



Azure AD Integration With Angular And ASP.NET Core Web API



Umashanker Pandey

Updated date Aug 18, 2021

5.6k

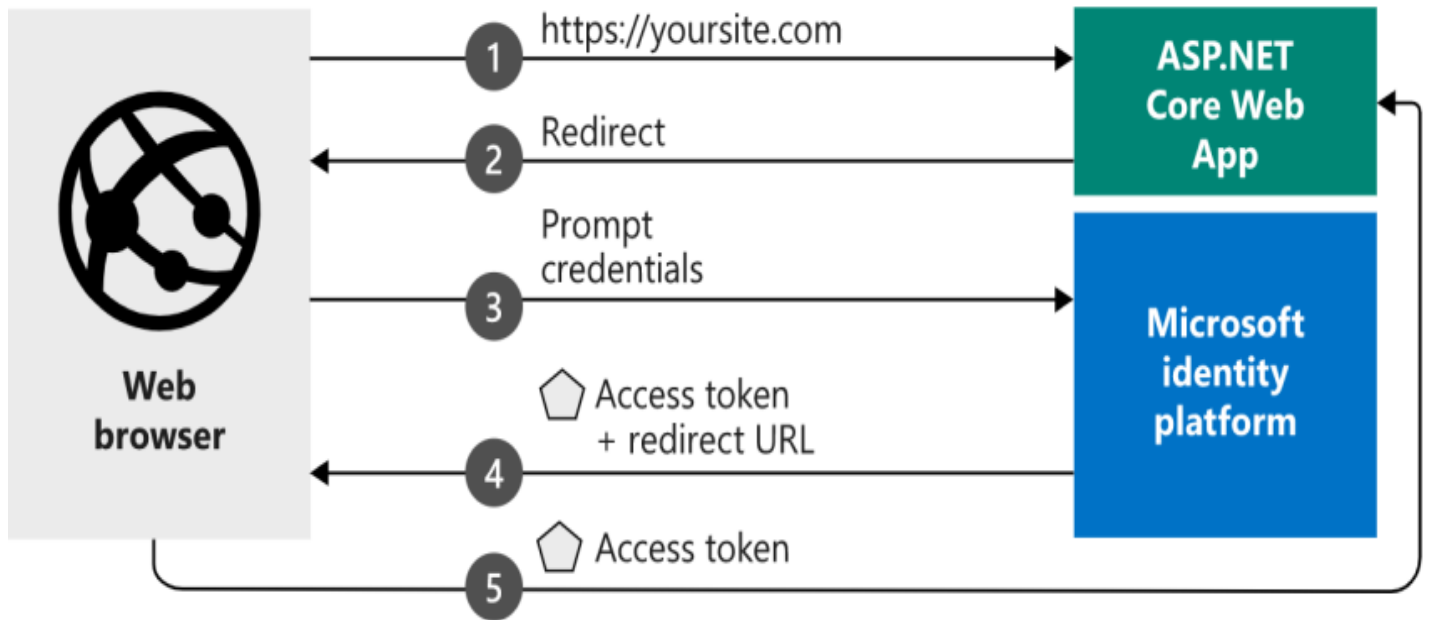
6

10

[Download Free .NET & JAVA Files API](#)[Try Free File Format APIs for Word/Excel/PDF](#)[AzureAdDemoApps.zip](#)

Introduction

This blog is going to showcase to you how we can implement Azure AD with web API and Angular application.

