.

|  |  |
| --- | --- |
|  | Azure Active Directory B2C  Module 7 Lab – Adding a REST API |
|  |  |
|  |  |

Contents

[Overview 3](#_Toc458770156)

[Learning Objectives 3](#_Toc458770157)

[1. Add Wingtip REST API Claims Provider 3](#_Toc458770158)

# Overview

During this lab, you will create an advanced policy and add a REST API claims provider to the extensions policy.

**Estimated time to complete this lab: 45 minutes**.

# Learning Objectives

After completing the exercises in this lab, you will be able to:

* Add a REST API to your trust framework policy
* Add the necessary keys to authenticate to the REST API
* Learn how to use the recorder

## Add Wingtip REST API Claims Provider

|  |  |
| --- | --- |
| Task | Detailed Steps |
| Check if you have your keys for the tenant | 1. Run the **Windows Azure Active Directory PowerShell** Module as Administrator 2. Switch to the directory where ExploreAdmin is stored and enter the command:   Import-Module .\ExploreAdmin.dll   1. If you have not already done so in lab 5, run these commands now to generate the crypto keys:   New-CpimKeyContainer <tenant> TokenSigningKeyContainer key0 rsa 2048 0 0  New-CpimKeyContainer <tenant> TokenEncryptionKeyContainer key0 rsa 2048 0 0 |
| Upload crypto keys to the tenant | 1. Run the following PowerShell commands:   Set-CpimKeyContainer -Tenant <tenant> -StorageReferenceId B2cRestClientId -UnencodedAsciiKey 12345678  Set-CpimKeyContainer -Tenant <tenant> -StorageReferenceId B2cRestClientSecret -UnencodedAsciiKey abcdef  **Note:** This configures the tenant with preset values for the ClientId (12345678) and ClientSecret (abcdef) as basic authentication credentials. |
| Add claim type used by the REST API | 1. Edit the **Base\_Extensions** policy 2. Navigate to the **c:\B2C\B2CDemoTools-master\B2CPolicies\Final** folder (or wherever you have saved it) and open **b2ccharm.onmicrosoft.com\_B2C\_1A\_base\_Extensions.xml**   **Note:** To save you a lot of typing we are going to copy the code from here.   1. Copy the section **ClaimsSchema** under BuildingsBlocks - this defines the Loyalty claims used by the REST API 2. Paste this into your **Base\_Extensions** policy within the BuildingsBlocks node   **Note:** This adds a Loyalty Number text box to the schema. The regular expression is an interesting additional example, but serves no purpose here as we don’t ask the consumer to fill in the loyalty number (we did with eyecolor). |
| Add the REST API claims provider to Base\_Extensions.xml. | 1. Still in **Base\_Extensions**, uncomment the **ClaimsProviders** node 2. Back in the **b2ccharm.onmicrosoft.com\_B2C\_1A\_base\_Extensions.xml**, search for **Loyalty Rest** **API**, and copy the **ClaimsProvider** node (from the line above <ClaimsProvider> until </ClaimsProvider>) 3. Back in the **Base\_Extensions** policy, paste the copied text within the ClaimProviders node (that you just uncommented)   **Note:** We now have added a reference to the REST API to our policies. |
| Add REST API to the user journey | 1. Still in **Base\_Extensions**, uncomment the **UserJourneys** node 2. Open the **Base** policyand search for **<UserJourney Id="SignUpOrSignIn">** 3. Copy the entire **UserJourney** node (all 10 steps) 4. Back in the **Base\_Extensions** policy, paste the copied text within the UserJourneys node (that you just uncommented) 5. Rename the **userJourney** Id from SignUpOrSignIn to **SignUpOrSignInWithRestAndIdp** (we will be adding the IdP in the next lab) |
| Change the SignUpOrSignInWithRestAndIdp user journey to include the REST API | 1. Find **<OrchestrationStep Order="10" Type="SendClaims"** in the SignUpOrSignInWithRestAndIdp user journey, and change this step’s **Order** to **11** 2. Add the following additional **step 10** (before 11!):   <OrchestrationStep Order="10" Type="ClaimsExchange">  <ClaimsExchanges>  <ClaimsExchange Id="LoyaltyRestExchange"   TechnicalProfileReferenceId="LoyaltyRest" />  </ClaimsExchanges>  </OrchestrationStep>   1. Save and upload the **Base\_Extensions** policy to the portal |
| Create a SignupOrSigninWithRestAndIdp policy | 1. From your working folder, copy the **signuporsignin** policy and save it as **<Tenant>\_SignupOrSignInWithRestAndIdp.xml** 2. Change the **PolicyID** in the TrustFrameworkPolicy element to **signuporsigninwithrestandidp** 3. Change the **DefaultUserJourney** element to **SignupOrSigninWithRestAndIdp** 4. **Save** the file and **upload** it |
| Add the recorder to the policy | 1. Navigate to <http://guidgen.com> and copy a newly generated GUID (we will refer to this as your unique GUID) 2. Back in your **SignupOrSigninWithRestAndIdp** policy, add two attributes to the root **TrustFrameworkPolicy** element as follows:   DeploymentMode="Development"  UserJourneyRecorderEndpoint="[https://b2crecorder.azurewebsites.net/stream?id](https://na01.safelinks.protection.outlook.com/?url=https%3a%2f%2fb2crecorder.azurewebsites.net%2fstream%3fid&data=01%7c01%7cRonny.Bjones%40microsoft.com%7c8b6721f3c2e24b9a335c08d37ea82324%7c72f988bf86f141af91ab2d7cd011db47%7c1&sdata=iKOYf0GjkzrL93WfJxTMS5d%2buzym1y8i%2fg%2bYH1FzLNA%3d)=<your unique GUID>”  **Note:** You can get the unique name from the B2C portal by clicking B2C\_1A\_signuporsigninwithrestandidp. |
| Add an output claim | 1. Add an OutputClaim to the OutputClaims section at the end of the policy as follows:   <OutputClaim ClaimTypeReferenceID="Loyalty" />  **Note:** This is just so that you can see the value of the Loyalty Number in the token. You could also add an Output claim for eyecolor, to see that your attribute is still stored in the directory.   1. **Save** and **upload** the policy |
| Test the policy | 1. Select the policy and click **Run Now** 2. Click **sign up**   **Note:** You will of course need a new account, perhaps you have another email address you can use for a local account. If not maybe you can swap accounts with another student. When you complete the sign up (with a new account), the token will show you a value for the loyalty number in the form of a GUID.   1. Close the browser tab and click **Run Now** to execute the policy again, this time signing in with the account you just created   **Note:** You should see that the token includes the loyalty number with the same GUID as before. While a GUID may not be a very useful value for the purpose of loyalty discounts(!) all that we are trying to prove is that a value can be looked up and retrieved from a 3rd party provider using a REST API. |
| View your user journey in the recorder | 1. Navigate to **https://b2crecorder.azurewebsites.net/trace\_102.html?id=<your unique GUID>** and click **Refresh Now** to see the details of the user journey |