

Hello Patient Sample App

In this tutorial we will build a simple app using c# and show:

1. How to authenticate to Resolute FHIR instance using Azure AD
2. Retrieve Patient record

Connecting to Azure AD and obtaining token

Code below, acquires security token from Azure AD using an device code that the App displays on the screen.

```
AuthenticationContext authCtx = null;
authCtx = new
AuthenticationContext("https://login.microsoftonline.com/common");
AuthenticationResult result = null;
try{
    DeviceCodeResult codeResult = await
authCtx.AcquireDeviceCodeAsync(resource,clientId);
    Console.ResetColor();
    Console.WriteLine("You need to sign in.");
    Console.WriteLine($"Message: {codeResult.Message} \n");
    result = await authCtx.AcquireTokenByDeviceCodeAsync(codeResult);
}
catch (Exception ex)
{
    Console.ForegroundColor = ConsoleColor.Red;
    Console.WriteLine("Something went wrong.");
    Console.WriteLine($"Message: {ex.Message} \n");
}
return result;
```

Retrieving Patient record

To retrieve patient record, we created a function where we pass jwt token we obtained in step 1.

```
var httpClient = new HttpClient();
httpClient.DefaultRequestHeaders.Authorization = new
AuthenticationHeaderValue("Bearer",jwt);
try{
    var request = new HttpRequestMessage(HttpMethod.Get,ResoluteURI +
"/Patient");
    var response =
httpClient.SendAsync(request).GetAwaiter().GetResult();
    var content =
response.Content.ReadAsStringAsync().GetAwaiter().GetResult();
    if (!response.IsSuccessStatusCode)
```

```

        {
            throw new HttpRequestException($"{{{(int)response.StatusCode}}}
{content}");
        } else
        {
            Console.WriteLine(content.ToString());
        }
    }
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
}

```

Full source code for sample below is

```

using Microsoft.IdentityModel.Clients.ActiveDirectory;
using System;
using System.Threading.Tasks;
using System.Net.Http.Headers;
using System.Net.Http;

namespace FHIRSampleApp1
{
    class Program
    {
        // Resolute FHIR resource URI
        public const string resource =
"https://microsoft.onmicrosoft.com/resolute-test.cloud";
        // Application Id registered in Azure AD
        public const string clientId = "5b6bfa6c-c858-45a8-b2ec-
99d3c32ef7b0";
        // FHIR Service endpoint URI
        public const string ResoluteURI =
"https://resolutedemo.fhir.projectresolute.ms";
        string jwt;
        public static void Main(string[] args)
        {
            Console.WriteLine("Resolute FHIR Server - Hello Patient App");
            Console.WriteLine();
            Console.WriteLine("Connecting to Azure AD to get the token for
FHIR Service");

            Authenticate().Wait();
        }
        static async Task Authenticate()
        {
            AuthenticationResult ar = await GetToken();
            if (ar != null)
            {
                Console.WriteLine(ar.AccessToken.ToString());
                HelloPatient(ar.AccessToken.ToString());
            }
        }
    }
}

```

```

    }
    else {
        Console.WriteLine("Failed to acquire token.");
    }
}

static async Task<AuthenticationResult> GetToken()
{
    AuthenticationContext authCtx = null;
    authCtx = new
AuthenticationContext("https://login.microsoftonline.com/common");
    AuthenticationResult result = null;
    try{
        DeviceCodeResult codeResult = await
authCtx.AcquireDeviceCodeAsync(resource,clientId);
        Console.ResetColor();
        Console.WriteLine("You need to sign in.");
        Console.WriteLine($"Message: {codeResult.Message} \n");
        result = await
authCtx.AcquireTokenByDeviceCodeAsync(codeResult);
    }
    catch (Exception ex)
    {
        Console.ForegroundColor = ConsoleColor.Red;
        Console.WriteLine("Something went wrong.");
        Console.WriteLine($"Message: {ex.Message} \n");
    }
    return result;
}

static void HelloPatient(string jwt)
{
    Console.WriteLine();
    Console.WriteLine("Reading Patient records");

    var httpClient = new HttpClient();
    httpClient.DefaultRequestHeaders.Authorization = new
AuthenticationHeaderValue("Bearer",jwt);
    try{
        var request = new
HttpRequestMessage(HttpMethod.Get,ResoluteURI + "/Patient");
        var response =
httpClient.SendAsync(request).GetAwaiter().GetResult();
        var content =
response.Content.ReadAsStringAsync().GetAwaiter().GetResult();
        if (!response.IsSuccessStatusCode)
        {
            throw new HttpRequestException($"
{{{(int)response.StatusCode}} {content}");
        } else
        {
            Console.WriteLine(content.ToString());
        }
    }
    catch (Exception ex)

```

```
        {  
            Console.WriteLine(ex.Message);  
        }  
    }  
}
```