Facebook sdk unity

This open-source library allows you to integrate Facebook into your Unity app. Learn more about the provided samples, documentation, integrating the SDK into your app, accessing source code and more, HTTPS://DEVELOPERS.FACEBOOK.COM/DOCS/UNITY try to figure it out please give feedback to the features You can join the Facebook Developers Group on Facebook (or ask questions on Stack Overflow ( ) contributions We are able to accept contributions to the Facebook SDK for integration. Please do the following to contribute. Follow the instructions in contributing.mdown. Submit your pull request to dev branch. This allows us to merge our changes into our internal master and then push the changes into the next release. The SDK setup is licensed under the Build Plugin Binary./Scripts/Setup.sh./Scripts/package_sdk.sh license other than otherwise mentioned, Facebook SDK for Unity Facebook Platform License (. Unless required by applicable law or agreed in writing, the software distributed under the license is distributed on the grounds as it is, without warranty or conditions of any kind, either express or implied. See licenses for specific language that controls permissions and limitations under the license. By enabling developer terms Facebook integration, including through this SDK, you can share information with Facebook, including information about people's use of your app. Facebook will use information obtained in accordance with our Data Usage Policy (, including providing you with information about the effectiveness of your ads and the use of your app. These integrations also enable us and our partners to serve ads on and off Facebook. You can limit sharing your information with us by updating the Insights control in developer tools (app_id]/settings/advanced). If you use Facebook Integration, including sharing information with us, you agree and confirm that you have provided and obtained appropriate and sufficiently prominent information about such collection, use and disclosure from your users (including, at least, through your Privacy Policy). You further agree that you will not share information with us about children under the age of 13. You agree to comply with all applicable laws and regulations and also agree to our Terms (including our Platform Policies (and Advertising Guidelines, as applicable (are included. You agree to these terms by using the Facebook SDK for Unity. Page 2 View 80 Star 307 Fork 204 You cannot do that action at this time. You have another tab or window Signed in. To reload your session. You signed out in another tab or window. Reload to refresh your session. Package Date Changelog 8.1.1 Nov 13, 2020 Change Log 8.1.0 Oct 20, 2020 Change Log 8.0.0 Sep 28, 2020 Change Log 7.21.2 Aug 14, 2020 Change Log 7.21.1 Jun 25, 2020 Change Log 7.21.0 Jun 23, 2020 Change Log 7.20.0 Jun 15, 2020 Change Log 7.19.2 April 8, 2020 Change Log 7.19.1 Mar 23, 2020 Change Log 7.19.0 Mar 11, 2020 Change Log 7.18.1 Feb 3, 2020 Change Log 7.18.0 Oct 11, 2019 Change Log 7.17.2 July 18, 2019 Change Log 7.17.1 July 5, 2019 Change Log 7.17.0 June 27, 2019 Change Log 7.16.1 June 21, 2019 Change Log 7.16.0 June 6, 2019 Change Log 7.15.1 Jan 10, 2019 Change Log 7.15.0 Dec 17, 2018 Change Log 7.14.1 Nov 29, 2018 Change Log 7.14.0 Nov 2 , 2018 Change Log 7.13.0 June 26, 2018 Change Log 7.12.2 June 6, 2018 Change Log 7.12.1 May 18, 2018 Change Log 7.12.0 May 11, 2018 Change Log 7.11.1 March 18, 2018 Change Log 7.11.0 January 31, 2018 Change Log 7.10.1 October 20, 2017 Change Log 7.10.0 June 16, 2017 Change Log 7.9.4 February 13, 2016 Change Log 7.9.0 November 9, 2016 Change Log 7.8.0 July 28, 2016 Change Log 7.7.0 July 6, 2016 Change Log 7.6.0 June 7, 2016 Change Log 7.5.0 April 18, 2016 Change Log 7.4.0 Feburary 1, 2016 Change Log 7.3.0 December 9, 2015 Change Log 7.2.2 November 2, 2015 Change Log 7.2.1 October 26, 2015 Change Log 7.2.0 October 13, 2015 Change Log 7.1.0 September 21, 2015 Change Log 6.2.2 April 14 , 2015 Change Log 6.2.1 January 28, 2015 Change Log 6.2 December 2, 2014 Change Log 6.1 December 17, 2014 Change Log 6.0 August 11, 2014 Change Log 5.1 April 1, 2014 Change Log 5.0.4 February 20, 2014 Change Log 5.0.3 January 23, 2014 Change Log 4.3.6 November 21, 2013 Change Log 4.3.4 October 22, 2013 Change Log 4.3.3 October 13, 2013 Change Log 4.2.4 September 24, 