B2C integration guide

This document is oriented to app and web developers to integrate FIFA users into their apps and websites. Is out of the scope of the present guideline to explain in deep the standard authentication/authorization protocols. It presents the basics to allow third parties to integrate their solutions and will point to official documentation for deeper details. Nevertheless some topics will be covered for better understanding.

Introduction

FIFA users are managed thru Azure Active Directory B2C. From integrators point of view it is possible to use all standard documentation provided by Microsoft.

FIFA Core Digital Platform (FDCP) provides API to be consumed from Apps and websites, some of the API are public and can be consumed anonymously and others can be consumed only by authenticated users. The API are compatible with user tokens and administrative tokens. Administrative tokens should be used by BackOffice. An App or Web developer should use always user tokens. In this document is will be explained both administrative and user tokens, but if you are an App developer you can skip all administrative points.

There are two main scenarios:

- App integration. Apps should use B2C OAuth authorization flow. The steps are:
  - Get an authorization code.
  - Get a token. To get this token is necessary to use the authorization code. The result also includes a refresh token.
  - Use the token. The token will be presented to the API to perform an operation that is secured.
  - Refresh the token. It is necessary to use the refresh token obtained in the second step. It also provides a new refresh token that can be used later again.
- Web integration. Webs should use B2C OAuth OpenID Connect flow. This flow is an extension of the previous one. The steps are:
  - Send an authentication request. In this step the user may indicate the scope for which needs authorization. This operation returns a JWT, the ID token.
  - Validate the ID token. The client must validate the ID token's signature and verify the claims in the token per your app's requirements. Azure AD B2C uses JSON Web Tokens (JWTs) and public key cryptography to sign tokens and verify that they are valid.
  - Get a token. Previous steps are necessary to authenticate the user. If it is necessary to access protected API, for instance to retrieve the user profile, it is necessary to get a token. The result also includes a refresh token.
  - Use the token. The token will be presented to the API to perform an operation that is secured.
  - Refresh the token. It is necessary to use the refresh token obtained in the second step. It also provides a new refresh token that can be used later again.
  - Sign-out. When you want to sign the user out of the app, it is not enough to clear your app's cookies or otherwise end the session with the user. You must also redirect the user to Azure AD to sign out. If you fail to do so, the user might be able to reauthenticate to your app without entering their credentials again. This is because they will have a valid single sign-on session with Azure AD.

Official B2C documentation includes detailed examples of both flows regardless the implementation technology or development language by detailing the HTTP requests to be performed. In addition also provides samples and standard libraries that implement previous flows only by providing the settings of the corresponding environment. It is recommended to use standard libraries when possible as the integration is simplier and are easier to update when there are new features or security fixes. The libraries can be consumed directly or it is possible to get the source code.

Integration samples:

- iOS Swift using MSAL: https://github.com/Azure-Samples/active-directory-b2c-ios-swift-native-msal
- iOS ObjC using App Auth: https://docs.microsoft.com/en-us/azure/active-directory-b2c/active-directory-b2c-devquickstarts-ios
- Android using MSAL: https://github.com/Azure-Samples/active-directory-b2c-android-native-msal
- Android using App Auth: https://docs.microsoft.com/en-us/azure/active-directory-b2c/active-directory-b2c-devquickstarts-android
- .Net: https://github.com/Azure-Samples/active-directory-b2c-dotnet-desktop
- Xamarin: https://github.com/Azure-Samples/active-directory-b2c-xamarin-native
- ASP.Net: https://docs.microsoft.com/en-us/azure/active-directory-b2c/active-directory-b2c-devquickstarts-web-dotnet-susi
- ASP.Net Core: https://github.com/Azure-Samples/active-directory-b2c-dotnetcore-webapp
- Node.js: https://docs.microsoft.com/en-us/azure/active-directory-b2c/active-directory-b2c-devquickstarts-web-node

All previous samples uses standard libraries from known repositories (eg NuGET) and can be updated automatically. If the solution to integrate uses different technology it is possible to follow the standard flow by executing the HTTP requests previous explained in B2C OAuth authorization flow and B2C OAuth OpenID Connect flow.

Helper libraries:
For .NET: Microsoft Authentication Library
For iOS: nxoauth2

How to integrate an App.

