iOS HELP

iOS SWIFT 13 Change
Changes in iOS 13

didRegisterForRemoteNotificationsWithDeviceToken
[deviceToken description]

iOS 12
<dd6ea57a 19bf0a05 c51692b6 e116c512 25fa04ef 9f84081d 576cc42f 639b62c3>

iOS 13
{length = 32, bytes = 0x35b57c5f cf576f17 55d221d4 67892ec4 ... 065ecf46 e1181548 }

You may use this method to fetch the device token on iOS 13 onwards:

Objective-C:

```
+ (NSString *)stringFromDeviceToken:(NSData *)deviceToken {
   NSUInteger length = deviceToken.length;
   if (length == 0) {
      return nil;
   }
   const unsigned char *buffer = deviceToken.bytes;
   NSMutableString *hexString  = [NSMutableString stringWithCapacity:(length * 2)];
   for (int i = 0; i < length; ++i) {
      [hexString appendFormat:@"%02x", buffer[i]];
   }
   return [hexString copy];
}
```

Swift 5.0 (Untested)

```
class func string(fromDeviceToken deviceToken: Data?) -> String? {
   let length = deviceToken?.count ?? 0
```

```
    if length == 0 {
        return nil
    }
    let buffer = UInt8(deviceToken?.bytes ?? 0)
    var hexString = String(repeating: "\0", count: length * 2)
    for i in 0..<length {
        hexString += String(format: "%02x", buffer[i])
    }
    return hexString
}
```

____

Common Bundle Id :

com.ebizz.temp

Notification Center Inline :

```
NotificationCenter.default.post(name: Notification.Name("ChatScreen"), object:
info, userInfo: nil)

NotificationCenter.default.addObserver(forName:
NSNotification.Name(rawValue: "ChatScreen"), object: nil, queue: nil)
{ (notification) in

            let object = notification.object as? NSDictionary ?? nil
//            UserDefaults.standard.set(object, forKey: "notification")
            if(object != nil){
                let userToChat : User = User()
                userToChat.firebaseUID = object!["sender_id"] as? String ?? ""
                userToChat.strName = object!["username"] as? String ?? ""
                userToChat.strImageNAme = object!["image_url"] as? String ?? ""

                let chatViewController =
self.storyboard?.instantiateViewController(withIdentifier: "ChatViewController")
as! ChatViewController
                chatViewController.userToChat = userToChat
                self.navigationController?.pushViewController(chatViewController,
animated: true)
            }
        }
```

Background task Mgr :

```
    var streamInBackground: UIBackgroundTaskIdentifier!


applicationDidEnterBackground
{
          self.beginBackgroundTask()
}

applicationDidBecomeActive
{
        self.endBackgroundTask()
}

extension AppDelegate {

    func beginBackgroundTask() {
        self.streamInBackground =
UIApplication.shared.beginBackgroundTask(withName: "streamInBackground",
expirationHandler: {
            self.endBackgroundTask()
        })
    }

    func endBackgroundTask() {
        if self.streamInBackground != UIBackgroundTaskIdentifier.invalid &&
(self.streamInBackground != nil) {
            UIApplication.shared.endBackgroundTask(self.streamInBackground)
            self.streamInBackground = UIBackgroundTaskIdentifier.invalid
        }
    }

    func setupNotification() {
        let notificationCenter = UNUserNotificationCenter.current()
        notificationCenter.delegate = self
        let options: UNAuthorizationOptions = [.alert, .sound, .badge]
        notificationCenter.requestAuthorization(options: options) {
            (didAllow, error) in
            if !didAllow {
                print("User has declined notifications")
            }
        }
    }

    func userNotificationCenter(_ center: UNUserNotificationCenter, willPresent
notification: UNNotification, withCompletionHandler completionHandler:
@escaping (UNNotificationPresentationOptions) -> Void) {
        completionHandler([.alert,.sound])
    }

}
```

P12 to Pem :

```
openssl pkcs12 -in pushcert.p12 -out pushcert.pem -nodes -clcerts
```