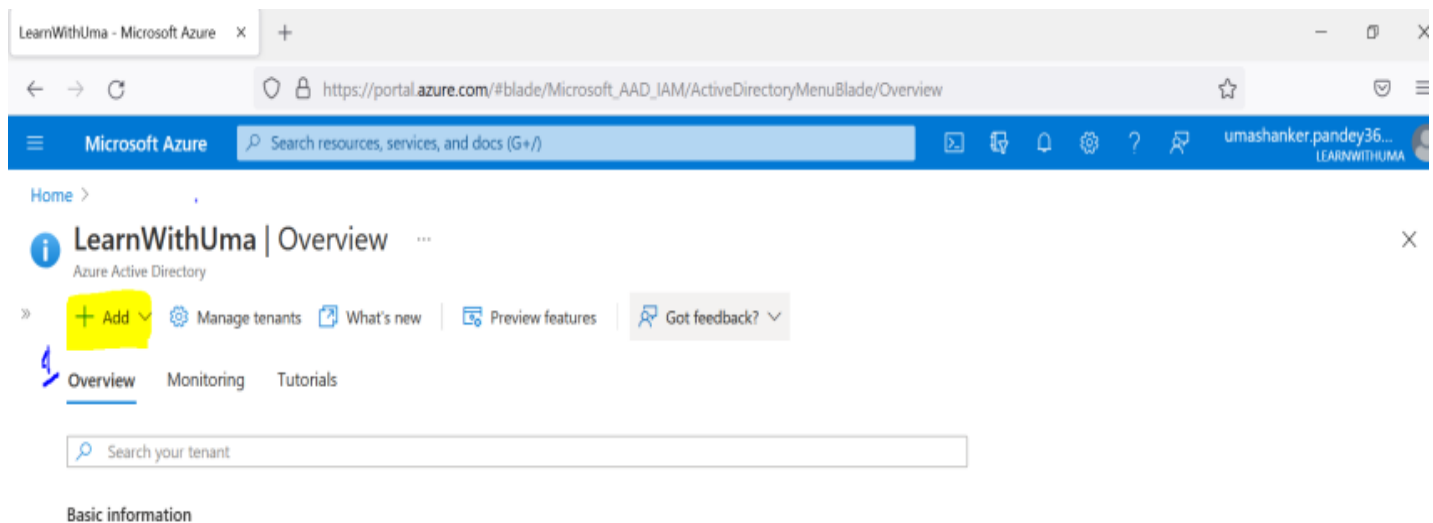


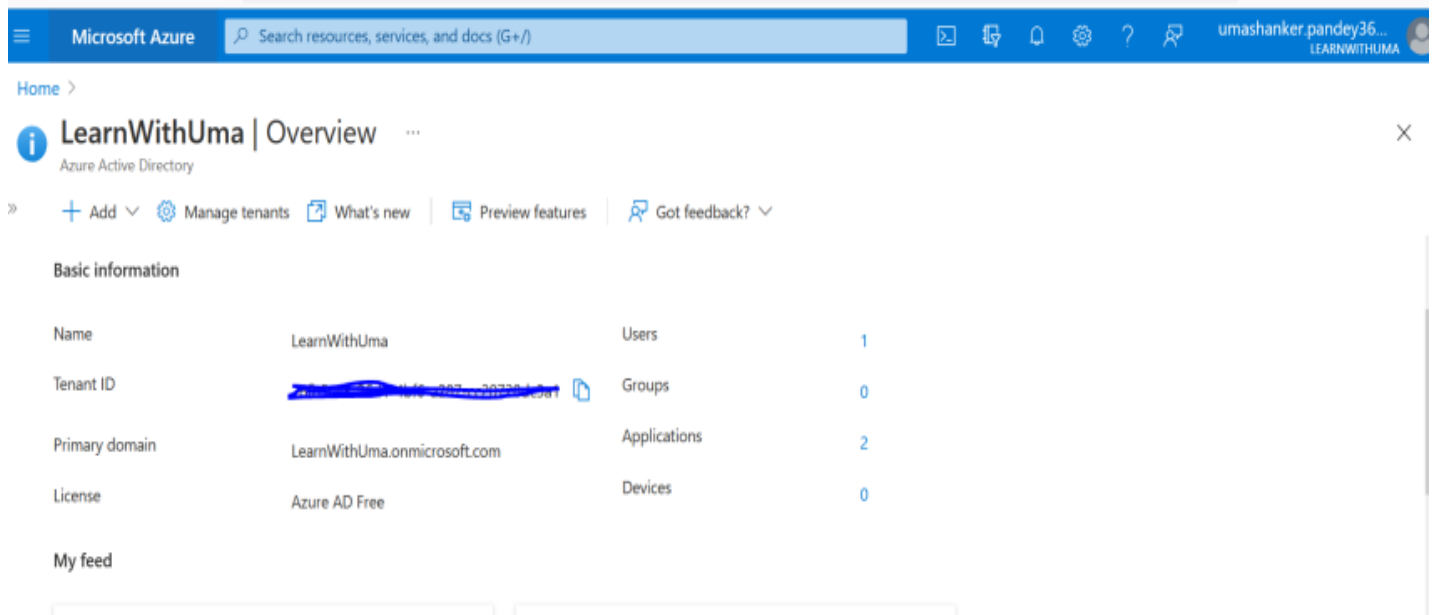
To connect azure ad with angular and asp.net core web API 5.0.

There are four-step processes to implement Azure AD in angular and asp.net core web API.

Step 1

Create Azure AD Account and Register the SPA(Single Page Application) application in Azure AD App Registration blade.





Microsoft Azure

Search resources, services, and docs (G+)

umashanker.pandey36...
LEARNWITHUMA

Home >

LearnWithUma | Overview

Azure Active Directory

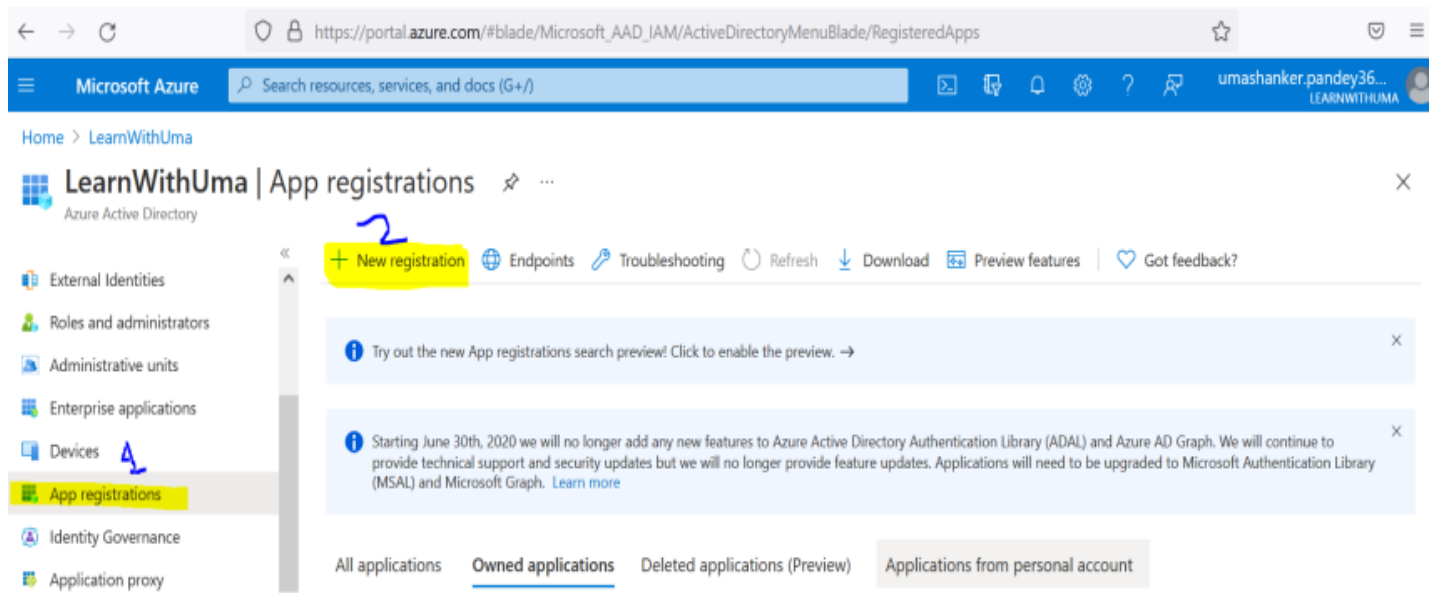
» + Add Manage tenants What's new Preview features Got feedback?

Basic information

Name	LearnWithUma	Users	1
Tenant ID	72f988bf-86f1-41af-91ab-2d7cd011db47	Groups	0
Primary domain	LearnWithUma.onmicrosoft.com	Applications	2
License	Azure AD Free	Devices	0

My feed

First, click on the App Registration button and then click on New Registration Button.