2013 Change Log 4.2.1 September 12, 2013 Change Log 3.1.2 September 6, 2013 Change Log 3.1.1 September 3, 2013 Change Log 3.1 August 26, 2013 Change Log 3.0 August 20, 2013 Change Log For comprehensive detail on the SDK and its supported API, please see the SDK Reference.FeedbackReport bugs with the Unity SDK tag on our bug tool. Please post feedback on our Facebook group or on Stack Overflow using the tag facebook-unity-sdk. We appreciate your help. สวัสดีปีใหม่ครับบบบบเริ่มต่อกันเลยนะครับ फेसबुक1 पर ऐप बनाने के बाद। मंच चयन है कि हम चाहते हैं (उदाहरण से में एंड्रॉयड योग्य चुनूंगा) 3 बनाओ। कुंजी हैश अनुभाग में, यदि हम Google Play को प्रकाशित करना चाहते हैं, तो कीटूल -निर्यातकर्ट <android release= key=> -उर्फ -कीस्टोर <% HOMEPATH%\.android\release.keystore> l base64 को बदलकर <एंड्रॉइड रिलीज की> हमारे द्वारा बनाई गई कीस्टोर के उपनाम पर, फिर हमारे स्टोर में < पथ बदल दिया गया>। Unity3D खोलें, फिर फेसबुक लॉगिन.5 के लिए एक बटन बनाएं। फिर एक स्क्रिप्ट सी # (FacebookController.cs) बनाएं और स्क्रिप्ट लें। </android>गेमऑब्जेक्ट 6। ทำการเพิ่ม using Facebook.Unity; เข้าไปใน Script7. เพิ่มโค้ดในส่วนของ Initializing SDK ใน function Awake() void Awake() { if (!FB.IsInitialized) { // เป็นการเรียกใช้ Initialize บน Facebook SDK ซึ่งจะมีการเรียก function callback FB.Init(InitCallback, OnHideUnity); } else { // ถ้ามีการ Initialize แล้วก็ให้ app ทำงานต่อ // ActivateApp จะเป็นการเก็บข้อมูลของ User ซึ่งจะส่งไปยัง Facebook FB.ActivateApp(); } } private void OnHideUnity(bool isGameShown) { if (!isGameShown) { // ให้หยุดเกม เมื่อมีการแสดง Time.timeScale = 0; } else { // Resume the game - we're getting focus again Time.timeScale = 1; } } private void InitCallback() { if (FB.IsInitialized) { // ถ้ามีการ Initialize แล้วก็ให้ app ทำงานต่อ // ActivateApp จะเป็นการเก็บข้อมูลของ User ซึ่งจะส่งไปยัง Facebook FB.ActivateApp(); } else { // Initialize ไม่สำเร็จ Debug.Log(Failed to Initialize the Facebook SDK); } } 8. ทำการสร้าง Publc function เพื่อใส่ให้กับปุ่ม Login // function ใช้สำหรับอ่านข้อมูล User public void LoginWithReadPermission() { // กำหนดสิทธิ์การเข้าถึง และ แสดงหน้า Login var perms = new List<string>() { public_profile, email, user_friends }; FB.LogInWithReadPermissions(perms, AuthCallback); } private void AuthCallback(ILoginResult result) { if (FB.IsLoggedIn) { // ทำการ Get Name ของ User FB.API(/me?fields=name, HttpMethod.GET, DisplayUsername); } else { Debug.Log(User cancelled login); } } private void DisplayUsername(IGraphResult result) { if (result.Error != null) { return; } Debug.Log(result.ResultDictionary[name]); } 9. ทำการ Add function LoginWithReadPermission() ให้กับปุ่ม Login10. จากนั้นทำการรันเพื่อทดสอบได้เลยครับวิธีการทดสอบ1. जब आप लॉगिन चलाते हैं और दबाते हैं, तो प्रेस एक्सेस टोकन ढूंढें और फेसबुक पेज खोल देगा, उस लिंक को दबाएंगे जो कहता है कि अनुमति देने की आवश्यकता है, आपको उपयोगकर्ता टोकन मिलेगा, इसे कॉपी करेंगे और इसे उपयोगकर्ता एक्सेस टोकन फील्ड में दर्ज करेंगे और प्रेस सेंड सक्सेस में प्रवेश करेंगे। मैं vdo बनाने के लिए समय खोजने के लिए यह आसान समझने के लिए यह अब लागू किया जा सकता है बनाने की कोशिश नहीं करने के लिए माफी चाहता हूं, कोई गेम कोई सपना नहीं!!!! इस ट्यूटोरियल में, हम अपने गेमस्पार्क्स एसडीके में एकता फेसबुक एसडीके को एकीकृत करने के तरीके को कवर करेंगे और अपने फेसबुक खाते का उपयोग करके अपने गेम में अपने प्रमाणित उपयोगकर्ता या लॉगिन करें। इस ट्यूटोरियल के लिए आपको आवश्यकता होगी: हमारे गेमस्पार्क्स एसडीके का वर्तमान संस्करण, यहां उपलब्ध है। ये एसडीके परिवर्तन के अधीन हैं और अक्सर तरीके बेमानी हो जाते हैं या फेसबुक में लॉगइन करने का प्रारूप बदल जाएगा। </string>A working understanding of the Unity3D platform, as well as how our Game Sparks certification works. In this tutorial we will look at the following themes: creating a new Facebook app to connect your users to your game via Facebook, you must first install a new Facebook app. Before doing so, you need to register as a Facebook developer. We won't be going through this process in this tutorial, but if you want to figure out how to register then here's a step-by-step guide available. 