SharePoint Framework (SPFx) App Testing & Approval



Policies & Procedures

Prepared for

CUSTOMER NAME 2/24/2022

Version 0.4 Draft

Prepared by

**Steve Hose, Gunjan Datta, Dade Register**

Senior Consultant

Contributors

**Steve Nihan, Ami Schreiber**

Revision and Signoff Sheet

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change Reference |
| 1/28/22 | Steve Hose | .1 | Initial draft for review/discussion |
| 2/4/22 | Gunjan Datta | .2 | Review and finalized for draft submission |
| 2/18/22 | Gunjan Datta | .3 | Final review of the draft document |
| 2/24/22 | Dade Register | .4 | Generalized as a template for multiple customers |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version Approved | Position | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Table of Figures 5](#_Toc96070656)

[1 Introduction 6](#_Toc96070657)

[2 App Submission/Approval Summary 7](#_Toc96070658)

[3 App Testing Policies 8](#_Toc96070659)

[3.1 Required Checks 8](#_Toc96070660)

[3.1.1 Test application in the required browsers 8](#_Toc96070661)

[3.1.2 Web part should work in all hosts in which it is exposed 8](#_Toc96070662)

[3.1.3 All properties exposed in the property pane should be functional 8](#_Toc96070663)

[3.1.4 All URLs displayed in the web part must be working 8](#_Toc96070664)

[3.1.5 Application doesn't stop responding 9](#_Toc96070665)

[3.1.6 Is the solution complete 9](#_Toc96070666)

[3.1.7 Is the application's UI complete 9](#_Toc96070667)

[3.1.8 Verify that browsers don't block your pop-ups 9](#_Toc96070668)

[3.1.9 Correct first-launch experience 9](#_Toc96070669)

[3.1.10 Test syncing to Teams 9](#_Toc96070670)

[3.1.11 Acquire account or license 10](#_Toc96070671)

[3.1.12 No demo content if free trial is included in the offering 10](#_Toc96070672)

[3.2 Recommended checks 10](#_Toc96070673)

[3.2.1 Web part icon should be 64x64px with transparent background 10](#_Toc96070674)

[3.2.2 Microsoft Teams logos will need to be using correct image sizes 10](#_Toc96070675)

[3.2.3 Full page image must be sized properly 10](#_Toc96070676)

[3.2.4 Web part should have a title that's configurable in the property pane 10](#_Toc96070677)

[3.2.5 Properly handle different themes 11](#_Toc96070678)

[3.3 Test Cases Checklist 11](#_Toc96070679)

[4 SPFx App Technical Review 13](#_Toc96070680)

[4.1 Technical Review 13](#_Toc96070681)

[4.1.1 Out of the Box 13](#_Toc96070682)

[4.1.2 Review Permission Requests 13](#_Toc96070683)

[4.1.3 Review all instances of AadHttpClient 16](#_Toc96070684)

[4.1.4 Review all “externals” entries in config.json 17](#_Toc96070685)

[4.1.5 Review all Import Statements 18](#_Toc96070686)

[4.1.6 Review all httpClient references 18](#_Toc96070687)

[4.2 Tools for Reviewing Packages 18](#_Toc96070688)

[5 App Test Procedures 20](#_Toc96070689)

[6 App Deployment Request Approval Procedures 21](#_Toc96070690)

[Appendix – Additional Resources 23](#_Toc96070691)

## Table of Figures

[Figure 1 - App Submission/Approval Summary 7](#_Toc96070692)

[Figure 2 – App Permission Requests 13](#_Toc96070693)

[Figure 3 - Graph API Permission Requests 14](#_Toc96070694)

[Figure 4 - Web API Permission Requests 15](#_Toc96070695)

[Figure 5 - API Permission Requests 16](#_Toc96070696)

[Figure 6 - External Library References 17](#_Toc96070697)

[Figure 7 - Imported Libraries 18](#_Toc96070698)

[Figure 8 - App Test Process 20](#_Toc96070699)

[Figure 9 - App Approval Process 21](#_Toc96070700)

1. Introduction

Before a SharePoint Framework (SPFx) solution is published in the site collection or tenant application catalog, it must go through submission, approval, and deployment process. These processes are highly recommended and are meant to ensure that your app functions properly and that its page clearly describes its purpose and features. This document will go over the approval and testing processes for a SPFx solution.