← → ↺ https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps

umashanker.pandey36...
LEARNWITHUMA

Home > LearnWithUma

LearnWithUma | App registrations

Azure Active Directory

« + New registration Endpoints Troubleshooting Refresh Download Preview features Got feedback?

Try out the new App registrations search preview! Click to enable the preview. →

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

All applications Owned applications Deleted applications (Preview) Applications from personal account

- External Identities
- Roles and administrators
- Administrative units
- Enterprise applications
- Devices
- App registrations**
- Identity Governance
- Application proxy

Fill in the Register Application Details.

← → ↻ https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps ☆

Microsoft Azure Search resources, services, and docs (G+/)

Home > LearnWithUma >

Register an application

* Name

The user-facing display name for this application (this can be changed later).

AzureDemoSPA ✓

Supported account types

Who can use this application or access this API?

☒ Accounts in this organizational directory only (LearnWithUma only - Single tenant)

☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

☐ Personal Microsoft accounts only

[Help me choose...](#)

← → ↻ https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps ☆

Microsoft Azure Search resources, services, and docs (G+/)

Home > LearnWithUma >

Register an application

☒ Accounts in this organizational directory only (LearnWithUma only - Single tenant)

☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

☐ Personal Microsoft accounts only

[Help me choose...](#)

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Single-page application (SPA) http://localhost:4200/ ✓

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

After clicking on the Register button. SPA application successfully gets registered and on the overview page, you get the register application details like Client ID, Tenant ID, etc.

The screenshot shows the Azure portal interface for the 'AzureDemoSPA' application. The left sidebar contains a search bar and a list of navigation items: Authentication, Certificates & secrets, Token configuration, API permissions, Expose an API, App roles, Owners, Roles and administrators | Preview. The main content area is titled 'AzureDemoSPA' and includes a search bar and action buttons: Delete, Endpoints, and Preview features. Under the 'Essentials' section, the following details are listed: Display name (AzureDemoSPA), Application (client) ID (redacted), Object ID (redacted), Directory (tenant) ID (redacted), and Supported account types (My organization only). On the right side, there are links for Client credentials (Add a certificate or secret), Redirect URIs (0 web, 1 spa, 0 public client), Application ID URI (Add an Application ID URI), and a note that the application is managed in the local directory (AzureDemoSPA).

Step 2

Register Web API applications in the same way. But in the API registration process, we do not need to provide the application redirect URL because our SPA Application is used to redirect the page.

The screenshot shows the 'Register an application' form in the Azure portal. The form has a title 'Register an application' and a subtitle 'The user-facing display name for this application (this can be changed later)'. The 'Name' field is filled with 'AzureAPIDemo' and has a green checkmark. Below this, the 'Supported account types' section asks 'Who can use this application or access this API?'. The first option, 'Accounts in this organizational directory only (LearnWithUma only - Single tenant)', is selected with a radio button. Other options include 'Accounts in any organizational directory (Any Azure AD directory - Multitenant)', 'Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)', and 'Personal Microsoft accounts only'. At the bottom, there is a checkbox for 'By proceeding, you agree to the Microsoft Platform Policies' and a green 'Register' button.

The screenshot shows the Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and the user profile 'umashanker.pandey36...'. The left sidebar contains a navigation menu with options like Overview, Quickstart, Integration assistant, and various 'Manage' options including Branding, Authentication, and API permissions. The main content area is titled 'AzureAdAPIDemo' and shows the 'Overview' tab. It displays essential information such as the Display name (AzureAdAPIDemo), Application (client) ID, Object ID, Directory (tenant) ID, and Supported account types (My organization only). There are also links for Client credentials, Redirect URIs, and Application ID URI. A notification banner at the bottom states that starting June 30th, 2020, new features for ADAL and Azure AD Graph will no longer be added, and applications need to be upgraded to MSAL and Microsoft Graph.

Then need to Expose An API,

The screenshot shows the 'Expose an API' page in the Azure portal. The top navigation bar and left sidebar are consistent with the previous screenshot. The main content area is titled 'AzureAdAPIDemo | Expose an API'. It shows the 'Application ID URI' and a section for 'Scopes defined by this API'. Below this, there is a table of scopes. The table has columns for 'Scopes', 'Who can consent', 'Admin consent display name', 'User consent display name', and 'State'. One scope is listed: 'https://a8c4baaf-d580-4212-b01a-123456789012/.default'. The 'Who can consent' column shows 'Authenticated users', 'Admin consent display name' is 'Admin API access', and the 'State' is 'Enabled'. Below the table, there is a section for 'Authorized client applications'.

The screenshot shows the Azure portal interface. On the left, the navigation pane is open, and 'Expose an API' is selected under the 'Manage' section. The main area displays the 'Add a scope' dialog for the API 'AzureAdAPIDemo'. The dialog has the following fields:

- Scope name ***: e.g. Files.Read
- Who can consent?**: Admins and users (selected), Admins only
- Admin consent display name ***: e.g. Read user files
- Admin consent description ***: e.g. Allows the app to read the signed-in user's files.
- User consent display name**: e.g. Read your files
- User consent description**

At the bottom of the dialog are 'Add scope' and 'Cancel' buttons.

After successfully exposing the API and scope has been added you need to go into the SPA application and add the permission for this API. Follow the process and click as per the number in below pic.