1. Once you've logged into your developer profile, you can create a new app by navigating to the 'My Apps' drop-down panel in the top-right corner of the page. Website changes! Facebook periodically changes the layout of its website and includes a Facebook developer page, so this page may not look like the image above at all. 2. Add a new app option. You will be presented with a panel where you can select the platform for which you want to create the app. For this tutorial we will select an iOS app, but the setup is practically the same for both iOS and Android, and for this tutorial you will not need a Mac to complete it. Apps for desktops are not supported! Currently Facebook does not support apps for desktop platforms. This means you won't be able to create apps compatible with Windows or Mac OS. When you choose to create a new app, a pop-up appears that allows you to fill in your app details: 3. Enter details for your new app: Display Name - Name Your app will be given on your Facebook page. Namespace - an important unique ID for your app. Category/sub-category - the last thing to choose is a category and sub-category for your app - we've chosen the game/puzzle for this example. Test apps? You can select your app to become a trial version if you wish. We encourage you to read details about facebook test version apps, but this tutorial doesn't require it, so the option is yours. 4. Click on Create App ID. Once your app is created, you'll be presented with the home-page for your app: we'll come back to this page, but the key details for now are the app ID and app secret. When we set up our Gamespark game with our Facebook app we'll need them in the next section. Integrating your Facebook app with Game Sparks this section of the tutorial assumes that you've already created a game on the Game Sparks platform. However, if you are new to the platform, you can check out our quick start guide available here. So, the next thing we want to do is to integrate our Facebook app with our Game Sparks game. We do this by adding Facebook app details to our list of third-party integrations. 1. When you enter your game's landing page in the Game Sparks portal, you need to navigate to Configurator > Integration. 2. Select Facebook. 3. Click edit. Pages allow you to edit your Facebook configuration Adjusts for. 4. Enter Facebook app ID and Facebook secret, which you can get from your Facebook app's home page. are. Seen in the last section. 5. Click to save Facebook configuration. When you've saved these details, your GamesSpark game is ready to start authenticating your Facebook users. Setting up the Unity Project in this section, we are going to use a new 2D project so that we can set up a very basic interface to connect Gamespark to Facebook. Setting up the Game Sparks SDK is the first thing we need to do to import our Game Sparks SDK package. 1. You can do this by navigating to the top menu of the Unity3D Editor and selecting the asset/import package/custom package. 2. Then select the package downloaded from the link given in the introduction. When the Game Sparks SDK has been compiled, you should see that the Game Sparks option appears in the top-menu bar of the Unity3D editor: 3. If you click edit settings option in the drop-down menu, you'll find The Game Sparks Settings Manager in the Inspector window. The important details we need here are the Game Sparks API key and GameSpark API Secret, which you can find in the game overview page of your Game Sparks game: To get the Game Sparks API secret, you can click on the show to reveal it and then click on copy. Enter game sparks API keys and Game Sparks API secret in related areas under Game Sparkssetting in Unity3D Editor. Once you've entered the key and secret, you can click on the Test Configuration button – so make sure your details are correct! If you've entered the correct details, you should see gamessparks connect properly in test-view. Setting up the Facebook SDK so we need to import the Facebook Unity SDK. This is done in the same way as Game Sparks was outlined for importing the SDK. 1. Once the package has been compiled, you should see the Facebook option in the top-menu bar of unity3D editor. 