# Deploying A Custom React Webpart to Microsoft online

Table of Contents

[Deploying A Custom React Webpart to Microsoft online 1](#_Toc68179006)

[1. Configure the Webpart: 1](#_Toc68179007)

[a. Update the webpart manifest (…/src/webparts/mywebpart/mywebpart.manifest.json) 1](#_Toc68179008)

[b. Add API permissions for this app the package-solution.json 2](#_Toc68179009)

[2. Build the app package 3](#_Toc68179010)

[3. Install the app package on the tenant site 3](#_Toc68179011)

[4. Approve Requested Permissions 4](#_Toc68179012)

[5. Create a SharePoint page for the app and publish the app 4](#_Toc68179013)

[Appendix 5](#_Toc68179014)

[Application Registration and Authentication 5](#_Toc68179015)

[2) Configure your app package-solution.json file with “Microsoft graph” resource and scope pairs 5](#_Toc68179016)

[How to deploy the App on a single site on a tenant using PowerShell 6](#_Toc68179017)

[Useful Links 6](#_Toc68179018)

[Jaxspdev tenant App catalog: 6](#_Toc68179019)

[Jax tenant – Financial Services App catalog 6](#_Toc68179020)

[Online Resources 6](#_Toc68179021)

## Configure the Webpart:

### Update the webpart manifest (…/src/webparts/mywebpart/mywebpart.manifest.json)

Web part can be configured to be exposed as an app page. This configuration is performed in the web part manifest file by adjusting the supportedHosts value. Web part will be exposed as an option in the upcoming app pages picker user interface, if the supportedHosts value contains "SharePointFullPage" value.

{

"$schema": "https://developer.microsoft.com/json-schemas/spfx/client-side-web-part-manifest.schema.json",

"id": "e82f1b2b-b1bb-4262-bf1f-43739b07ed89",

"alias": "RqcSiteWebPart",

"componentType": "WebPart",

"version": "\*",

"manifestVersion": 2,

"requiresCustomScript": false,

**"supportedHosts":** ["SharePointWebPart","SharePointFullPage"],

"preconfiguredEntries": [{

"groupId": "5c03119e-3074-46fd-976b-c60198311f70",

"group": { "default": "Other" },

"title": { "default": "RQC SITE" },

"description": { "default": "RQC Access Approval App" },

"officeFabricIconFontName": "Page",

"properties": {

"description": "RQC SITE",

"requestedby": {},

"user": {},

….

"acctunit": [],

"activity": [],

"comment": ""

}

}]

}

* 1. Add API permissions for this app the package-solution.json (…/config/package-solution.json)

Since this webpart uses graph api to read user profile and also to access data stored in Microsoft document libraries, we need to add the User.read, and Sites.read.all permissions to our solution.

{

"$schema": "https://developer.microsoft.com/json-schemas/spfx-build/package-solution.schema.json",

"solution": {

"name": "rqcsite-client-side-solution",

"id": "ba0e2514-b228-4fe2-8a4e-2443a8e77cb7",

"version": "1.0.0.0",

"includeClientSideAssets": true,

"skipFeatureDeployment": true,

"isDomainIsolated": false,

"webApiPermissionRequests": [

{

"resource": "Microsoft Graph",

"scope": "User.Read" //Read current signed in user profile

},

{

"resource": "Microsoft Graph",

"scope": "Sites.Read.All" //Read items in all site collections

}

]

},

"paths": {

"zippedPackage": "solution/rqcsite.sppkg"

}

}

## Build the app package

* 1. Cd to your webpart root directory
  2. Run: gulp build
  3. Run: gulp bundle –ship
  4. Run: gulp package-solution --ship

Fix any error detected when running step 2 or 3 before running step 4. After running step 4, check that the package (mywebpart.sppkg) was generated under sharepoint/solution/

