PortalGuard Tailored Authentication Documentation

Date: 2021-11-02
Title: Custom Self-Registration

Objective

Pacific College is establishing an automated procedure for new student onboarding. It involves Salesforce, Campus Vue and PortalGuard. Pacific College needs support for a new entry point into PortalGuard's existing Self Registration feature to ensure only registered new students can create their own username and password in Active Directory. This TA defines the changes required in PortalGuard to support this automated Self-Registration flow.

Installation

The upgrade kit with this documentation is meant for use on servers which already have PortalGuard installed. The updated files are intended for use on PortalGuard version **6.5.2.2**. The steps to upgrade and configure the new feature are as follows:

Update Steps

1. Copy this full, original zip file to the PortalGuard Server
2. Ensure the zip file is not "Blocked" before extracting it on the PG server (otherwise ALL files it contains will be marked as "Blocked")
3. All the UI files in the “InetPub” folder of the kit should be copied into the existing "InetPub\PortalGuard" folder as follows:

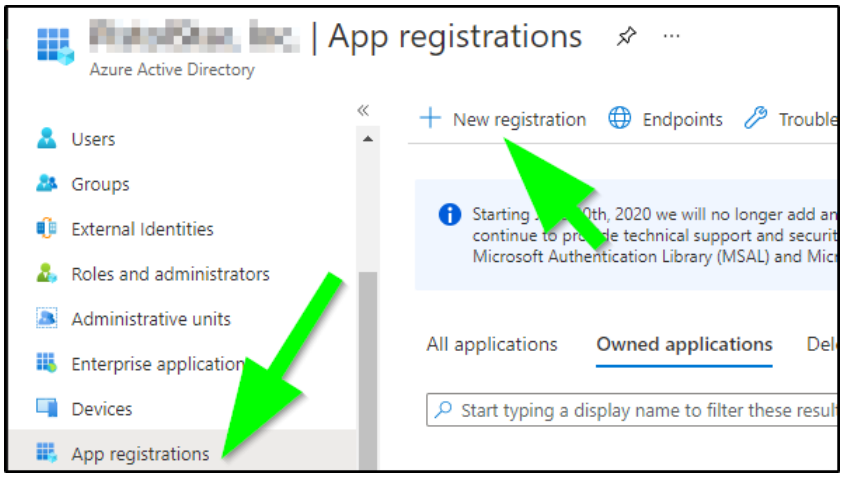
File	Folder
web.config	(root)
Pacific-SelfReg-TA.ashx.cs	App_Code
register.ashx.cs	App_Code
Newtonsoft.Json.dll	bin
Pacific-SelfReg-TA.ashx	_layouts\PG
register.aspx	_layouts\PG
pg_selfreg.js	_layouts\images\PG\js

4. From the “Policies” folder of the kit, copy the two HTML files to the “\Program Files\PistolStar\PortalGuard\Policies” folder.
5. From the “SQL” folder of the kit, run the ***PacificCollegeTA.sql*** script in Microsoft SQL Server Management Studio against PortalGuard’s SQL server instance

