B2C REST API

Hands-on Lab

Released:

Terms of Use

Microsoft Confidential - For Internal Use Only ***-*** Do Not Distribute Outside of Microsoft

© 2016 Microsoft Corporation. All rights reserved.

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see **Microsoft Copyright Permissions** at [**http://www.microsoft.com/permission**](http://www.microsoft.com/permission)

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The Microsoft company name and Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

**This document reflects current views and assumptions as of the date of development and is subject to change.  Actual and future results and trends may differ materially from any forward-looking statements.  Microsoft assumes no responsibility for errors or omissions in the materials.**

**THIS DOCUMENT IS FOR INFORMATIONAL AND TRAINING PURPOSES ONLY AND IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT.**

|  |  |
| --- | --- |
| About the Author | |
| **Author**: | **Ronny Bjones** |
| **Bio**: | https://www.linkedin.com/in/ronnybjones/ |

“About the Authors” is an important item as it establishes the credibility of the text by explaining how much experience the author team has in the field. Add as many “About the Authors” as needed from the Quick Parts Gallery. Change “Author” to a different title, such as “Subject Matter Expert” if that is more appropriate. Include SMEs who assisted with the development of the course. Add this to every Module.

Table of Contents

[Lab Overview 1](#_Toc451244538)

[Exercise 0: Setup 2](#_Toc451244539)

[Exercise 1: Run local account CRUD operations 4](#_Toc451244540)

[Exercise 2: Use custom attributes 5](#_Toc451244541)

# Lab Overview

Several paragraphs – or longer – describing an overview of the lab including a description of the lab and why certain topics are covered. This is also called the ‘LAB ABSTRACT’ that will be used for hand-off to conferences as part of the content hand-off process

###### Abstract

During this lab, you will run several exercises that will help you understand how to integrate REST API’s in Azure AD B2C.

###### Learning Objectives

After completing the exercises in this lab, you will learn:

* How a REST API can be added as a validation technical profile
* How a REST API can be added as a claims exchange

**Estimated time to complete this lab: *10* minutes**

# Exercise 0: Setup

Each exercise consists of a scenario and learning objectives, the scenario describes the purpose of the exercices, while the objectives are listed and have bullet points.

#### Scenario

You will need to complete the following walkthrough and get it running on your tenant.

See https://docs.microsoft.com/en-us/azure/active-directory-b2c/active-directory-b2c-get-started-custom.<https://docs.microsoft.com/en-us/azure/active-directory-b2c/active-directory-b2c-get-started-custom>

We are using the **SocialAndLocalAccounts** SocialAndLocalAccounts starter pack. You can use the policies published here: https://github.com/beejones/RestApiDemoAzureFunctions/tree/master/DemoPolicies

<https://docs.microsoft.com/en-us/azure/active-directory-b2c/active-directory-b2c-get-started-custom>These policies need to be customized with your tenant id, applications and facebook clientid and client secret.

<https://docs.microsoft.com/en-us/azure/active-directory-b2c/active-directory-b2c-get-started-custom>

# Exercise 1: Add AzureFunctions-ValidateClaims as Validation Technical Profile

Each exercise consists of a scenario and learning objectives, the scenario describes the purpose of the exercices, while the objectives are listed and have bullet points.

#### Scenario

Add the new claims provider ‘REST APIs’:

<ClaimsProvider>  
 <DisplayName>REST APIs</DisplayName>  
 <TechnicalProfiles>   
 <TechnicalProfile Id="AzureFunctions-ValidateClaims">  
 <DisplayName>Validate claims</DisplayName>  
 <Protocol Name="Proprietary" Handler="Web.TPEngine.Providers.RestfulProvider, Web.TPEngine, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" />  
 <Metadata>  
 <Item Key="ServiceUrl">https://restapidemoazurefunctions.azurewebsites.net/api/ValidateClaims?code=PG4IzZWG/JMinD/RgUf4US7M0Xl0DaQa0ULkqOI4b/IPT7WKjLYbAQ==</Item>  
 <Item Key="AuthenticationType">None</Item>  
 <Item Key="SendClaimsIn">Body</Item>  
 </Metadata>  
 <InputClaims>  
 <InputClaim ClaimTypeReferenceId="givenName"/>  
 <InputClaim ClaimTypeReferenceId="surName" />  
 </InputClaims>  
 <UseTechnicalProfileForSessionManagement ReferenceId="SM-Noop" />  
</TechnicalProfile>

</TechnicalProfiles>

</ClaimsProvider>

This REST API validates the input claims givenName and surName. The goal of this scenario is to validate whether the user belongs to an existing community. The implementation of the REST API just checks whether the givenname is equal ronny. This mimics the scenario that the user belongs to the community. In this case the user can progress to the next step in the user journey. If you enter a different givenname, an error message will be shown.