## Install the app package on the tenant site

Log onto https://mytenant\_name. sharepoint.com/sites/apps/SitePages/Home.aspx

A screenshot of a cell phone

Description automatically generated

Since we will distribute this app to SharePoint pages, then:

1. click on “Distribute apps for SharePoint”
2. Click on “New” if new app or click on “upload” if an update and follow the instructions

A screenshot of a cell phone

Description automatically generated

1. Click OK and follow the instructions – for global app visibility, accept

## Approve Requested Permissions

Since this App will use Microsoft Graph to make API calls to the tenant’s documents Library,

You will see the following message in the last screen of the App install:

*“Components in this package require access to additional endpoints. Please contact your Tenant Administrator to ensure access to such end points is enabled. These are the permissions that need to be reviewed: Microsoft Graph, User.Read; Microsoft Graph, Sites.Read.All*

*”*

you need to logon to your tenant admin console tenantName-admin.sharepoint.com to approve these one time permissions :

* click on “Advanced” -> “API Access” -> “Pending Requests”
* Check the radio button near each request then Approve

## Create a SharePoint page for the app and publish the app

* Go to your tenant a create a team site
* On the new team site click on “Page” -> “New”

## Appendix

### Application Registration and Authentication

For your application to access Microsoft backend data (Office 365, SharePoint, …), you have two options:

1. Register and configure your application with Microsoft Azure AD using your tenant account (Authorization, API, SCOPE).
2. Configure your app package-solution.json file with “Microsoft graph” resource and scope pairs. Example: if your app will use Microsoft graph to make api calls to both the user profile and files stored in Microsoft document libraries of your tenant site,
   1. Cd ot Dev Code Base: go to your app code base -> config -> package-solution.json then add this code block under the “solution”:{… } construct and save:

"webApiPermissionRequests": [

{

"resource": "Microsoft Graph",

"scope": "User.Read"

},

{

"resource": "Microsoft Graph",

"scope": "Sites.Read.All"

}

]

* 1. Cd to Dev Code Base: Package the App
     1. Cd to your app code base then:
     2. Build solution : gulp build
     3. Create a bundle: gulp bundle –-ship
     4. Create a package: gulp package-solution –-ship
  2. Tenant App Catalog: Upload the newly created package to you tenant App catalog : <https://jaxspdev.sharepoint.com/sites/apps/SitePages/Home.aspx>
     1. App catalog -> distribute apps for [sharepoint | office] -> upload
     2. Follow the wizard
  3. Tenant Admin: Grant API Access permissions
     1. <https://jaxspdev-admin.sharepoint.com/_layouts/15/online/AdminHome.aspx#/home>
     2. Login to your tenant admin console
     3. Click on “Advanced” -> “API access” -> “Pending Requests” and approve pending requests as needed

How to deploy the App on a single site on a tenant using PowerShell:

* Connect to your SharePoint online tenant admin console with your admin credentials using PowerShell:
  + Cmd: Connect-SPOService -Url https://myTenant-admin.sharepoint.com
* Next, create an app catalog on the target site permissions
  + *Cmd: Add-SPOSiteCollectionAppCatalog -Site https://myTenant.sharepoint.com/sites/myStite*
* Once the app catalog created on a site collection, people with permissions to this app catalog library they can directly deploy the app and start using on this site. Make sure you uncheck the “global visibility” check box

A screenshot of a cell phone

Description automatically generated

## Useful Links

### Jaxspdev tenant App catalog:

* <https://jaxspdev.sharepoint.com/sites/apps/AppCatalog/Forms/AllItems.aspx>
* <https://jaxspdev.sharepoint.com/_layouts/15/addanapp.aspx>
* <https://jaxspdev.sharepoint.com/sites/apps/SitePages/Home.aspx>

### Jax tenant – Financial Services App catalog

* <https://jacksonlaboratory.sharepoint.com/sites/FinancialServices/AppCatalog/Forms/AllItems.aspx>

### Online Resources

* How to deploy : <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/single-part-app-pages>