The screenshot shows the Azure portal interface for the 'AzureDemoSPA' application. The 'API permissions' section is open, and the 'Add a permission' button is highlighted. The 'Request API permissions' dialog is open, showing the 'Select an API' section. The 'My APIs' tab is selected, and the 'Add a permission' button is highlighted. The dialog shows the following table:

Name	Application (client) ID
AzureAdAPIDemo	12345678-1234-1234-1234-1234567890c4

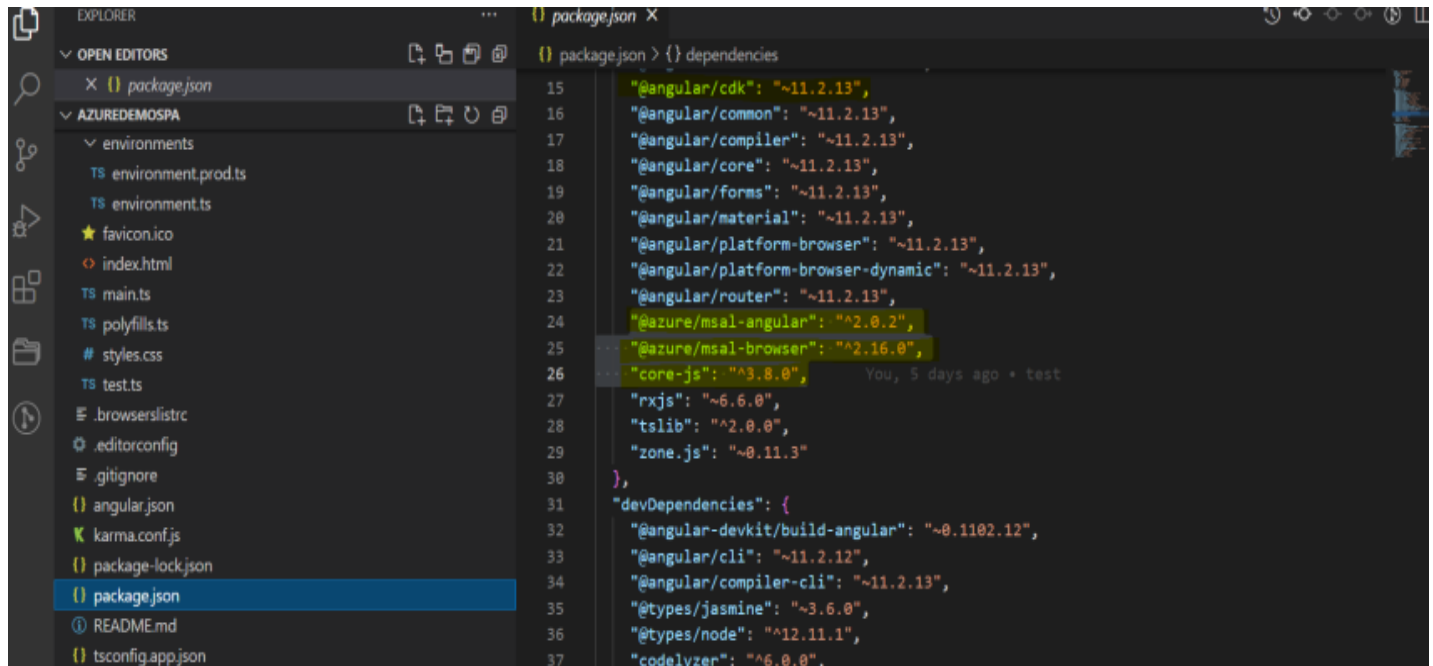
The dialog also shows the 'API / Permissions name' section with the following permissions:

- AzureAdAPIDemo (1)
 - AdminAccess
- Microsoft Graph (1)
 - User.Read

At the bottom of the dialog are 'Add a permission' and 'Cancel' buttons.

Step 3

Go to VS Code and Create New application in Angular, do the below changes. Add the below packages and do the npm install.



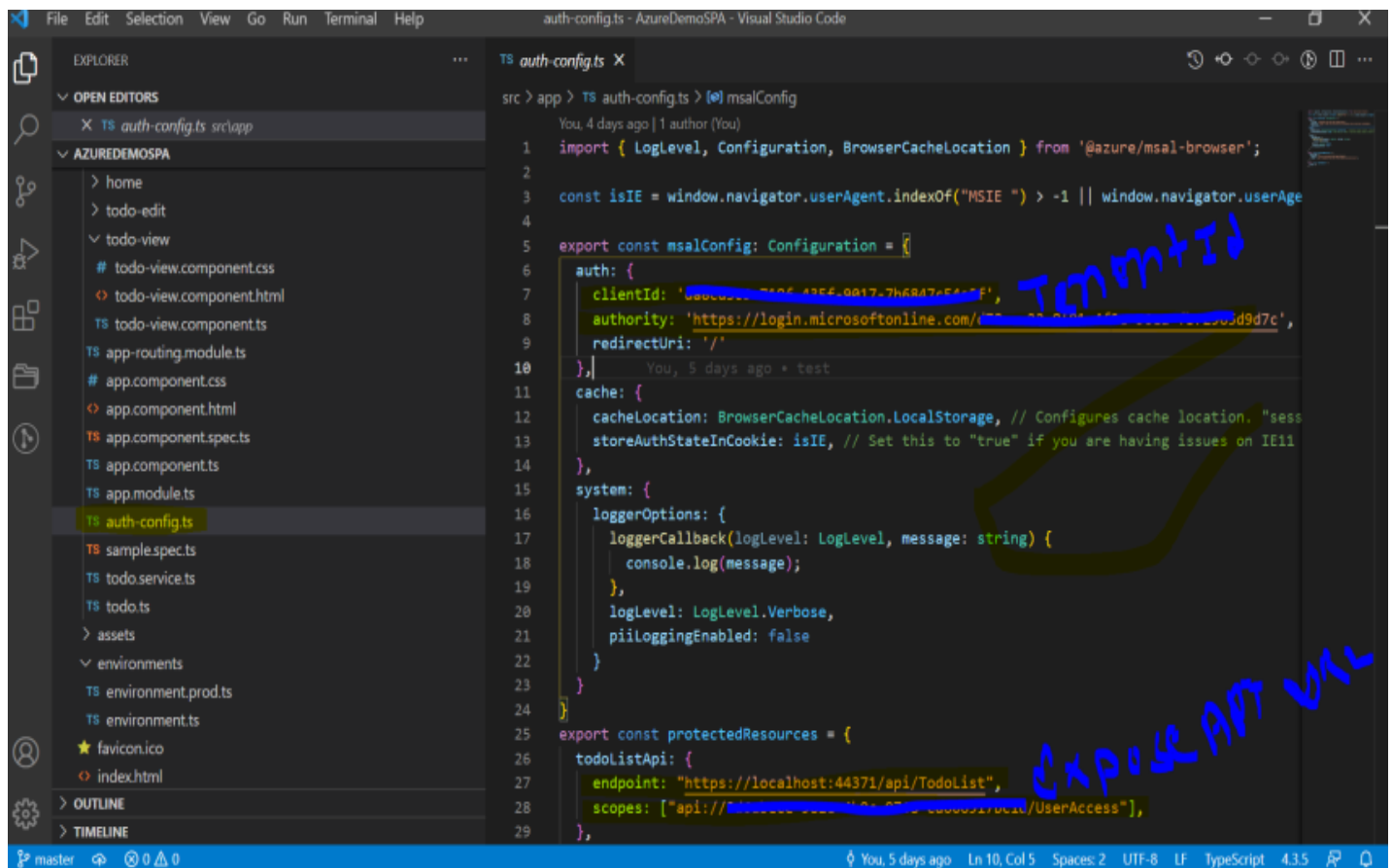
The screenshot shows the VS Code interface with the Explorer view on the left showing the file structure of a project named 'AZUREDEMOSPA'. The main editor displays the 'package.json' file, specifically the 'dependencies' section. The following packages are listed:

```

{
  "dependencies": {
    "@angular/cdk": "~11.2.13",
    "@angular/common": "~11.2.13",
    "@angular/compiler": "~11.2.13",
    "@angular/core": "~11.2.13",
    "@angular/forms": "~11.2.13",
    "@angular/material": "~11.2.13",
    "@angular/platform-browser": "~11.2.13",
    "@angular/platform-browser-dynamic": "~11.2.13",
    "@angular/router": "~11.2.13",
    "@azure/msal-angular": "^2.0.2",
    "@azure/msal-browser": "^2.16.0",
    "core-js": "^3.8.0",
    "rxjs": "~6.6.0",
    "tslib": "^2.0.0",
    "zone.js": "~0.11.3"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.1102.12",
    "@angular/cli": "~11.2.12",
    "@angular/compiler-cli": "~11.2.13",
    "@types/jasmine": "~3.6.0",
    "@types/node": "^12.11.1",
    "codelyzer": "~6.0.0",
  }
}