1. App Submission/Approval Summary



Figure 1 - App Submission/Approval Summary

The app submission process will have a test site created for the developers to complete testing in the client’s environment. As part of the approval process, a test site will be used to complete the test procedures to validate the app for deployment to the app catalog.

1. SPFx App Testing Policies

This section will go over the test criteria the app must/should pass for a successful deployment to the application catalog.

* 1. Required Checks

These policies will be required for app deployment.  
Test application in both root and non-root sites

When testing your application, verify that it's working as intended when installed both in the root site collection of a tenant as well as non-root sites. This will help you verify that it doesn't have any dependencies on specific site URLs.

* + 1. Test application in the required Web browsers

When testing your application against checks described in the following section, verify that it works as expected on browsers available to the client’s GFE. Additional testing in Teams will be required if the app is deployable to Teams.

* + 1. Web part should work in all hosts in which it is exposed

When building web parts using SharePoint Framework, you can choose [where they should be exposed](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/integrate-with-teams-introduction). Before submitting your application for review, ensure that your web part works as intended in all these hosts. The solution must work in both classic and modern pages.

* + 1. All properties exposed in the property pane should be functional

Web Part properties exposed in the property pane of your web part should be meaningful. Changing their settings should clearly change the behavior of your web part.

* + 1. All URLs displayed in the web part must be working

If your web part shows URLs, they must be pointing to existing resources and must not lead to non-existent or error pages.

* + 1. Application doesn't stop responding

When using your application, it should fully engage in responding to users. If it needs to perform an operation that could take a moment to complete, it communicates this to the user.

* + 1. Is the solution completed?

The solution should have all functionality as advertised in its description and screenshots. This process is not meant to be used for developer testing of an unfinished product.

* + 1. Is the application's UI complete?

The first impression users will have of your application will be by looking at its UI. The following items could give the impression that your application isn't completed:

* It doesn't look polished, or some elements are misaligned
* It uses many different fonts, or some placeholder text has not been replaced by its final value

**Tip:**

To ensure seamless integration with SharePoint and Teams, we recommend that you use Office UI Fabric which is the design language for Microsoft 365. For more information, see the [**design**](https://docs.microsoft.com/en-us/sharepoint/dev/design/design-guidance-overview) section of this documentation.

* + 1. Verify that browsers don't block your pop-ups

If your application uses pop-ups, verify that browsers with default settings don't block them interrupting the user flow.

* + 1. Correct first-launch experience

Your application should work when launched for the first time. If it needs to be pre-configured, clearly communicate the configuration settings and steps to the user. If the configuration is complex or requires additional information, provide clear instructions to help users set up your application.

* + 1. Test syncing to Teams

Deploying a solution to Teams will require the SPFx solution to be deployed to the tenant app catalog and deployed to all sites. Due to this requirement, testing in Teams should be completed after the app is approved for deployment. The SPFx webpart can still be tested in modern and classic pages.

Once the SPFx webpart(s) are approved and deployed, verify that they can be deployed to teams using the **Sync to Teams** option in the App Catalog without any errors. After syncing the configuration to Teams, verify that all web parts are exposed in the Teams' App Catalog as intended, can be installed, and work as expected.

* + 1. Acquire account or license

Solutions requiring additional licensing will not be allowed to be deployed.

* + 1. No demo content if free trial is included in the offering

Demo solutions and free trial apps will not be allowed to be deployed.

* 1. Recommended checks

The following checks are recommended to pass. While a failed check in this category won't prevent you from publishing your application in the Marketplace, it might have a negative impact on end-user experience.

* + 1. Web part icon should be 64x64px with transparent background

We recommend that you use a 64x64px image with transparent background for your web part's icon. This will allow it to be correctly displayed in the different UI modes in SharePoint.

* + 1. Microsoft Teams logos will need to be using correct image sizes

If your solution contains components targeted for Microsoft Teams, they are expected to use the official image sizes based on Microsoft Teams guidance. The size of outline image has to be at 32x32px and color image at 192x192px.