2. If you select the Edit Settings option in the drop-down menu, you should get the FacebookSettings window in the Inspector panel. You don't have to worry about everything here. Important details are app name and app ID. Now, it's different from the details used to link your Game Sparks game with your Facebook app: The app ID is the same, but the app name is the original name you gave your app: 3. Enter the app ID and app name in the respective areas under FacebookSettings in Unity3D Editor: Once you've entered the details in Unity, you're good to go! First of all setting up the test scene, you need to create a new scene. In this instance, we will call it FBConnect_TestScene. In this example, we are going to use the Unity-UI system and we need to install this view like this. Create a new empty game-object called GameSparksManager. Apply GameSparksUnity.cs script to that object. Add a new UI/Canvas to canvas: We're calling these fields username fields and connection infofields, respectively. We also added a panel behind these areas So they stand out. Add two input-fields: We have called these userNameInput and passwordinput We've also added a panel behind these areas so they stand out. Add three new buttons to the panel: we have called these buttons login, device login and Facebook login respectively. So with all these elements on the main canvas, your view should look something like this: Now that our test view is established, we need to link all the visual elements to our GamesSpark and Facebook login calls and buttons. 1. The first thing is to add the script to the Game Sparksmanager object. In this tutorial, we are calling this script LoginUI.cs. 2. First, we will combine text-fields and input-fields to our login UI script. To do this, we just need to add some public variables to the login UI script. Public Text Usernamefield, Connection Infofield; Public inputfield username, passwordinput; 3. You'll then need to link each of these variables to ui objects created in the view: When GamesSpark is connected at this point, the first thing we do is something to give us some dialog that lets us know when GamesSpark is connected. Our users should not attempt login if GamesPark is not connected or login will fail: to do this we will use GS. Game Sparks reps available. It is called whenever gamesparks are connected or disconnected. The best strategy is to put it in awake() or start() methods and all we will do for this tutorial is to print some dialogs for connection infofield that will tell us we are connected to GamesSpark: Zero Start(){ GS. Game Sparks Available += OnGameSparksConnected; } Private Zero OnGameSparksConnected (bool _isConnected) { (_isConnected) { Connection InfoField.text = GameSparks Connected...; } and { ConnectionInfoField.text = GameSparks not connected...; } Now, if you run your view, you should look after a second or two connection infofield will show you that you have been linked to GamesPark: User Authentication Next, we will install user authentication using userNameInput, passwordinput and login button. For this tutorial, we're going to show a different approach to this authentication to show how registration and authentication can happen at the same time. This process works as follows: Send an authentication store to enter the username and password. If there is no error in the response, we tell the user that they are certified. If the response has a 'Description: Unfamiliar' error, the user is not registered, so we send a registration request with the same details. If there is no error in the registration response we tell the user that they are certified. Here are the setup steps: 1. This code will be in a public method that we will link to the login button: public zero UserAuthentication_Bttn(){ debug.log (attempted user log...); Debug.Log (username:+ userNameInput.text); Debug.Log (password:+ passwordInput.text); New Game Sparks.API.request. Authentication request(). Sethusarname (userNameInput.text). SetPassword If (!auth_response. HasErrors){ ConnectionInfoField.text = GameSparks certified...; userNameField.text = auth_response. displayname; Otherwise debug.logang (auth_response. Errors. JSON); If (auth_response. Errors. GetString (Description) == Non-Identity) { Debug.Log (User does not exist, registration user...); new gamesparks.api.request. RegistrationRequest(). SetDisplayName (userNameInput.text). Sethusarname (userNameInput.text). SetPassword (passwordInput.text). Send ((reg_response) = > { (!reg_response. HasErrors){ ConnectionInfoField.text = GameSparks certified...; userNameField.text = reg_response. displayname; Otherwise debug.logang (auth_response. Errors. JSON); } }); } } }); } 2. Now, back in your view, you need to link your login button to the method we created. 