```

Add Auth-Config File and do the below settings as per the SPA application Registered.



The screenshot shows the VS Code interface with the Explorer view on the left showing the file structure of a project named 'AZUREDEMOSPA'. The main editor displays the 'auth-config.ts' file. The configuration is as follows:

```

import { LogLevel, Configuration, BrowserCacheLocation } from '@azure/msal-browser';

const isIE = window.navigator.userAgent.indexOf("MSIE ") > -1 || window.navigator.userAgent

export const msalConfig: Configuration = {
  auth: {
    clientId: 'XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX',
    authority: 'https://login.microsoftonline.com/XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX',
    redirectUri: '/'
  },
  cache: {
    cacheLocation: BrowserCacheLocation.LocalStorage, // Configures cache location. "sess
    storeAuthStateInCookie: isIE, // Set this to "true" if you are having issues on IE11
  },
  system: {
    loggerOptions: {
      loggerCallback(logLevel: LogLevel, message: string) {
        console.log(message);
      },
      logLevel: LogLevel.Verbose,
      piiLoggingEnabled: false
    }
  }
};

export const protectedResources = {
  todoListApi: {
    endpoint: "https://localhost:44371/api/TodoList",
    scopes: ["api://XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX/UserAccess"],
  },
};

```

Handwritten blue annotations are present on the image:

- TenantId** is written over the `clientId` and `authority` fields.
- Expose API URL** is written over the `endpoint` field in the `protectedResources` object.

Add below setting in App.Module.ts,

```
1  import { BrowserModule } from '@angular/platform-browser';
2  import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
3  import { NgModule } from '@angular/core';
4  import { FormsModule } from '@angular/forms';
5  import { AppRoutingModule } from './app-routing.module';
6  import { AppComponent } from './app.component';
7  import { HomeComponent } from './home/home.component';
8  import { TodoEditComponent } from './todo-edit/todo-edit.component';
9  import { TodoViewComponent } from './todo-view/todo-view.component';
10 import { TodoService } from './todo.service';
11 import { HTTP_INTERCEPTORS, HttpClientModule } from '@angular/common/http';
12 import { IPublicClientApplication, PublicClientApplication, InteractionType };
13 import { MsalGuard, MsalInterceptor, MsalBroadcastService, MsalInterceptorConfiguration };
14 import { msalConfig, loginRequest, protectedResources } from './auth-config';
15 export function MSALInstanceFactory(): IPublicClientApplication {
16     return new PublicClientApplication(msalConfig);
17 }
18 export function MSALInterceptorConfigFactory(): MsalInterceptorConfiguration {
19     const protectedResourceMap = new Map < string,
20         Array < string >> ();
21     protectedResourceMap.set(protectedResources.todoListApi.endpoint, protectedResources.todoListApi.allowedScopes);
22     return {
23         interactionType: InteractionType.Redirect,
24         protectedResourceMap
25     };
26 }
27 export function MSALGuardConfigFactory(): MsalGuardConfiguration {
28     return {
29         interactionType: InteractionType.Redirect,
30         authRequest: loginRequest
31     };
32 }
33 @NgModule({
34     declarations: [
35         AppComponent,
36         HomeComponent,
37         TodoViewComponent,
38         TodoEditComponent
39     ],
```

```
40     imports: [  
41         BrowserModule,  
42         BrowserAnimationsModule,  
43         AppRoutingModule,  
44         HttpClientModule,  
45         FormsModule,  
46         MsalModule  
47     ],  
48     providers: [{  
49         provide: HTTP_INTERCEPTORS,  
50         useClass: MsalInterceptor,  
51         multi: true  
52     }, {  
53         provide: MSAL_INSTANCE,  
54         useFactory: MSALInstanceFactory  
55     }, {  
56         provide: MSAL_GUARD_CONFIG,  
57         useFactory: MSALGuardConfigFactory  
58     }, {  
59         provide: MSAL_INTERCEPTOR_CONFIG,  
60         useFactory: MSALInterceptorConfigFactory  
61     },  
62     MsalService,  
63     MsalGuard,  
64     MsalBroadcastService,  
65     TodoService  
66 ],  
67     bootstrap: [AppComponent, MsalRedirectComponent]  
68 })  
69 export class AppModule {}
```