Delete Archieve :

```
~/Library/Developer/Xcode/Archives
```

OTP Textfield :

```
//Setup TextField
        txtOTP1.addTarget(self, action: #selector(textFieldDidChange(textField:)),
for: .editingChanged)
        txtOTP2.addTarget(self, action: #selector(textFieldDidChange(textField:)),
for: .editingChanged)
        txtOTP3.addTarget(self, action: #selector(textFieldDidChange(textField:)),
for: .editingChanged)
        txtOTP4.addTarget(self, action: #selector(textFieldDidChange(textField:)),
for: .editingChanged)


/**
     this the text file change method
    **/
    @objc func textFieldDidChange(textField: UITextField){

        let text = textField.text

        if (text?.utf16.count)! >= 1 {
            switch textField {
            case txtOTP1:
                txtOTP2.becomeFirstResponder()
            case txtOTP2:
                txtOTP3.becomeFirstResponder()
            case txtOTP3:
                txtOTP4.becomeFirstResponder()
            case txtOTP4:
                txtOTP4.resignFirstResponder()
            default:
                break
            }
        } else {

        }
    }


 //MARK:- UITextField Delegate
    func textField(_ textField: UITextField, shouldChangeCharactersIn range:
NSRange, replacementString string: String) -> Bool {
        let maxLength = 1
        let currentString: NSString = textField.text! as NSString
        let newString: NSString =
            currentString.replacingCharacters(in: range, with: string) as NSString
        return newString.length <= maxLength
    }


extension OTPViewController: UITextFieldDelegate {
    func textFieldDidBeginEditing(_ textField: UITextField) {
        textField.text = ""
    }
}
```

Notification delay:

https://stackoverflow.com/questions/9204767/how-much-delay-of-ios-push-notification#
https://docs.telerik.com/platform/backend-services/javascript/push-notifications/troubleshooting/push-trb-ios.html
https://medium.com/yalantis-mobile/uber-underlying-technologies-and-how-it-actually-works-526f55b37c6f

HELP - HUBUC - User Defaults

Replace data of For Loop in clearAll() in ALUserDefaultsHandler.m

```
NSMutableArray *arrMyData = [[NSMutableArray alloc]
initWithObjects:@"isUserLogin",@"LoginData",@"ReturnedToken", @"OtherReturnedToken", @"UserRole",
@"device_token", @"CurrentLatitude", @"CurrentLongitude",  nil];

    if (![arrMyData containsObject:defaultKeyString]) {
      if([defaultKeyString hasPrefix:KEY_PREFIX] && ![defaultKeyString isEqualToString:APN_DEVICE_TOKEN])
      {
        [userDefaults removeObjectForKey:defaultKeyString];
        [userDefaults synchronize];
      }
    }
```

Cell Button click

# Swift :-

```
var btnOpenDateTap : (() -> Void)? = nil

@IBAction func stepBtnOpenDatePicker(_ sender: Any) {
    if let function = self.btnOpenDateTap{
         function()
    }

}

Cell.btnOpenDateTap = {

}
```

# Obj C :-

```
@property (nonatomic, copy) void(^btnCloseTap)(void);

- (IBAction)btnCloseClick:(id)sender {
   self.btnCloseTap();
}

cell.btnCloseTap = ^{

};
```

IMAGE PICKER

Delegate :

```
        UIImagePickerControllerDelegate, UINavigationControllerDelegate
```

Initialisation :

```
    //Image Picker
    let imagePicker = UIImagePickerController()
    var imageData:String = ""
```

btnClick :

```
        imagePicker.delegate = self
        imagePicker.allowsEditing = true
        imagePicker.sourceType = .photoLibrary
        present(imagePicker, animated: true, completion: nil)
```

UITextView Placeholder

```
- (void)viewDidLoad {
    commentTxtView.text = @"Comment";
    commentTxtView.textColor = [UIColor lightGrayColor];
    commentTxtView.delegate = self;

}
- (BOOL) textViewShouldBeginEditing:(UITextView *)textView {
    commentTxtView.text = @"";
    commentTxtView.textColor = [UIColor blackColor];
    return YES;
}

-(void) textViewDidChange:(UITextView *)textView {

    if(commentTxtView.text.length == 0) {
        commentTxtView.textColor = [UIColor lightGrayColor];
        commentTxtView.text = @"Comment";
        [commentTxtView resignFirstResponder];
    }
}
```

```
-(void) textViewShouldEndEditing:(UITextView *)textView {

    if(commentTxtView.text.length == 0) {
        commentTxtView.textColor = [UIColor lightGrayColor];
        commentTxtView.text = @"Comment";
        [commentTxtView resignFirstResponder];
    }
}
```