* + 1. Full page image must be sized properly

If solution contains components that are targeted to be exposed as single part app pages, each of these components must have fullPageAppIconImageUrl attribute with specific image designed to be used when new page is created in SharePoint. This image must be sized as 193x158px.

* + 1. Web part should have a title that's configurable in the property pane

We recommend that all web parts have a title that clearly describes the information the web part presents. Page authors should be able to adjust this title through property pane, so that if the web part is configurable, the title can correctly describe its contents.

**Tip**

You can easily add a configurable title to your web part by using the **[WebPartTitle](https://pnp.github.io/sp-dev-fx-controls-react/controls/WebPartTitle/)** PnP Reusable React control.

* + 1. Properly handle different themes

Your application should properly handle light- and dark-color themes. Switching between the different themes shouldn't have a negative effect on the usability of your application. While not supporting themes will not prevent you from publishing your app in the Marketplace, it's recommended that you support them to offer optimal user experience.

**Tip**

For more information about using themes in your application, see the [**Use theme colors in your SharePoint Framework customizations**](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/use-theme-colors-in-your-customizations) article.

* 1. Test Cases Checklist

The table below will be completed by the submitter of the application.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Test Type** | **Test Info** | **Pass/Fail** |
| Tested in root site | Required | This must be done outside of the client’s tenant. |  |
| Tested in non-root site | Required | This must be done in the test site provided by the client. |  |
| Tested in Chrome, Edge (Classic), Edge (Chromium), IE, Firefox, and Opera | Required | The app must be tested and working in the client’s browsers. |  |
| Tested in a modern page | Required | The app must work in a modern page. |  |
| Tested in a classic page | Required | The app must work in a classic page. |  |
| Tested in Teams | Required | The app must work in teams, if able to deploy to it. |  |
| Valid webpart properties | Required | Validate the webpart properties are meaningful. |  |
| Validate URLs | Required | If URLs are present, validate that they work. |  |
| Validate response time | Required | The application should load within ten seconds. A wait screen is acceptable, but the page should load within ten seconds. |  |
| Validate app description | Required | Validate the description of the application matches its description. |  |
| Validate user interface (UI) | Required | Ensure that the user interface is complete, and elements are not misaligned. |  |
| Validate pop-ups | Required | Validate that the application does not block any pop-ups created by the application. |  |
| Validate installation | Required | Validate that the installation of the application. |  |
| Validate licensing | Required | Validate that no additional licenses are required by the application. |  |
| Validate application icon | Optional | Validate that the icon is 64x64 pixels. |  |
| Validate the Teams logo | Optional | Ensure that the Teams application logo is of the correct size. |  |
| Validate sizing | Optional | Validate that all images and the application resize correctly after the window is resized. |  |
| Validate themes | Optional | Validate the application renders correctly when the theme changes. |  |

1. SPFx App Technical Review

This section will give an overview of reviewing a SPFx application.

* 1. Technical Review

A technical review of the SPFx application source code will be required before deploying the solution.

* + 1. Out of the Box

To prevent reinventing the wheel scenarios, the application should be analyzed and rejected when if the application features can be completed using out of the box methods.

* + 1. Review Permission Requests

The permission requests can be found in the SPFx source code, under the “config/package-solution.json” file.

* + - 1. App Requests (Add-In)

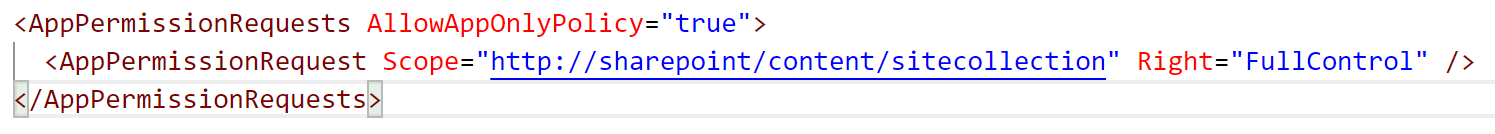


Figure – App Permission Requests

The app permission requests are used in SharePoint Add-Ins, also supported by the application catalog. This method of authentication can be used to elevate permissions and run under the context of the app. If the application is utilizing app permission requests, then the code should be investigated to ensure the client secret value isn’t within the code. It is recommended to use the Graph API or Azure to execute logic with an elevated permission outside the context of the user.