3. Now you should be able to login using the username and password input field: the next login option we are going to cover is device login: this kind of authentication uses the device's unique identifiers to generate a profile for the user. If you try to authenticate with that device since then, GamesSpark will recognize that device as a unique user. 1. The code for this is similar to user authentication, except that it does not expect a username or password. However, it allows you to input the display name. We will create a method to call this request: public zero DeviceAuthentication_bttn(){ Debug.Log (attempted device authentication...); new game sparks.api.request.DeviceAuthenticationRequest(). Setdisplanem (Player 2). Send ((auth_response) = > { if (!auth_response. HasErrors){ { // For the next part, check to see if we have an error i.e. authentication failed connection IninfoField.text = GameSparks certified...; userNameField.text = auth_response. displayname; Otherwise debug.logang (auth_response. Errors. JSON); If we have errors, print them out }}); } 2. Now, once you've linked your button to the method, you'll have the option to authenticate your user using a unique device ID. Connecting our Facebook users to Game Sparks for the last part of this tutorial, we will join Facebook and then link the user's Facebook account to Game Sparks. This process will work like this: we call the FacebookConnect_bttn() method from the view-UI. We check to see if the Facebook SDK has been initiated, and if so, we will activate the app. If not, we will try to get Facebook started. Once we know the Facebook app is activated, we can try connecting the Game Sparks user to the Facebook account. When we are logged-in, we will then try to link the account to your Game Sparks user using the Facebook access token. Public Zero FacebookConnect_bttn(){ Debug.Log (linking Facebook with Game Sparks...); if (!FB. IsInitialized){ Debug.Log (Facebook Start...); FB. Init (ConnectGamesstogames, Zero); } Other { FB. ActivationApp(); ConnectGamesstoGames (); } Private Zero ConnectGamescontoGames {(FB) Isins}{Isin}{ Debug.Log (enter Facebook...); var perms = new list <string>(){public_profile, email, user_friends}; FB. LogInWithReadPermissions (perms, (result) = > { (FB. IsLoggedIn){ Debug.log (login, FB...) via GS connecting.); New GameSparks.Api.requests.FacebookConnectRequest(). SetAccessToken (AccessToken.CurrentAccessToken.TokenString). SetDoNotLinkToKertPlayer (False). SetSwitifposible (true). Send ((fbauth_response) = > { (!fbauth_response. HasErrors){ ConnectionInfoField.text = Game Sparks certified with Facebook...; userNameField.text = fbauth_response. displayname; } Other { debug. logagg (fbauth_response. Errors. JSON); } }); } Other { Debug.LogWarning (Facebook login failed:+ Result. error; } }); } Other } FacebookConnect_bttn(); } } Once you connect this method to the Facebook login button, you should be able to process the login. When you try to login the first thing you will see is that you will be completed with the following panel: this is a Facebook prompt, which is used to test the login through the editor. It won't appear when the app is deployed to your device, but it lets you emulate logins from the editor while running the game from the desktop (remember that Facebook doesn't support desktop platforms): you'll need an access token to login. Here are the steps to follow: 1. Click on the Find Access Token button and you will be brought to your Facebook developer profile and you will be told that you need to allow your app to get an access token: 2. If you go and click on this option, you will be presented with the access token. You need to copy this token: and use it back in the Unity Editor in the User Access Token field: 3. Once you have the access token set up, you can complete the login using the Send Success button. The view must go through the Facebook login process and, once completed, it will use the Facebook access token to log into Game Sparks. Your user's Facebook profile is now linked to Game Sparks. In this tutorial we covered: how to set up a new Facebook app and integrate that app into your Game Sparks game. The basic method of authentication using the Game Sparks platform and Unity3D. How to link user's Facebook profile with a GameSparks account. Privacy. Site Terms. Cookie preferences. © Game Sparks Technologies Ltd. 2020 2020 </string>