Photo Picker - iOS

```
- (IBAction)addImage:(id)sender {
    UIActionSheet *action = [[[UIActionSheet alloc] initWithTitle:@"Select image
from"
                                                         delegate:self
                                                cancelButtonTitle:@"Cancel"
                                           destructiveButtonTitle:nil
                                                otherButtonTitles:@"From library",@"From
camera", nil] autorelease];

    [action showInView:self.view];
}

#pragma mark - ActionSheet delegates

- (void)actionSheet:(UIActionSheet *)actionSheet clickedButtonAtIndex:
(NSInteger)buttonIndex
{
    if( buttonIndex == 0 ) {

        if ([UIImagePickerController
isSourceTypeAvailable:UIImagePickerControllerSourceTypeCamera]) {
             UIImagePickerController *pickerView =[[UIImagePickerController
alloc]init];
             pickerView.allowsEditing = YES;
             pickerView.delegate = self;
             pickerView.sourceType = UIImagePickerControllerSourceTypeCamera;
             [self presentViewController:pickerView animated:YES completion:nil];
        }
    }else if( buttonIndex == 1 ) {

        UIImagePickerController *pickerView = [[UIImagePickerController alloc]
init];
```

```
        pickerView.allowsEditing = YES;
        pickerView.delegate = self;
        [pickerView
setSourceType:UIImagePickerControllerSourceTypePhotoLibrary];
        [self presentViewController:pickerView animated:YES completion:nil];
    }
}

#pragma mark - PickerDelegates

- (void)imagePickerController:(UIImagePickerController *)picker
didFinishPickingMediaWithInfo:(NSDictionary *)info{

    [self dismissViewControllerAnimated:YES completion:nil];

    UIImage * img = [info valueForKey:UIImagePickerControllerEditedImage];

    myImageView.image = img;
}
```

Action sheet for iPad

```
alert.popoverPresentationController?.sourceView = sender
alert.popoverPresentationController?.sourceRect = sender.bounds
alert.popoverPresentationController?.permittedArrowDirections = .up
```

//Device support File

```
/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/
DeviceSupport
```

View Animation :

```
//MARK:- Pop in and out animation
    func openDialog(hideView: UIView, animationView: UIView, type: String){
        hideView.isHidden = false
        animationView.transform = .identity

        animationView.transform = CGAffineTransform(scaleX: 0.01, y: 0.01)
        UIView.animate(withDuration: 0.2, delay: 0, options: .curveEaseOut,
animations: {() -> Void in
            animationView.transform = .identity
        }, completion: {(finished: Bool) -> Void in
            // do something once the animation finishes, put it here
        })
    }

    func closeDialog(hideView: UIView, animationView: UIView, type: String){
        UIView.animate(withDuration: 0.2, delay: 0, options: .curveEaseOut,
animations: {() -> Void in
            animationView.transform = CGAffineTransform(scaleX: 0.01, y: 0.01)
        }, completion: {(finished: Bool) -> Void in
            hideView.isHidden = true
            if(type == "true"){
                self.getExamList(start: self.strStartDateMain, end:
self.strEndDateMain)
            }
        })
    }
```

Redirection - Profile

//F1cebook (P1geId)

```
if ([[UIApplic,tion sh,redApplic,tion] c,nOpenURL:[NSURL
URLWithString:@"fb://"]]) {
[[UIApplic,tion sh,redApplic,tion] openURL:[NSURL URLWithString:@"fb://
profile/100022140846004"]];
}
else {
[[UIApplic,tion sh,redApplic,tion] openURL:[NSURL URLWithString:@"https://
www.f,cebook.com/qer,,h.,pp.96"]];
}
```

//Twitter (usern1me)

```
if ([[UIApplic,tion sh,redApplic,tion] c,nOpenURL:[NSURL
URLWithString:@"twitter://"]]) {
[[UIApplic,tion sh,redApplic,tion] openURL:[NSURL URLWithString:@"twitter://
user?screen_n,me=qer3,h"]];
}else{
[[UIApplic,tion sh,redApplic,tion] openURL:[NSURL
URLWithString:@"https://twitter.com/qer3,h"]]; }
```

//Inst1gr1m (usern1me)

```
if ([[UIApplic,tion sh,redApplic,tion] c,nOpenURL:[NSURL
URLWithString:@"inst,gr,m://"]]) {
```

```
[[UIApplic,tion sh,redApplic,tion] openURL:[NSURL
URLWithString:@"inst,gr,m://user?usern,me=qer3,h"]];
}else{
[[UIApplic,tion sh,redApplic,tion] openURL:[NSURL
URLWithString:@"https://www.inst,gr,m.com/qer3,h/"]]; }
```