\*.Facebook integration

(inside LoginActivity)

1.variables

**private static final int *FB\_SIGN\_IN*** =64206 ;

Button **btn\_sign\_in\_facebook**;  
  
  
**private** CallbackManager **mCallbackManager**;  
  
  
FirebaseAuth **mauth**;  
FirebaseAuth.AuthStateListener **mAuthListener**;

2.onCreate()

@Override  
**protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 *//facebook initialization* FacebookSdk.*sdkInitialize*(getApplicationContext());  
 AppEventsLogger.*activateApp*(**this**);  
 **mCallbackManager**=CallbackManager.Factory.*create*();  
 **if**(AccessToken.*getCurrentAccessToken*()!=**null**){  
 LoginManager.*getInstance*().logOut();  
 }  
  
  
  
 setContentView(R.layout.***activity\_login***);  
  
 *//firebase* **mauth**=FirebaseAuth.*getInstance*();  
 **mAuthListener** = **new** FirebaseAuth.AuthStateListener() {  
 @Override  
 **public void** onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {  
  
 }  
 };

3.

@Override  
**protected void** onStart() {  
 **super**.onStart();  
  
}  
  
@Override  
**protected void** onStop() {  
 **super**.onStop();  
  
}

4.

**case** R.id.***btn\_sign\_in\_facebook***:  
 Toast.*makeText*(**this**, **"fb signin"**, Toast.***LENGTH\_SHORT***).show();  
 **progressDialog**.show();  
 fbLogin();  
 **break**;

5.

*//facebook method* **public void** fbLogin() {  
 *// Initialize Facebook Login button  
// mCallbackManager = CallbackManager.Factory.create();  
// LoginButton loginButton = (LoginButton) findViewById(R.id.button\_facebook\_login);  
// loginButton.setReadPermissions("email", "public\_profile");  
// loginButton.registerCallback(mCallbackManager, new FacebookCallback<LoginResult>() {* LoginManager.*getInstance*().logInWithReadPermissions(**this**, Arrays.*asList*(**"public\_profile"**, **"user\_friends"**, **"email"**));  
 LoginManager.*getInstance*().registerCallback(**mCallbackManager**, **new** FacebookCallback<LoginResult>() {  
  
 @Override  
 **public void** onSuccess(LoginResult loginResult) {  
 *// Log.d(TAG, "facebook:onSuccess:" + loginResult);* handleFacebookAccessToken(loginResult.getAccessToken());  
 }  
  
 @Override  
 **public void** onCancel() {  
 *//Log.d(TAG, "facebook:onCancel");* Toast.*makeText*(LoginActivity.**this**, **"User cancels"**, Toast.***LENGTH\_SHORT***).show();  
 **progressDialog**.dismiss();  
 }  
  
 @Override  
 **public void** onError(FacebookException error) {  
 *//Log.d(TAG, "facebook:onError", error);* Log.*d*(**"error:"**, error.toString());  
 Toast.*makeText*(LoginActivity.**this**, **"User gets an error"**, Toast.***LENGTH\_SHORT***).show();  
 **progressDialog**.dismiss();  
 }  
 });  
  
 }  
  
 *//[Authenticating with facebook]* **private void** handleFacebookAccessToken(AccessToken token) {  
 Log.*d*(**"Reached\_handler"**, **"handleFacebookAccessToken:"** + token);  
  
 AuthCredential credential = FacebookAuthProvider.*getCredential*(token.getToken());  
 **mauth**.signInWithCredential(credential)  
 .addOnCompleteListener(**this**, **new** OnCompleteListener<AuthResult>() {  
 @Override  
 **public void** onComplete(@NonNull Task<AuthResult> task) {  
 Log.*d*(**"Reached\_handler"**, **"signInWithCredential:onComplete:"** + task.isSuccessful());  
  
 *// If sign in fails, display a message to the user. If sign in succeeds  
 // the auth state listener will be notified and logic to handle the  
 // signed in user can be handled in the listener.* **if** (!task.isSuccessful()) {  
 Log.*w*(**"Reached\_handler"**, **"signInWithCredential"**, task.getException());  
 Toast.*makeText*(LoginActivity.**this**, **"Authentication failed."**,  
 Toast.***LENGTH\_SHORT***).show();  
 **progressDialog**.dismiss();  
 }  
 **else** {  
 startActivity(**new** Intent(LoginActivity.**this**,HomeActivity.**class**));  
 Toast.*makeText*(LoginActivity.**this**, **"Successfully logged in"**, Toast.***LENGTH\_SHORT***).show();  
 **progressDialog**.dismiss();  
 }  
  
 }  
 });  
 }  
  
  
 @Override  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
  
 *//Differentiate fb and google on the basis of requestCode* **if**(requestCode ==***GOOGLE\_SIGN\_IN***) {  
 *//for google* GoogleSignInResult result=Auth.***GoogleSignInApi***.getSignInResultFromIntent(data);  
 handleSignInResult(result);  
 }  
 **else if**(requestCode==***FB\_SIGN\_IN***){  
 *//for fb* **mCallbackManager**.onActivityResult(requestCode, resultCode, data);  
 }  
 }

**Facebook (Internet settings)**

1.android google firebase

<https://firebase.google.com/docs/android/setup>

2.go to end of page->authentication->

**Federated identity provider integration->**

|  |  |
| --- | --- |
|  |  |
| Facebook | [iOS](https://firebase.google.com/docs/auth/ios/facebook-login) [ANDROID](https://firebase.google.com/docs/auth/android/facebook-login) [WEB](https://firebase.google.com/docs/auth/web/facebook-login) [C++](https://firebase.google.com/docs/auth/cpp/facebook-login) |

Choose android

3.this page ->

<https://firebase.google.com/docs/auth/android/facebook-login>

4.developers.facebook.com link->

<https://developers.facebook.com/>

# choose Facebook Login->getting started

# ->Add Facebook Login to Your App or Website

# ->android

# 5.quickStart link opens

# <https://developers.facebook.com/docs/facebook-login/android>

# //follow the link……….

# 6.include gradle code,manifest code,enter project name,enter dev key,release key,enter hash generated (in android studio),enable single signon.

# 7.code for development key in android studio SplashScreenActivity.

*/\* //for key hash  
 try {  
 PackageInfo info = getPackageManager().getPackageInfo(  
 "com.kpf.sujeet.chat",  
 PackageManager.GET\_SIGNATURES);  
 for (Signature signature : info.signatures) {  
 MessageDigest md = MessageDigest.getInstance("SHA");  
 md.update(signature.toByteArray());  
 Log.d("KeyHash:", Base64.encodeToString(md.digest(), Base64.DEFAULT));  
 }  
 } catch (PackageManager.NameNotFoundException e) {  
  
 } catch (NoSuchAlgorithmException e) {  
  
 }\*/*