* + - 1. Graph API Requests

Graphical user interface, text, application, email

Description automatically generated

Figure - Graph API Permission Requests

The Graph API will require the permissions to be specified. This will ensure that the application can only do what the application is requesting permissions for. Review the Graph API permission requests, as shown in the figure below.

Graphical user interface, text, application, email

Description automatically generated

Figure - Web API Permission Requests

Requests for read all, as shown above, shouldn’t be allowed. This will allow the app to access all data in the tenant, and not based on the permissions of the user.

* + 1. Review all instances of AadHttpClient

Graphical user interface, text, application, email

Description automatically generated

Figure 5 - API Permission Requests

The web api permission requests can be pointed to a custom API. Azure functions would be an example of a custom API which can be utilized to elevate permissions and execute the request under a different context. Review the justification and logic of the API to ensure they match.

* + 1. Review all “externals” entries in config.json

Text

Description automatically generated

Figure - External Library References

Review the external libraries consumed by this library. Refer to section 2.3 of the App Submission guide for additional information on external libraries.

* + 1. Review all Import Statements

Text

Description automatically generated

Figure - Imported Libraries

Import statements are a way to statically or dynamically load external references and libraries into an app.

See <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/dynamic-loading>

Review the external libraries consumed by this library. Refer to section 2.3 of the App Submission guide for additional information on external libraries.

* + 1. Review all httpClient references

The source code will need to be reviewed for HTTP requests that are made to external addresses for validity.

See <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/connect-to-anonymous-apis>

* 1. Tools for Reviewing Packages

Tools are available to help automate the scanning of code for vulnerabilities.

<https://www.cvedetails.com/>

<https://snyk.io/>

<https://security.snyk.io/> Open Source Vulnerability Database

<https://rencore.com/products/code/> SPCAF

1. App Test Procedures



Figure 8 - App Test Process

The application test procedures are shown in the figure above. The submitter of the application must complete the test case checklist as defined in section 3.3. Once the tests are completed and all required tests have passed, the submitter should submit a request for deployment. The checklist will be required as part of the submission process.

1. App Deployment Request Approval Procedures



Figure 9 - App Approval Process

The application approval process will include an evaluation of the tests completed and a technical review, as stated in sections 3 and 4. The test case checklist, defined in section 3.3, will be reviewed for completion. All tests must be completed, including the optional tests even though they are not required for approval. The technical review will then be scheduled to take place. Once this process is completed, the application should be deployed to the application catalog and the request approved.

# Appendix – Additional Resources

* [**Publish SharePoint Framework applications to the marketplace**](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/publish-to-marketplace-overview) Considerations for publishing SharePoint Framework applications to the marketplace
* **[Tenant Wide Deployment of SharePoint Framework Extensions](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/extensions/basics/tenant-wide-deployment-extensions)** Activating SharePoint Framework extensions across tenant from centralized location.
* **[Modifying Microsoft Search experience using SharePoint Framework extensions](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/building-search-extensions)** You can modify the Microsoft Search experiences using SharePoint Framework extensions
* **[Host extension from Microsoft 365 CDN (Hello World part 4)](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/extensions/get-started/hosting-extension-from-office365-cdn)** Deploy your SharePoint Framework Application Customizer to be hosted from a Microsoft 365 CDN and deploy that to SharePoint for end users.
* **[SharePoint Framework guidance on dynamic loading of packages](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/dynamic-loading)** Techniques to optimize the run-time size and execution speed of your SharePoint Framework component.
* **[Provision SharePoint assets with your solution package](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/toolchain/provision-sharepoint-assets)** SharePoint Framework toolchain allows you to package and deploy SharePoint items with your client-side solution package. These items are then provisioned when the client-side solution is installed on a site.
* **[SharePoint solution packaging](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/basics/notes-on-solution-packaging)** The package-solution gulp task looks at /config/package-solution.json for various configuration details in SharePoint Framework, including ISolution and IFeature definitions.
* **[Validate web part property values](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/guidance/validate-web-part-property-values)** Validate property values in SharePoint Framework client-side web parts by validating the value directly inside a web part's code, or by calling an external API.