SharePoint Framework (SPFx) App Submission



Policies & Procedures

Prepared for

CUSTOMER NAME 2/24/2022

Version 0.5 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 |
| 2/10/22 | Gunjan Datta | .3 | Incorporated client feedback |
| 2/18/22 | Gunjan Datta | .4 | Final review of the draft document |
| 2/24/22 | Dade Register | .5 | Generalized as a template for multiple customers |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Reviewers

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

Table of Contents

[Table of Figures 5](#_Toc96070743)

[1 Introduction 6](#_Toc96070744)

[2 SPFx App Submission Policies 7](#_Toc96070745)

[2.1 App Package 7](#_Toc96070746)

[2.1.1 Use SharePoint Framework v1.11 or newer 7](#_Toc96070747)

[2.1.1 Solution package must contain valid developer metadata 7](#_Toc96070748)

[2.1.2 Web part metadata description text must briefly describe its functionality and value proposition 8](#_Toc96070749)

[2.1.3 Solution's description must clearly describe the different components included in the package 8](#_Toc96070750)

[2.1.4 Microsoft Teams targeted solutions will need to be tenant scoped 8](#_Toc96070751)

[2.2 App Naming 9](#_Toc96070752)

[2.2.1 Don't use names of features in Teams and SharePoint in your solution's name 9](#_Toc96070753)

[2.2.2 Don't use names of other Microsoft's products in your solution's name 9](#_Toc96070754)

[2.2.3 Don't use keywords that are related to Microsoft's products 9](#_Toc96070755)

[2.2.4 Naming your solution 9](#_Toc96070756)

[2.3 App Libraries and References 9](#_Toc96070757)

[2.3.1 Provide justification for any API requests 9](#_Toc96070758)

[2.3.2 Don't include vulnerable resources 10](#_Toc96070759)

[2.3.3 Use Approved Resources from Approved Locations 10](#_Toc96070760)

[2.4 App Security 10](#_Toc96070761)

[2.4.1 Supporting end users to inject scripts is not allowed 10](#_Toc96070762)

[2.4.2 Application shouldn't prompt users without explicit interaction 10](#_Toc96070763)

[2.4.3 Don't include malicious code 10](#_Toc96070764)

[2.4.4 App permissions shouldn’t access all data in tenant 10](#_Toc96070765)

[2.5 App Functionality 11](#_Toc96070766)

[2.5.1 Ensure SharePoint Framework Contract Conformance 11](#_Toc96070767)

[2.5.2 Don't include additional advertising in your app's description and screenshots 11](#_Toc96070768)

[2.5.3 The package file can be uploaded to App Catalog without errors 11](#_Toc96070769)

[2.5.4 The app can be deployed without errors 11](#_Toc96070770)

[2.5.5 Links with additional information are correct 11](#_Toc96070771)

[2.5.6 Ensure that your application doesn't use resources excessively 11](#_Toc96070772)

[3 App Submission Procedures 12](#_Toc96070773)

[3.1 App Information 12](#_Toc96070774)

[3.2 App Assets 13](#_Toc96070775)

[4 App Approval Procedures 14](#_Toc96070776)

[4.1 App Approval Requirements 14](#_Toc96070777)

[4.2 Review App Information 15](#_Toc96070778)

[4.2.1 App Justification 15](#_Toc96070779)

[4.3 Review App Security Justification 15](#_Toc96070780)

[4.4 Review App Functionality 15](#_Toc96070781)

[4.4.1 Deploy to Test Site 15](#_Toc96070782)

## Table of Figures

[Figure 1 - App Submission Procedure 12](#_Toc96070783)

[Figure 2 - App Approval Procedure 14](#_Toc96070784)

1. Introduction

Before a SharePoint Framework (SPFx) solution is published in the site collection or tenant application catalog, it must go through a 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 submission process of a SPFx solution.

1. SPFx App Submission Policies

Submission of a SPFx application will require activities from both the developer(s) submitting the application and the approval group. This section will go over the information a developer must know before submitting their app through this process.

* 1. App Package

This section will review the requirements for the SPFx application package. All SPFx packages must follow the policies listed in this section.

* + 1. Use SharePoint Framework v1.11 or newer

To build solutions that you want to publish to the site collection or tenant application catalog, you need to use SharePoint Framework v1.11 or higher. Refer to the [SPFx roadmap](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/roadmap) for details of the latest version. Starting from version 1.11, SharePoint Framework solution manifest has additional information about your organization and solution, which is required for publishing your application in the application catalog.

**Tip**

If your application is built using an older version of the SharePoint Framework, you can easily upgrade it to SharePoint Framework v1.11 or newer using CLI for Microsoft 365. For more information, visit [**https://pnp.github.io/cli-microsoft365/cmd/spfx/project/project-upgrade/**](https://pnp.github.io/cli-microsoft365/cmd/spfx/project/project-upgrade/).

* + 1. Solution package must contain valid developer metadata

Solution's manifest (managed in the package-solution.json file) must have a new section named developer with the following properties:

|  |  |  |
| --- | --- | --- |
| SOLUTION PACKAGE MUST CONTAIN VALID DEVELOPER METADATA | | |
| Attribute | **Description** | **Mandatory** |
| name | Name of the organization that built the application | Yes |
| description | Description of the application being submitted | Yes |
| websiteUrl | URL of a website with additional information about the application | Yes |
| privacyUrl | Privacy statement URL | Yes |
| termOfUseUrl | Terms of use URL | Yes |

Including this additional information allows you to clearly communicate information about your application towards your users and in the future should allow you to access additional usage data.

The developer information is displayed when installing web parts in Microsoft Teams. Information about the application and the developer that is displayed in the application catalog is managed separately in the site.