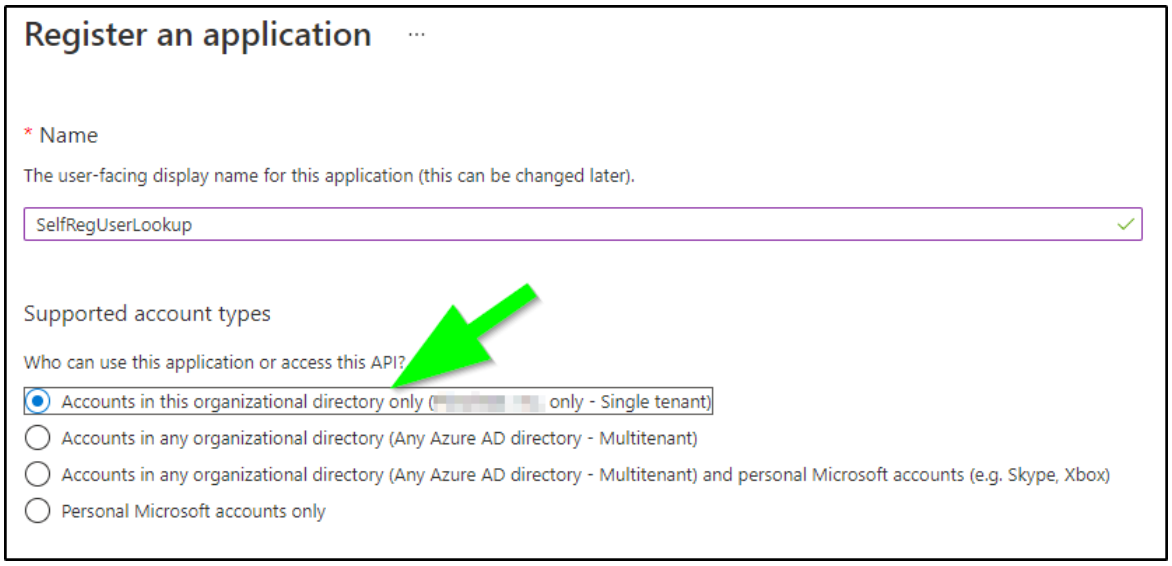
Azure AD App Registration

In order to communicate with Azure AD and ensure chosen usernames do not already exist in Azure AD, a new App Registration must be created. The steps are:

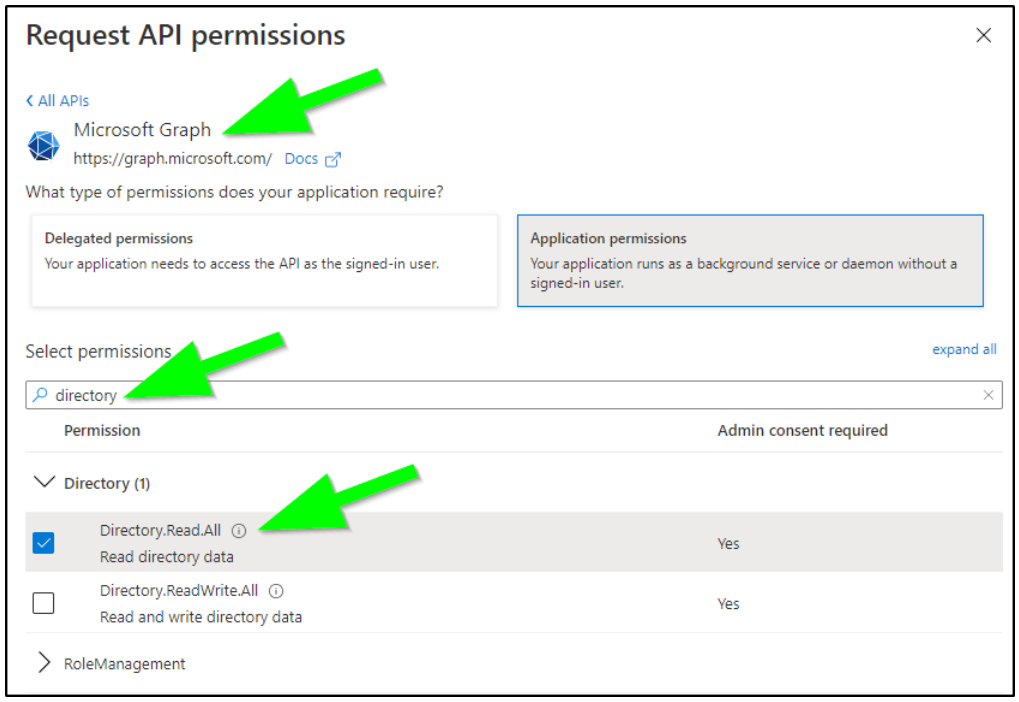
1) Log into portal.azure.com as an administrative user
2) Open the "Azure Active Directory" applet/panel
3) Under "App Registrations" on the right, click the "New registration" button:



4) Enter a new Name, e. g. "SelfRegUserLookup" and choose the top "Single tenant" radio button:

