidUnload
670 {
671     [super viewDidUnload];
672     //[[NSNotificationCenter defaultCenter]removeObserver:self];
673
674     // Release any retained subviews of the main view.
675 }
676
677 - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
678 {
679     return (interfaceOrientation == UIInterfaceOrientationPortrait);
680 }
681
682
683 @end
```

Controller - Function

```
-(void)renderBioUserFriend{
    @autoreleasepool {

        /* Setup Bio for user */

        if (!appDelegate.friendProfile[@"bio"] || [appDelegate.friendProfile[@"bio"] isKindOfClass:[NSNull class]]
            ] || [appDelegate.friendProfile[@"bio"] length]==0) {
            self.contentBio.hidden = YES;
            self.contentSubScroller.frame = CGRectMake(0, 131, self.contentSubScroller.frame.size.width, self.
                contentSubScroller.frame.size.height);
        }else{
            self.contentBio.hidden = NO;
            self.lbBio.font = [UIFont fontWithName:@"OpenSans" size:12.0];
            self.contentBio.frame = CGRectMake(self.contentBio.frame.origin.x, self.contentBio.frame.origin.y,
                320, [self.lbBio resizeToFit]);
            self.contentSubScroller.frame = CGRectMake(0, self.contentBio.frame.origin.y + self.contentBio.frame.
                size.height+20, self.contentSubScroller.frame.size.width, self.contentSubScroller.frame.size.
                height);
            [self.content setContentSize:CGSizeMake(320, self.contentSubScroller.frame.origin.y+self.
                contentSubScroller.frame.size.height)];

            self.circleBio.frame = CGRectMake(self.circleBio.frame.origin.x, self.circleBio.frame.origin.y, 16,
                16);
            [self.contentBio.subviews enumerateObjectsUsingBlock:^(id obj, NSUInteger idx, BOOL *stop) {
                if ([obj isKindOfClass:[UILabel class]]) {
                    UILabel *lb = (UILabel*)obj;
                    lb.textColor = [UIColor colorWithRed:166.0/255.0 green:74.0/255.0 blue:135.0/255.0 alpha:1.0]
                    ;
                }
            }];
            self.lbTitleAbout.font = FONDWithSemiBold(12);
            self.lbTitleFriends.font = FONDWithSemiBold(12);
            self.lbTitleInterest.font = FONDWithSemiBold(12);
            self.lbBio.textColor = [UIColor blackColor];
            NSMutableParagraphStyle *paragraphStyles = [[NSMutableParagraphStyle alloc] init];
            paragraphStyles.alignment
                = NSTextAlignmentJustified;
            paragraphStyles.firstLineHeadIndent
                = 0.05; // Very IMP

            NSString *stringTojustify
                = NotNullString(appDelegate.friendProfile[@"bio"]);
            NSDictionary *attributes
                = @{@"NSParagraphStyleAttributeName": paragraphStyles};
            NSAttributedString *attributedString
                = [[NSAttributedString alloc] initWithString:stringTojustify
                    attributes:attributes];

            self.lbBio.attributedText
                = attributedString;
            self.lbBio.numberOfLines
                = 0;
            [self.lbBio sizeToFit];