App.Component Settings,

```
1  import { Component, OnInit, Inject, OnDestroy } from '@angular/core';  
2  import { MsalService, MsalBroadcastService, MSAL_GUARD_CONFIG, MsalGuardCo  
3  import { AuthenticationResult, InteractionStatus, InteractionType, PopupRe  
4  import { Subject } from 'rxjs';  
5  import { filter, takeUntil } from 'rxjs/operators';  
6  @Component({  
7      selector: 'app-root',
```

```
8     templateUrl: './app.component.html',
9     styleUrls: ['./app.component.css']
10 })
11 export class AppComponent implements OnInit, OnDestroy {
12     title = 'Azure Ad Testing Demo';
13     isIframe = false;
14     loginDisplay = false;
15     private readonly _destroying$ = new Subject < void > ();
16     constructor(@Inject(MSAL_GUARD_CONFIG) private msalGuardConfig: MsalGu
17     ngOnInit(): void {
18         this.isIframe = window !== window.parent && !window.opener;
19         /**
20          * You can subscribe to MSAL events as shown below. For more info,
21          * visit: https://github.com/AzureAD/microsoft-authentication-libr
22          */
23         this.msalBroadcastService.inProgress$.pipe(filter((status: Interac
24             this.setLoginDisplay();
25         });
26     }
27     setLoginDisplay() {
28         this.loginDisplay = this.authService.instance.getAllAccounts().ler
29     }
30     login() {
31         if (this.msalGuardConfig.interactionType === InteractionType.Popu
32             if (this.msalGuardConfig.authRequest) {
33                 this.authService.loginPopup({
34                     ...this.msalGuardConfig.authRequest
35                 }
36                 as PopupRequest).subscribe((response: AuthenticationRe
37                 this.authService.instance.setActiveAccount(response.ac
38             ));
39             } else {
40                 this.authService.loginPopup().subscribe((response: Authent
41                 this.authService.instance.setActiveAccount(response.ac
42             ));
43         }
44     } else {
45         if (this.msalGuardConfig.authRequest) {
46             this.authService.loginRedirect({
47                 ...this.msalGuardConfig.authRequest
48             }
```

```
49         as RedirectRequest);
50     } else {
51         this.authService.loginRedirect();
52     }
53 }
54 }
55 login() {
56     this.authService.logout();
57 }
58 // unsubscribe to events when component is destroyed
59 ngOnDestroy(): void {
60     this._destroying$.next(undefined);
61     this._destroying$.complete();
62 }
63 }
```

app.component.html settings,

```
1 <mat-toolbar color="primary">
2   <a class="title" href="/">{{ title }}</a>
3   <div class="toolbar-spacer"></div>
4   <a mat-button [routerLink]="['todo-view']">TodoList</a>
5   <button mat-raised-button *ngIf="!loginDisplay" (click)="login()">Login<
6   <button mat-raised-button color="accent" *ngIf="loginDisplay" (click)="l
7 </mat-toolbar>
8 <div class="container">
9   <!--This is to avoid reload during acquireTokenSilent() because of hidde
10   <router-outlet *ngIf="!isIframe"></router-outlet>
11 </div>
12 <footer *ngIf="loginDisplay">
13   <mat-toolbar>
14     <div class="footer-text"> How did we do? </div>
15   </mat-toolbar>
16 </footer>
```

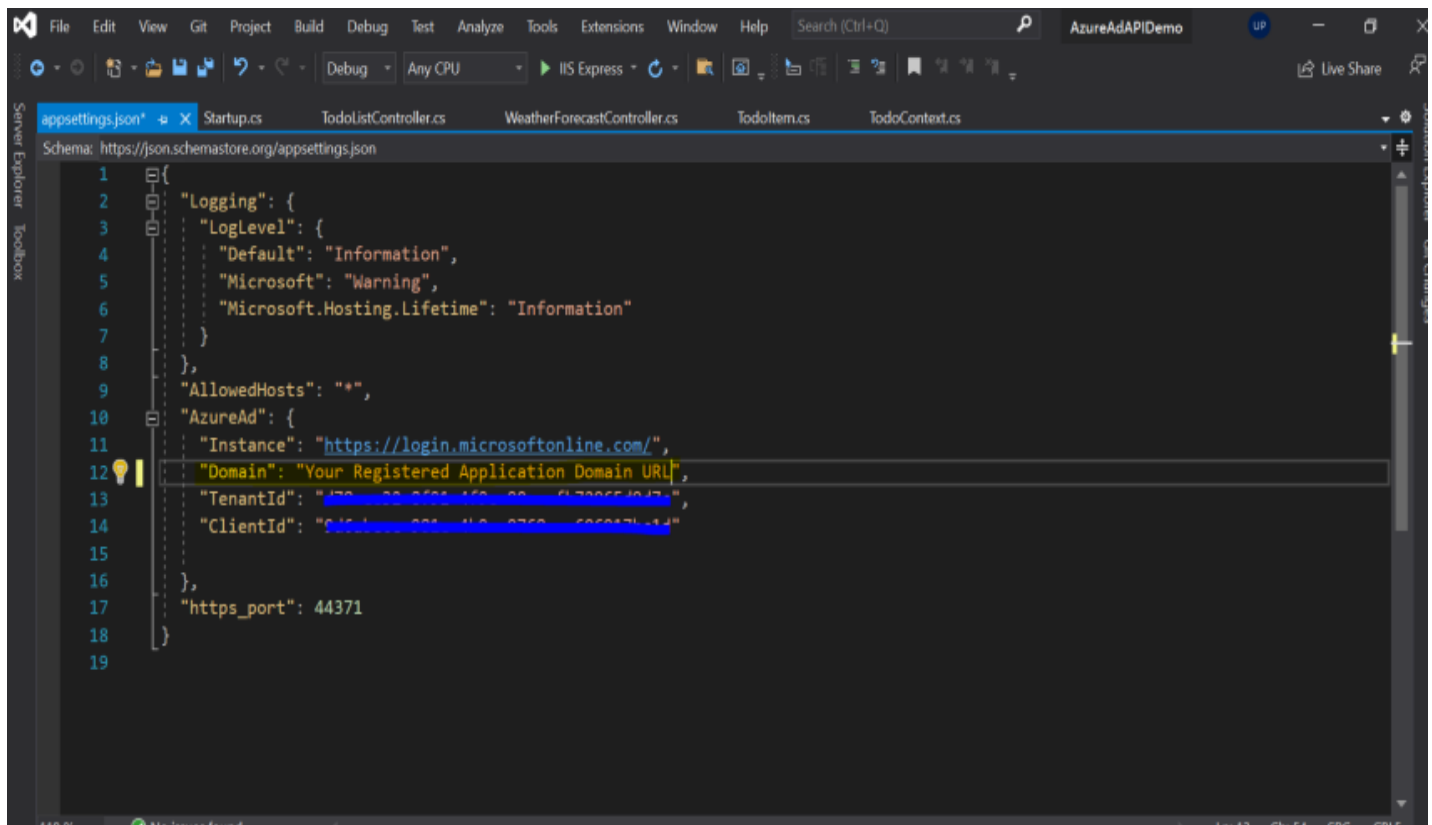
Index.html settings,

```
1 <!doctype html>
2 <html lang="en">
3   <head>
```

```
4 <meta charset="utf-8">
5 <title>AzureDemoSPA</title>
6 <base href="/">
7 <meta name="viewport" content="width=device-width, initial-scale=1">
8 <link rel="icon" type="image/x-icon" href="favicon.svg">
9 <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500">
10 <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
11 </head>
12 <body>
13 <app-root></app-root>
14 <app-redirect></app-redirect>
15 </body>
16 </html>
```