To integrate an App or Web with B2C is necessary to know the following parameters:

- **Tenant**: It represents the instance of B2C which hosts the users. FIFA has a tenant per environment (DEV, TEST, QA, PRE PROD and PROD).
- **Client Id**. Every App or site should be registered in a tenant to be able to use it. To integrate new Apps is necessary to contact the tenant administrator to obtain the client id and its secret.
- **Client Secret**: This secret will be also provided by the administrator. The client Id and the client secret are used to secure the communication between the client app and B2C.
- The **policy** or policies to be used in the App. A policy represents a specific flow for an App or web interacting with B2C. It is necessary to provide it as a parameter. Policy samples
  - Signup/SignIn policies
  - Edit profile.
  - Password reset.
- **Scope**: in some scenarios is necessary to provide the scope or scopes needed by an App. For example, some App require access to user's contact information while other only require access to core user data.

Here there are the parameter values for standard access (FIFA club).

Existing **B2C policy** to use in DEV, TEST, QA, PRE PROD & PRODUCTION Environments: **b2c_1a_fifa_signuporsignin**
Tenants

	Tenant
DEV Environment	fddevaadfans.onmicrosoft.com
TEST Environment	fdtestaadfans.onmicrosoft.com
QA Environment	fdqaaadfans.onmicrosoft.com
PRE PROD Environment	fdpreaadfans.onmicrosoft.com
PRODUCTION Environment	fdprdaadfans.onmicrosoft.com

Client Ids

	Android	iOS	Web
DEV Environment	12a7c526-b9d4-4993-847e-e90a6843f6ad	c289656e-7f68-4575-bcd7-b247bac03f18	e7f2152e-46e3-4f52-bae4-9d43b9114711
TEST Environment	fbb439cf-2076-4c7c-b1aa-bd1e71a39d10	ee98de0a-653b-46ff-9c5d-59e064df2b9b	082c3174-241c-4f9a-8e30-339bad462717
QA Environment	f57c8de2-a3b2-428e-b269-1277cf9cd156	8489e131-9177-4ee3-b920-c695946763aa	a9934bba-2a65-43a7-99b5-928d88226448
PRE PROD Environment	237195be-f89f-4581-b5d0-f2595c912e58	ef160527-4b37-4d19-9ceb-96564af1ecd2	b36f4b20-e1bc-46d8-ae62-d1f12ee00c72
PRODUCTION Environment	05a605bd-9f0b-4b4e-ac43-bc92d9353bbe	e70866aa-01e1-4088-8f1f-83f0862f17c5	64e9afa8-c5c0-413d-882b-bc9e6a81e264

Secrets

	Android	iOS	Web
DEV Environment			n^t67yJsx#l{AX)2
TEST Environment			%Du\2KED0$3\5o1{
QA Environment			AGMO75vm02M!,Ybk
PRE PROD Environment			m4NnGg}Ip0jIH05)
PRODUCTION Environment			?3L465,BaBGZ;So(

B2C Endpoints

	URL
DEV Environment	https://account-dev.fifa.com/ff86ecd8-ee93-473e-ae50-b59c66bbcd25/oauth2/v2.0/authorize

TEST Environment	https://account-test.fifa.com/c0c5789b-8537-4dfa-9b4d-28bdb24e050a/oauth2/v2.0/authorize
QA Environment	https://account-qa.fifa.com/eff51d9d-33ff-4811-8149-7698be1c56e6/oauth2/v2.0/authorize
PRE PROD Environment	https://account-pre.fifa.com/ab180054-8ef1-4762-9cde-824188b5cf03/oauth2/v2.0/authorize
PRODUCTION Environment	https://account.fifa.com/5a7baeb7-e706-4830-ad9f-103eba126311/oauth2/v2.0/authorize

B2C Tenant Ids

	Tenant Id
DEV Environment	ff86ecd8-ee93-473e-ae50-b59c66bbcd25
TEST Environment	c0c5789b-8537-4dfa-9b4d-28bdb24e050a
QA Environment	eff51d9d-33ff-4811-8149-7698be1c56e6
PRE PROD Environment	ab180054-8ef1-4762-9cde-824188b5cf03
PRODUCTION Environment	5a7baeb7-e706-4830-ad9f-103eba126311