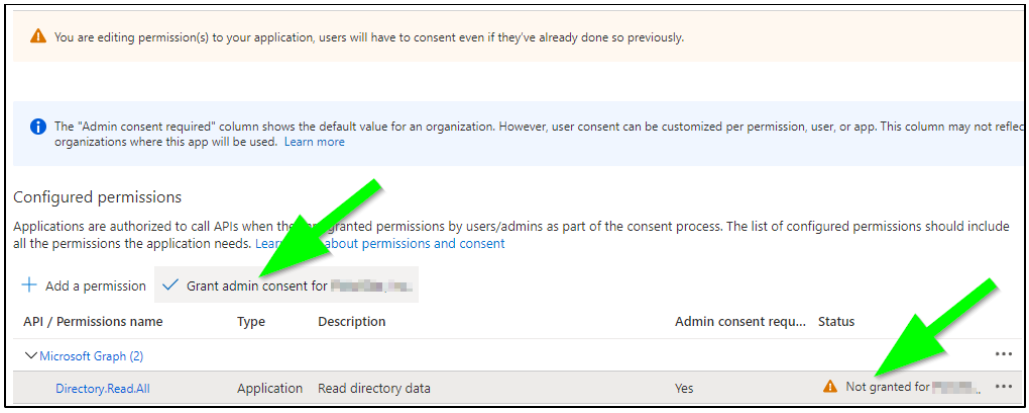


5) Besides the default ***User.Read*** permission, the ***Directory.Read.All*** permission under "Microsoft Graph" must also be *requested* for this application:

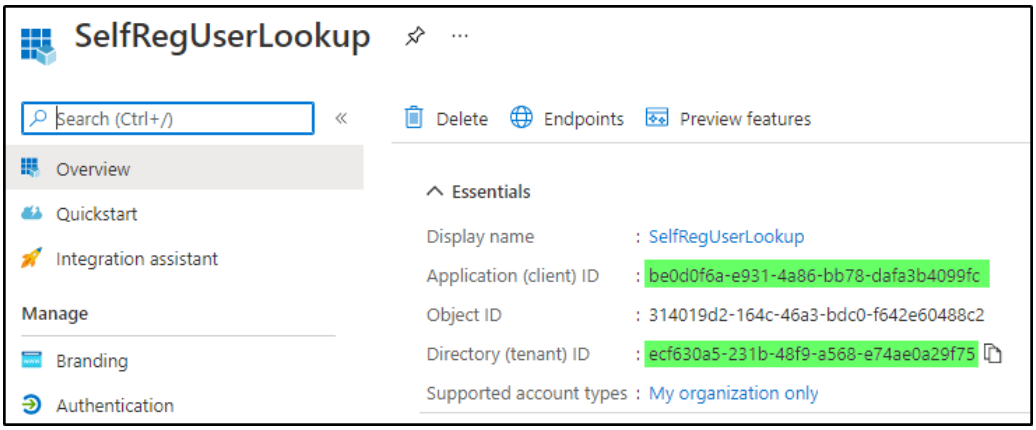


API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (2)				
Directory.Read.All	Delegated	Read directory data	Yes	Granted for PistolStar, Inc.
User.Read	Delegated	Sign in and read user profile	No	Granted for PistolStar, Inc.

6) The final step for the new app is to grant admin “consent” for the newly requested permission:



7) On the main page for the new app under the “Essentials” heading, copy the following values, they will need to be entered in the root web.config file later:
   a. Application (client) ID
   b. Directory (tenant) ID



8) The final step in Azure AD is to create an account that will actually authenticate and utilize this new app. This account will perform programmatic, background logins to Azure AD so it CANNOT require multi-factor authentication.

Configuration

The standard configuration for Self-Registration must be performed (e. g. the PG Service account must be delegated permissions to create Users in Active Directory). The **sysdir-snip.xml** in the deployment kit contains some default field mapping values that can be utilized.

Custom Groups

The **register.ashx.cs** file can be edited to programmatically create new values as a result of the user's submitted data. As an example, a dynamic group is currently written to a new "CampusGroup" field that is then used to assign this user to a specific group based on their chosen campus:

```
private int PreCreateUserAccount(ref HttpRequest req, ref HttpResponse resp) {
    // Ensure the selected username isn't in Azure AD!
    if (Utilities.existsInAzureAD(req.Form["Username"])) {
        errors.Add(new PGError(PGError.VLDERR.UNUSABLE_USERNAME_AAD, "Username"));
        //resp.Write(PortalGuard.Utilities.BuildErrorXML("Username exists in Azure AD"));
        resp.Write(BuildReturnXML());
        return 1;
    }

    // Map the "campus" to the proper AD group (must be the full DN of the target group!)
    if (0 == String.Compare(req.Form["campus"], "chicago", true)) {
        SetFormField(ref req, "CampusGroup", "CN=ChicagoCampus,OU=SelfReg,OU=Dev,DC=portalguard,DC=us");
    } else if (0 == String.Compare(req.Form["campus"], "new york", true)) {
        SetFormField(ref req, "CampusGroup", "CN=NewYorkCampus,OU=SelfReg,OU=Dev,DC=portalguard,DC=us");
    } else if (0 == String.Compare(req.Form["campus"], "san diego", true)) {
        SetFormField(ref req, "CampusGroup", "CN=SanDiegoCampus,OU=SelfReg,OU=Dev,DC=portalguard,DC=us");
    }
    return PGAPI_RC_NOERROR;
}
```