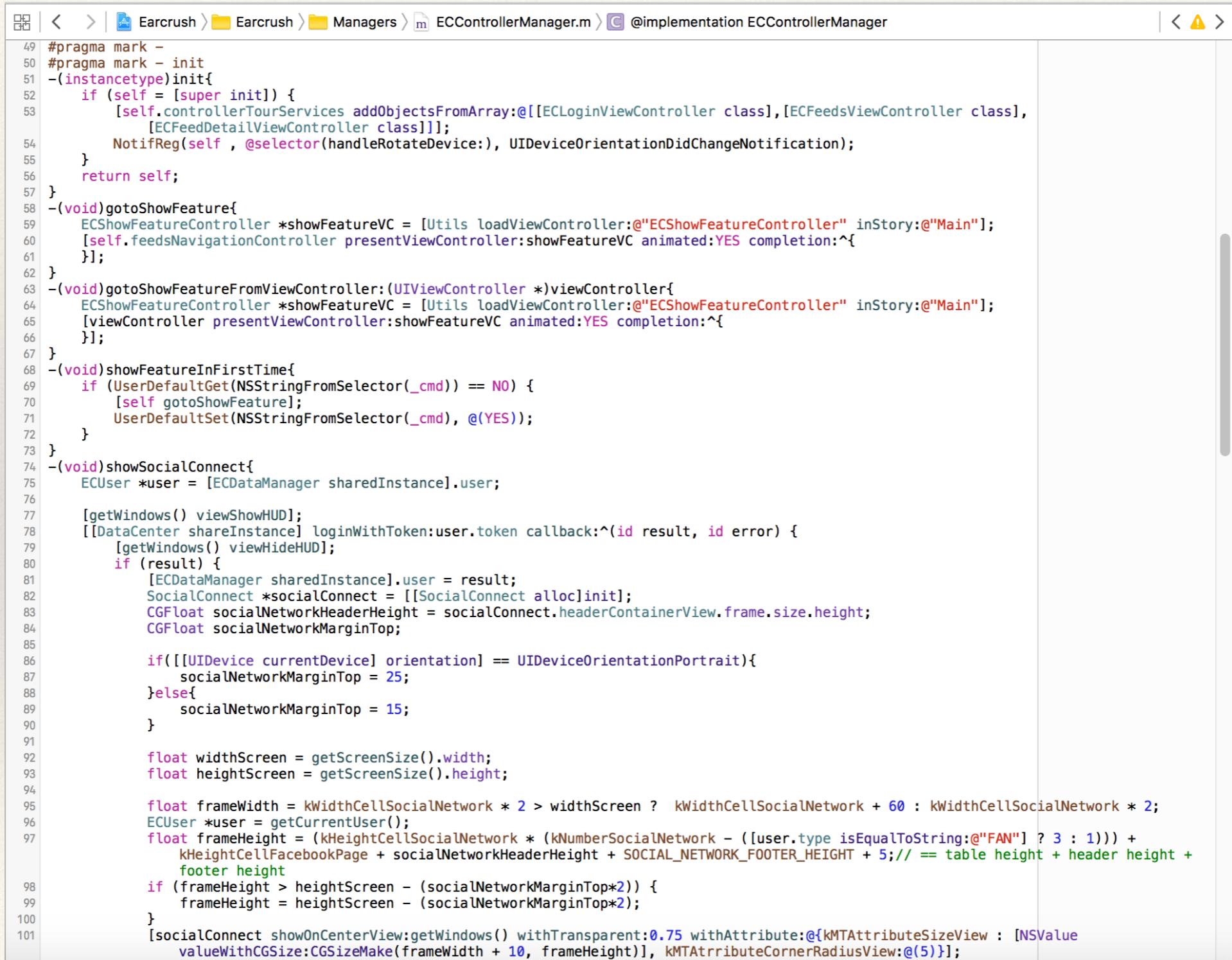
        /* Check status friend */
    }
}
```

Controller

```
#pragma mark - Customize Subview
-(void)customizeSubView{
    @autoreleasepool {
        /*set background color for content view */
        [self.contentView setBackgroundColor:[UIColor colorWithRed:244.0/255.0 green:244.0/255.0 blue:244.0/255.0 alpha:1.0]];

        /* Set font for label in content*/
        self.lbName.font = FONDWithFontBold(14);
        self.lbName.textColor = [UIColor colorWithRed:166.0/255.0 green:74.0/255.0 blue:135.0/255.0 alpha:1.0];
        self.lbAddress.font = FONDWithFontNormal(10);
        self.lbActive.font = FONDWithFontNormal(9);
        if ([appDelegate.friendProfile[@"age"] integerValue] > 0 || [appDelegate.friendProfile[@"age"]
            isKindOfClass:[NSNull class]] ) {
            self.lbName.text = [NSString stringWithFormat:@"%@ %@,
                %@",NotNullString(appDelegate.friendProfile[@"firstname"]),NotNullString(appDelegate.
                friendProfile[@"lastname"]),NotNullString(appDelegate.friendProfile[@"age"])];
        }else{
            self.lbName.text = [NSString stringWithFormat:@"%@%
                %@",NotNullString(appDelegate.friendProfile[@"firstname"]),NotNullString(appDelegate.
                friendProfile[@"lastname"])];
        }
        if ([appDelegate.friendProfile[@"city"] isKindOfClass:[NSNull class]] ||
            [appDelegate.friendProfile[@"city"] length] == 0) {
            self.lbAddress.text = [NSString
                stringWithFormat:@"%@,NotNullString(appDelegate.friendProfile[@"country_name"])];
        }else{
            self.lbAddress.text = [NSString stringWithFormat:@"%@,
                %@",NotNullString(appDelegate.friendProfile[@"city"]),NotNullString(appDelegate.friendProfile[@"c
                ountry_name"])];
        }
        self.lbFriends.text = [NSString
            stringWithFormat:@"(%@)",NotNullString(appDelegate.friendProfile[@"recent_friend"][@"total_friend"])]
        ;
        [self setUpTimeActive:NotNullInteger(appDelegate.friendProfile[@"active_time"])];
    }
}
```

Controller



The screenshot shows a code editor window in Xcode displaying the `ECControllerManager.m` file. The file is an implementation of the `ECControllerManager` class. The code handles various methods such as `init`, `gotoShowFeature`, `showFeatureInFirstTime`, and `showSocialConnect`. It uses Objective-C syntax with some modern C-style comments and includes imports for `ECLoginViewController`, `ECFeedsViewController`, `ECFeedDetailViewController`, and `ECShowFeatureController`. The code also interacts with `UserDefaults` and `DataCenter` instances.