B2C URL for the Web

```
https://{B2C_Endpoint}?
p={policy}
&client_Id={client_id}
&nonce={nonce}
&state={state}
&redirect_uri={redirect_uri}
&response_type=id_token
&response_mode={response_mode}
&prompt=login
```

Example TEST Environment: https://account-test.fifa.com/c0c5789b-8537-4dfa-9b4d-28bdb24e050a/oauth2/v2.0/authorize?p=b2c_1a_fifa_signupsignin&client_Id=082c3174-241c-4f9a-8e30-339bad462717&nonce=defaultNonce&redirect_uri=http%3A%2F%2Flocalhost&scope=openid&response_type=id_token&prompt=login&response_mode=query

Parameter	**Description**
response_mode	Specifies the method that should be used to send the resulting token back to your app. Can be one of query, form_post, or fragment.
state	A value included in the request that will also be returned in the token response. It can be a string of any content that you wish.
nonce	A value included in the request, generated by the app, that will be included in the resulting id_token as a claim. The app can then verify this value to mitigate token replay attacks. The value is typically a randomized, unique string that can be used to identify the origin of the request.
prompt	(optional) prompt=login will force the user to enter their credentials on that request, negating single-sign on.

Upon successful authentication, an id_token will be returned.

Web App B2C integration code samples

Demo Web App:

To use this demo is necessary to modify next parameters in web.config file:

```
<add key="ida:Tenant" value="Your tenant id"/>
<add key="ida:ClientId" value="Your Client Id"/>
<add key="ida:AadInstance" value="Your instance"/>
<add key="ida:RedirectUri" value="Your redirect url"/>
```

Mobile Apps B2C integration code samples

- Microsoft Authentication Library B2C Sample for Apple iOS in Swift
- Integrate Azure AD B2C into an Android App Using MSAL

- Notice for developers using Azure AD B2C tenants configured for Google sign-ins

Libraries for mobile apps' B2C integration:
MSAL library enables developers to build mobile apps that allow users to sign in using Azure AD B2C. MSAL supports adding authentication functionality to your .NET based client on Windows desktop (.NET 4.5+), UWP, .NET Core, Xamarin iOS and Xamarin Android. library name: microsoft.identity.client

- For .NET: Microsoft Authentication Library (MSAL) Preview for .NET, Windows Store, UWP, NetCore, Xamarin Android and iOS
- For iOS (Objective-C): Microsoft Authentication Library for iOS
- For Android: Microsoft Authentication Library (MSAL) for Android