Step 4 - Web API settings

In AppSettings.json file add your Registered API tenant and Client ID and Domain name information.



From Nuget Package manager add microsoft.identity.web package and do the changes in startup's file.

```
1  using AzureAdAPIDemo.Models;
2  using Microsoft.AspNetCore.Authentication.JwtBearer;
3  using Microsoft.AspNetCore.Builder;
4  using Microsoft.AspNetCore.Hosting;
5  using Microsoft.AspNetCore.HttpsPolicy;
6  using Microsoft.AspNetCore.Mvc;
7  using Microsoft.EntityFrameworkCore;
8  using Microsoft.Extensions.Configuration;
9  using Microsoft.Extensions.DependencyInjection;
10 using Microsoft.Extensions.Hosting;
11 using Microsoft.Extensions.Logging;
12 using Microsoft.Identity.Web;
13 using Microsoft.OpenApi.Models;
14 using System;
15 using System.Collections.Generic;
16 using System.Linq;
17 using System.Threading.Tasks;
18 namespace AzureAdAPIDemo {
19     public class Startup {
20         public Startup(IConfiguration configuration) {
21             Configuration = configuration;
22         }
23         public IConfiguration Configuration {
24             get;
25         }
26         // This method gets called by the runtime. Use this method to add
27         public void ConfigureServices(IServiceCollection services) {
28             // Setting configuration for protected web api
29             services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme);
30             // Creating policies that wraps the authorization requirements
31             services.AddAuthorization();
32             services.AddDbContext < TodoContext > (opt => opt.UseInMemoryDatabase());
33             services.AddControllers();
34             // Allowing CORS for all domains and methods for the purpose of development
35             // In production, modify this with the actual domains you want
36             services.AddCors(o => o.AddPolicy("default", builder => {
37                 builder.AllowAnyOrigin().AllowAnyMethod().AllowAnyHeader();
38             }));
39             services.AddSwaggerGen(c => {
40                 c.SwaggerDoc("v1", new OpenApiInfo {
41                     Title = "AzureAdAPIDemo", Version = "v1"
```

```

42         });
43     });
44 }
45 // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
46 public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
47 {
48     if (env.IsDevelopment())
49     {
50         app.UseDeveloperExceptionPage();
51         app.UseSwagger();
52         app.UseSwaggerUI(c => c.SwaggerEndpoint("/swagger/v1/swagger.json", "My API V1"));
53     }
54     app.UseCors("default");
55     app.UseHttpsRedirection();
56     app.UseRouting();
57     app.UseAuthentication();
58     app.UseAuthorization();
59     app.UseEndpoints(endpoints => {
60         endpoints.MapControllers();
61     });
62 }