```
#pragma mark -
#pragma mark - init
-(instancetype)init{
    if (self = [super init]) {
        [self.controllerTourServices addObjectsFromArray:@[[ECLoginViewController class],[ECFeedsViewController class],
                                                       [ECFeedDetailViewController class]]];
        NotifReg(self , @selector(handleRotateDevice:), UIDeviceOrientationDidChangeNotification);
    }
    return self;
}
-(void)gotoShowFeature{
    ECShowFeatureController *showFeatureVC = [Utils loadViewController:@"ECShowFeatureController" inStory:@"Main"];
    [self.feedsNavController presentViewController:showFeatureVC animated:YES completion:^{
    }];
}
-(void)gotoShowFeatureFromViewController:(UIViewController *)viewController{
    ECShowFeatureController *showFeatureVC = [Utils loadViewController:@"ECShowFeatureController" inStory:@"Main"];
    [viewController presentViewController:showFeatureVC animated:YES completion:^{
    }];
}
-(void)showFeatureInFirstTime{
    if (UserDefaultGet(NSStringFromSelector(_cmd)) == NO) {
        [self gotoShowFeature];
        UserDefaultSet(NSStringFromSelector(_cmd), @(YES));
    }
}
-(void)showSocialConnect{
    ECUser *user = [ECDataManager sharedInstance].user;

    [getWindows() viewShowHUD];
    [[DataCenter sharedInstance] loginWithToken:user.token callback:^(id result, id error) {
        [getWindows() viewHideHUD];
        if (result) {
            [ECDataManager sharedInstance].user = result;
            SocialConnect *socialConnect = [[SocialConnect alloc]init];
            CGFloat socialNetworkHeaderHeight = socialConnect.headerContainerView.frame.size.height;
            CGFloat socialNetworkMarginTop;

            if([[UIDevice currentDevice] orientation] == UIDeviceOrientationPortrait){
                socialNetworkMarginTop = 25;
            }else{
                socialNetworkMarginTop = 15;
            }

            float widthScreen = getScreenSize().width;
            float heightScreen = getScreenSize().height;

            float frameWidth = kWidthCellSocialNetwork * 2 > widthScreen ? kWidthCellSocialNetwork + 60 : kWidthCellSocialNetwork * 2;
            ECUser *user = getCurrentUser();
            float frameHeight = (kHeightCellSocialNetwork * (kNumberSocialNetwork - ([user.type isEqualToString:@"FAN"] ? 3 : 1))) +
                kHeightCellFacebookPage + socialNetworkHeaderHeight + SOCIAL_NETWORK_FOOTER_HEIGHT + 5;// == table height + header height +
                footer height
            if (frameHeight > heightScreen - (socialNetworkMarginTop*2)) {
                frameHeight = heightScreen - (socialNetworkMarginTop*2);
            }
            [socialConnect showOnCenterView:getWindows() withTransparent:0.75 withAttribute:@{kMTAttributeSizeView : [NSNumber
                valueWithCGSize:CGSizeMake(frameWidth + 10, frameHeight)], kMTAttributeCornerRadiusView:@(5)}];
        }
    }];
}
```

-
-
1. Backend side
 2. Controller quá lớn

Best Practice

Pods

```
platform :ios, '9.0'
use_frameworks!

def important_pods
  pod 'CocoaLumberjack' -> Log
  pod 'Nuke' -> Image-Caching
  pod 'JSONModel' -> JSONModel-
  pod 'Alamofire', '~> 3.4' -> Network
  pod 'Reachability', :git => 'https://github.com/ashfurrow/Reachability.git', :branch => 'frameworks' -> Internet
  pod 'BFKit-Swift' -> Helper classes
  pod 'TTTAttributedLabel' -> Attribute Label
end

def test_pods
  pod 'Quick'
  pod 'Nimble'
  pod 'OCMockito', '~> 3.0'
  pod 'FBSnapshotTestCase'
end

target "Tek Talk" do
  important_pods
end

target "Tek TalkTests" do
  important_pods
  test_pods
end
```

API Manager

<https://www.draw.io/#DAPIManager>

APIManager