Android Sample Steps, see: FIFA_Android_app.pdf

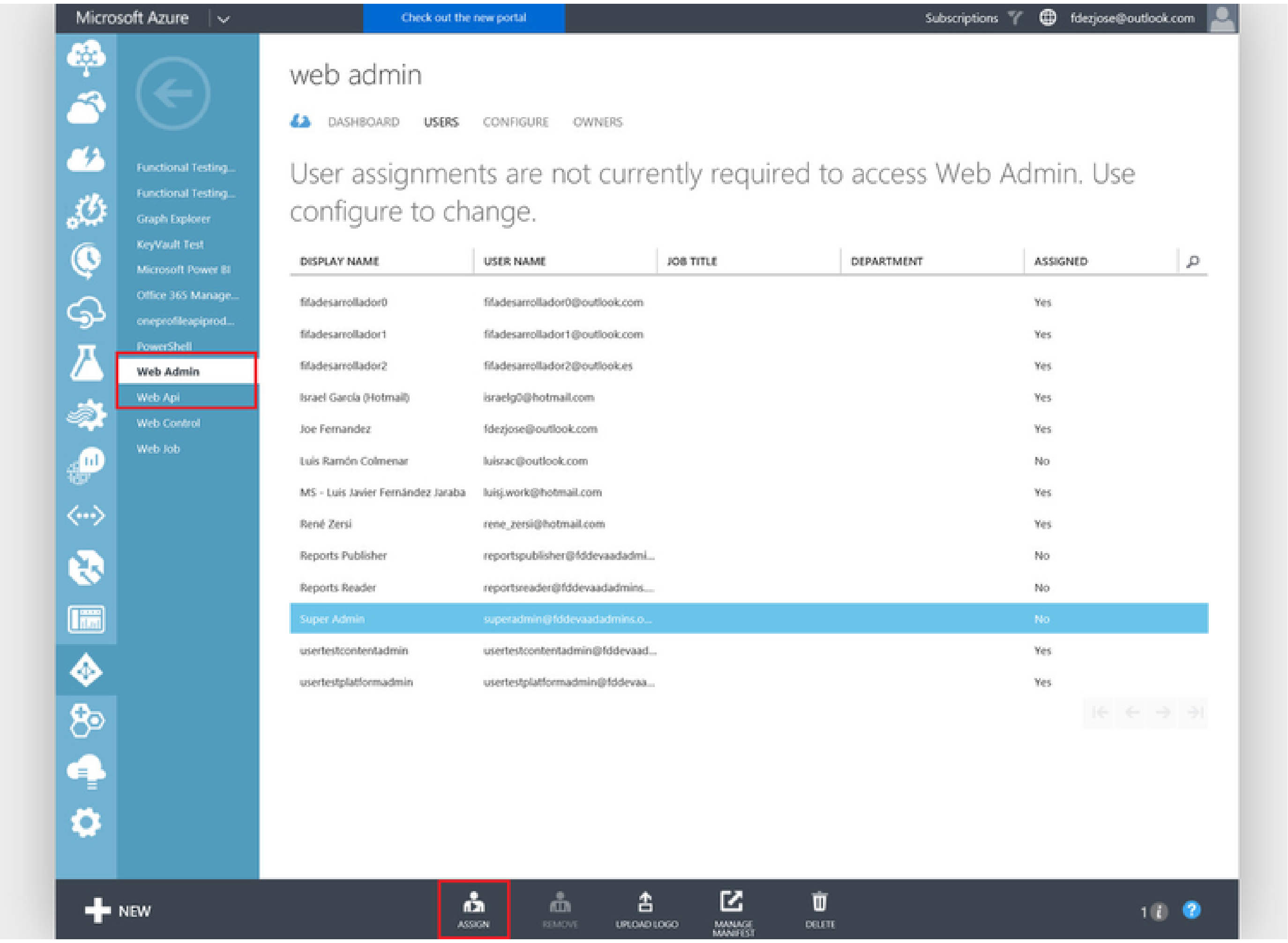
To get an **admin token**, the authentication is to be done against Azure Active Directory

Azure Active Directory Official Documentation
Azure Active Directory authorization flow

Azure Active Directory Code Samples
Tenants

	Tenant
DEV Environment	**fddevaadadmins.onmicrosoft.com**
TEST Environment	**fdtestaadadmins.onmicrosoft.com**
QA Environment	**fdqaaadadmins.onmicrosoft.com**

To grant a user account administrator privileges, the account has to be assigned Platform Content role on the Web Api and Web Admin on the admin tenants (fddevaadadmins, fdtestaadadmins, fdqaaadadmins). See the screenshot bellow:



web admin

User assignments are not currently required to access Web Admin. Use configure to change.

DISPLAY NAME	USER NAME	JOB TITLE	DEPARTMENT	ASSIGNED
fifadesarrollador0	fifadesarrollador0@outlook.com			Yes
fifadesarrollador1	fifadesarrollador1@outlook.com			Yes
fifadesarrollador2	fifadesarrollador2@outlook.es			Yes
Israel García (Hotmail)	israelg0@hotmail.com			Yes
Joe Fernandez	fdezjose@outlook.com			Yes
Luis Ramón Colmenar	luisrac@outlook.com			No
MS - Luis Javier Fernández Jaraba	luisj.work@hotmail.com			Yes
René Zersi	rene_zersi@hotmail.com			Yes
Reports Publisher	reportspublisher@fddevaadadmi...			No
Reports Reader	reportsreader@fddevaadadmins....			No
Super Admin	superadmin@fddevaadadmins.o...			No
usertestcontentadmin	usertestcontentadmin@fddevaad...			Yes
usertestplatformadmin	usertestplatformadmin@fddevaa...			Yes