In order to integrate this validation into the user journey we need to add an addition claims provider which is extending LocalAccountSignUpWithLogonEmail with a validation technical profile responsible for calling the ValidateClaims REST API.

<ClaimsProvider>  
 <DisplayName>Local Account SignIn</DisplayName>  
 <TechnicalProfiles>  
 <TechnicalProfile Id="LocalAccountSignUpWithLogonEmail">  
 <ValidationTechnicalProfiles>  
 <ValidationTechnicalProfile ReferenceId="AzureFunctions-ValidateClaims" />  
 </ValidationTechnicalProfiles>  
 </TechnicalProfile>  
</TechnicalProfiles>  
</ClaimsProvider>

# Exercise 2: Add AddClaims REST API

Each exercise consists of a scenario and learning objectives, the scenario describes the purpose of the exercices, while the objectives are listed and have bullet points.

#### Scenario

We will add a new claims exchange to the user journey. This claims exchange will call the AddClaims REST API to collect additional claims about the user.

Add the following technical profile to the REST APIs claims provider.

<TechnicalProfile Id="AzureFunctions-AddClaims">  
 <DisplayName>Add claims</DisplayName>  
 <Protocol Name="Proprietary" Handler="Web.TPEngine.Providers.RestfulProvider, Web.TPEngine, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" />  
 <Metadata>  
 <Item Key="ServiceUrl">https://restapidemoazurefunctions.azurewebsites.net/api/AddClaims?code=1W/yRjI4dyQ6dIAoJ2Nhlj4QmjUhnFVqF9DaAyzYPh4aVxLCUSx72w==</Item>  
 <Item Key="AuthenticationType">Basic</Item>  
 <Item Key="SendClaimsIn">Body</Item>  
 </Metadata>  
 <CryptographicKeys>  
 <Key Id="BasicAuthenticationUsername" StorageReferenceId="B2C\_1A\_RestClientId" />  
 <Key Id="BasicAuthenticationPassword" StorageReferenceId="B2C\_1A\_RestClientSecret" />  
 </CryptographicKeys>  
 <InputClaims>  
 <InputClaim ClaimTypeReferenceId="givenName" />  
 <InputClaim ClaimTypeReferenceId="surName" />  
 </InputClaims>  
 <OutputClaims>  
 <OutputClaim ClaimTypeReferenceId="city" />  
 <OutputClaim ClaimTypeReferenceId="profession" />  
 </OutputClaims>  
 <UseTechnicalProfileForSessionManagement ReferenceId="SM-Noop" />  
</TechnicalProfile>

For this to work you also need the add the building blocks (see B2C\_1A\_TRUSTFRAMEWORKEXTENSIONS). These settings extend the claims schema with two new claims used as output claims of the new AddClaims REST API. These claims are city and profession.

This scenario implements basic authentication between the B2C service and the REST API. We need to configure keys used to authenticate.

Go to Policy Keys in the B2C extension on the azure portal and add two keys:

B2C\_1A\_RestClientId

B2C\_1A\_RestClientSecret

Both key are fixed. So do Add manual on the Policy keys blade and use these values:

12345678 for B2C\_1A\_RestClientId

Abcdefg for B2C\_1A\_RestClientSecret

Next add the SignUpOrSignInDemo user journey as found in B2C\_1A\_TRUSTFRAMEWORKEXTENSIONS. This adds an additional claims exchange which calls the AddClaims API to collect two new claims.

Next add the city and profession claims as output claims to the Relying Party in B2C\_1A\_SIGNUP\_SIGNIN. This makes sure that the city and profession claims are returned to the relying party in the signin and the signup scenario.