Web.Config – SQL Connection String

In the root web.config file is a **<connectionStrings>** element. The "connectionString" attribute of the <add> element it contains must have the following values set in order for the .NET code to connect to the SQL database containing the self-registration data:

- **Server** – The Microsoft SQL server and instance name, e. g. *SQLserv\SQLEXPRESS*

- **User ID** – The SQL server user account with read/write permissions to the "pstar" database
- **Password** – The password for the prior SQL account. This must **not** contain a semi-colon

Web.Config – SMTP Settings

In order for this self-registration code to send email notifications to users, the following values must be set in the **<system.net> <mailSettings> <smtp>** element:

- **from** – The "From" address used for the emails, e. g. *noreply@YOUR-DOMAIN.com*
- **host** – The IP or host name of your SMTP relay, e. g. *Your.SMTP.Relay*
- **port** – The numeric port on which the SMTP server listens. This is typical 25 or 587
- **username** – The username to use if your SMTP server requires authentication
- **password** – The password for the SMTP account

Web.Config – Email Subjects and Body Templates

To control the content in the emails sent during the full self-registration process, the following values must be set in the **<appSettings>** element:

- **Staging_EmailSubj** – The subject for the initial email notification
- **Staging_EmailBody_TemplateFile** – The name of the file in the "Policies" folder containing the initial email notification. This should always be set to *email-template-STAGED.html*
- **Completed_EmailSubj** – The subject line for the second email notification
- **Completed_EmailBody_TemplateFile** – The name of the file in the "Policies" folder containing the second email notification. This should always be set to *email-template-COMPLETED.html*

Web.Config – Azure AD Connectivity

This Tailored Authentication project queries Azure AD in real-time to ensure the user's choice of a username does NOT already exist there. The following settings must be set in the **<appSettings>** element:

- AADTenantName – The human-readable Azure AD tenant name, e. g. *tenant.onmicrosoft.com*
- AADTenantID – The GUID for "Directory (tenant) ID" from the registered Azure AD app
- AADClientID – The GUID for "Application (client) ID" from the registered Azure AD app
- AADUser – The full username for background authentication to Azure AD, e. g. *genericuser@yourtenant.onmicrosoft.com*
- AADPassword – The password for the account above

Web.Config – Application Authentication

POST requests to the new *Pacific-SelfReg-TA.ashx* endpoint are what start the self-registration flow with PortalGuard. These requests must be authenticated to ensure only authorized users are allowed to self-register through PortalGuard. The initial implementation of this authentication requires the web application performing the POST request to send a shared secret in the "Authorization: Basic" request header. This header value is protected during transit by TLS/SSL, but if requested, BIO-key can

implement a more secure method once Pacific College is more comfortable with the overall flow and has the process working end-to-end. The following setting must be set in the **<appSettings>** element:

- **SharedSecret** – The value that PG will check to authorize the initiating POST request.

Email Templates

Edit the two HTML files in the "Program Files\PistolStar\PortalGuard\Policies" folder as follows:

- **email-template-STAGED.html** - The initial email sent to new students informing them they can complete registration by clicking the link in the email which takes them to the page where they can choose their own username and set their initial password.
- **email-template-COMPLETED.html** – The second email sent to students who have successfully completed self-registration through PortalGuard. In this scenario, the user account will have been created in Active Directory and their Office 365 account should be created automatically by Azure AD Connect within minutes. This email can provide helpful links, e. g. the student portal.

Both templates can utilize {PLACEHOLDER} values where fields from the self-registration data will be automatically entered into the final email. Based on the initial scope of this project, the available placeholders are:

- {FIRSTNAME}
- {LASTNAME}
- {CAMPUS}
- {EMAIL}
- {PHONE}