```

Then use [Authorize] on any controller.

```

{
    [Authorize]
    [Route("api/[controller]")]
    [ApiController]

    1 reference
    public class TodoListController : ControllerBase
    {
        // The Web API will only accept tokens 1) for users, and 2) having the access_as_user scope for this API
        static readonly string[] scopeRequiredByApi = new string[] { "access_as_user" };

        private readonly TodoContext _context;

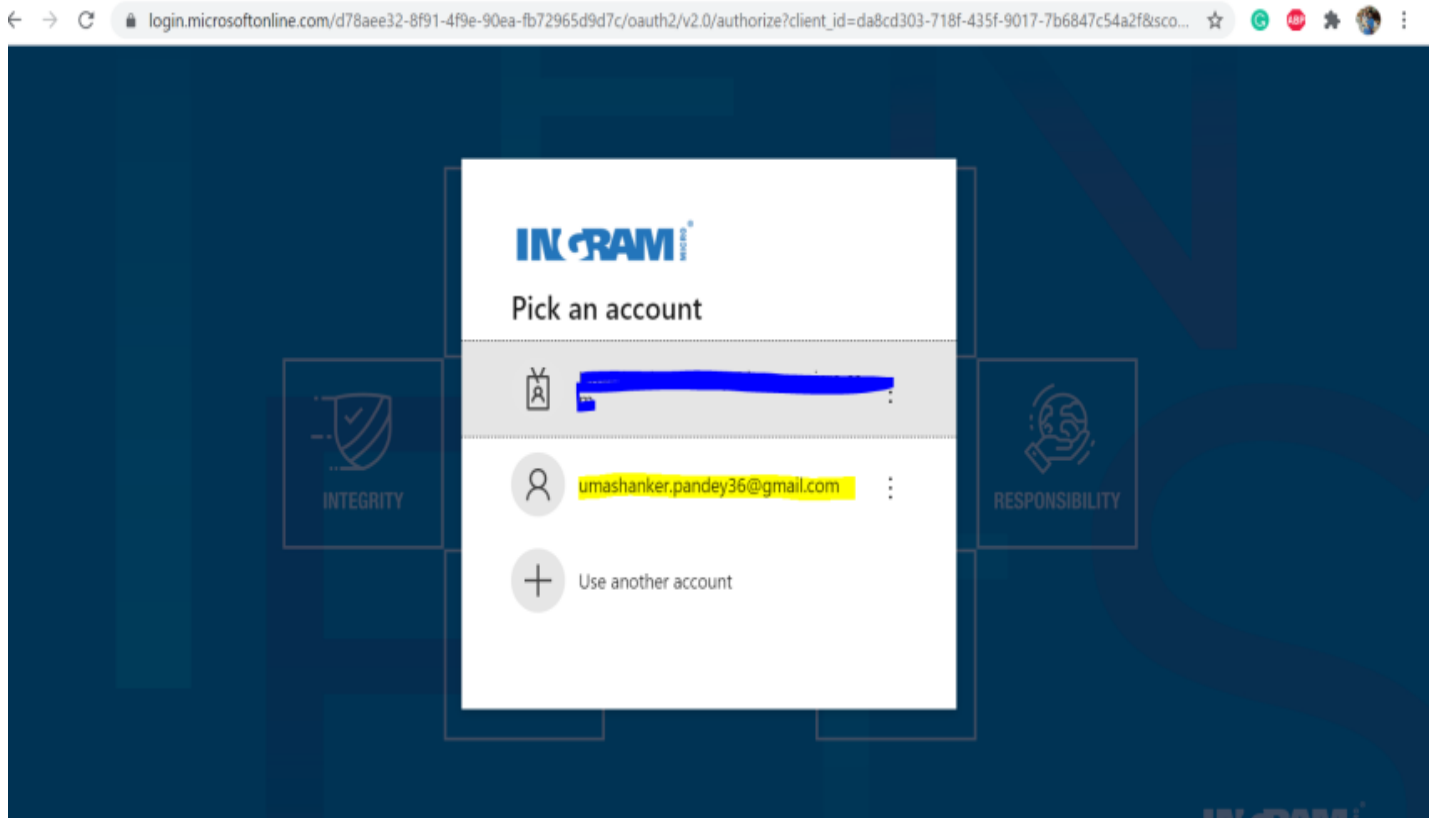
        0 references
        public TodoListController(TodoContext context)
        {
            _context = context;
        }
    }
}

```

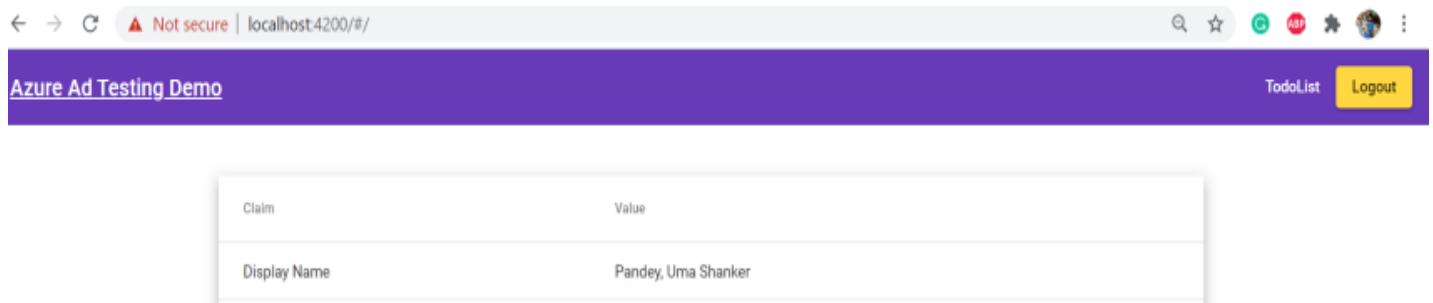
no issues found



After clicking the login button,



Select the account and log in successfully.



Summary

Follow the above steps and secure your solutions.

.NET 5.0

Angular

Azure AD

Next Recommended Reading

[Serialization In Web API With ASP.NET Core](#)**Umashanker Pandey**

Having more than 9+ years of experience in software development using different tech stacks. .Net Core | Angular | C# | Python | Azure/AWS Cloud | Docker | Kubernetes | Azure DevOps | SQL Server | Mongo db | AI/ML

<https://www.linkedin.com/in/umashanker-pandey-a2552b60/>

1963

13.2k

10

6



Type your comment here and press Enter Key (Minimum 10 characters)

Follow Comments



Thanks for posting this, it worked!

[Timeware Developers](#)

2052 5 0

0

Oct 06, 2021

0 Reply



Thanks. very nice.

[Vitthal Shinde](#)

2026 31 0

0

Oct 04, 2021

0 Reply



Far Better then Microsoft Docs.

[MANAS SANTRA](#)

2048 9 0

1

Sep 15, 2021

0 Reply



Very detailed information. Thanks for posting this. I am sure it will help both beginners and advance level developers.

[Amiya Padhi](#)

2053 4 0

2

Aug 26, 2021

0 Reply



Nicely explained

[AKASH CHAVAN](#)

1843 214 0

1

Aug 18, 2021

1 Reply



Thanks, Akash for your feedback.

[Umashanker Pandey](#)

1963 94 13.2k

Aug 18, 2021

LATEST BLOGS

How to become an author

Basic Knowledge On Generics

Azure - New Restrictions To Azurewebsites.net Domain

Difference Between Azure SQL Geo Replication and Azure SQL Failover

Generate User Mapping For Sharegate Migration Using PnP PowerShell

Creating SPO Site Using Powershell

Referencing File Names In Power Automate

Programmatically Fetching Files/Folders From Azure Files

Adding Multiple Users As Site Collection Admin

Dynamic Bar Chart For SharePoint List Using ChartJS

[View All](#) 

[About Us](#) [Contact Us](#) [Privacy Policy](#) [Terms](#) [Media Kit](#) [Sitemap](#) [Report a Bug](#) [FAQ](#) [Partners](#)
[C# Tutorials](#) [Common Interview Questions](#) [Stories](#) [Consultants](#) [Ideas](#) [Certifications](#)

©2021 C# Corner. All contents are copyright of their authors.