* + 1. Web part metadata description text must briefly describe its functionality and value proposition

Ensure that your web part's description in the metadata is meaningful so that users understand what your web part’s purpose is and what they can expect of it.

* + 1. Solution's description must clearly describe the different components included in the package

When submitting your package, be sure to clearly describe what different components (web parts, application customizers, etc.) are included in the package to manage user expectations and help them understand the impact of using your application.

* + 1. Microsoft Teams targeted solutions will need to be tenant scoped

SharePoint Framework solutions targeted to be also used in Microsoft Teams will need to be [tenant-scoped targeted](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/tenant-scoped-deployment). This setting is controlled with the skipFeatureDeployment attribute in the package-solution.json configuration file and setting should be true for any solution, which has components targeted also to Microsoft Teams.

* 1. App Naming

This section will go over the policies related to naming the application.

* + 1. Don't use names of features in Teams and SharePoint in your solution's name

In your solution name, don't use words matching features in Microsoft Teams or SharePoint, such as **Chat**, **Contacts**, **Calendar**, **Calls**, **Files**, **Meeting**, **Activity**, **Teams**, **Apps**, **Help**, **SharePoint**, **List**, **Page**, etc. as these names could be confused with the standard functionality in Teams and SharePoint.

* + 1. Don't use names of other Microsoft's products in your solution's name

Your solution will be rejected if its name contains a name of another Microsoft product, such as **Outlook**, **Word**, **Calendar**, **Tasks**, **Teams**, **Excel**, **PowerPoint**, **Visio**, etc.

* + 1. Don't use keywords that are related to Microsoft's products

Your solution must not use Microsoft Teams or SharePoint feature-specific keywords like - **Start meeting**, **Make call**, **Upload file**, **Start chat**, etc. These keywords are used in Microsoft's products and using them in your application could confuse users. If your application offers similar functionality, include the name of your name in the action, for example: **Start a Zoom Meeting**, **Upload file to Dropbox**, **Start Freshdesk chat**, etc.

* + 1. Naming your solution

If your solution is named after a common word, such as **Orders**, you will need to include your organization/unit’s name as well to clearly differentiate it from other solutions in the Marketplace, for example, **Contoso Orders**. Where *Contoso* refers to the name of the organization/unit submitting the solution.

* 1. App Libraries and References
     1. Provide justification for any API requests

If your application requires additional API access (for example to communicate with Microsoft Graph), it should request them in the solution manifest. A justification for requested external API requests will be required as part of the review/approval app.

* + 1. Don't include vulnerable resources

Verify, that your code doesn't contain any vulnerabilities. If you're using third-party dependencies, verify the vulnerability in the [NIST database](https://nvd.nist.gov/vuln/search).

* + 1. Use Approved Resources from Approved Locations

The Navy has established approved resources and approved locations from which app packages can reference. [NPM JS](https://www.npmjs.com/) should be used to incorporate 3rd party libraries to ensure the library is coming from an authorized source.

* 1. App Security
     1. Supporting end users to inject scripts is not allowed

The solution cannot enable end users to embed arbitrary scripts to SharePoint pages. This means that solutions which enable classic script editor or content editor scenarios in SharePoint are not allowed to be published to application catalog. This model is not allowed as it opens security concerns by normal end users embedding scripts on the portal.

* + 1. Application shouldn't prompt users without explicit interaction

The solution shouldn't prompt users without explicit interaction on their part.

* + 1. Don't include malicious code

Solutions that contain malicious code, link to malicious websites or otherwise illegal resources will be rejected from publishing in the application catalog.

* + 1. App permissions shouldn’t access all data in tenant

The solution shouldn’t require full or read rights to the entire tenant. This can cause a security issue if the app id/secret is released. An exception can be made for this but will require additional approval outside of this process.

* 1. App Functionality
     1. Ensure SharePoint Framework Contract Conformance

Your web part or customizer must only manipulate the DOM element provided through the domElement property. This element and the provided context object are the only approved ways to add functionality into sites through your app. Directly manipulating the page DOM will result in rejection of your solution.

* + 1. Don't include additional advertising in your app's description and screenshots

Make sure that your app's description, screenshots, text, and promotional images, describe only the app and don't contain any additional advertising.

* + 1. The package file can be uploaded to App Catalog without errors

Verify that the solution package can be uploaded to the App Catalog in your tenant without any errors.

* + 1. The app can be deployed without errors

Verify that your application can be installed in a specific site but also that it can be deployed globally to all sites without any errors.

* + 1. Links with additional information are correct

Verify, that links such as **Terms of Service**, **Privacy Policy**, **About**, etc. are valid and point to correct pages.

* + 1. Ensure that your application doesn't use resources excessively

When testing your application, verify that it doesn't lead to noticeable or sustained increase in CPU usage, that it doesn't consume a lot of memory and that it doesn't lead to the browser becoming unresponsive. In general, your application should run well on an average desktop computer with internet access. Also, it shouldn't take longer than 10 seconds for your application to load.

1. App Submission Procedures



Figure 1 - App Submission Procedure

This section will give an overview of the app submission procedures required by the developer.

* 1. App Information

The following information will be required on submission of an application.

|  |  |  |
| --- | --- | --- |
| **Property** | **Description** | **Required?** |
| Name | The name of the application. | Yes |
| Description | A clear description of the application. | Yes |
| Justification | The reason for this application to be deployed. | Yes |
| Permissions | The app or Graph API permissions being requested by the application. | No |
| Permissions Justification | The justification for any external api will be required for approval. | Yes if “Permissions” exist |
| Unit/Organization | The unit/organization responsible for the application. | Yes |

* 1. App Assets

The submission process will require the following assets:

* SPFx package file
* SPFx manifest file
* Source code of the solution (compressed)

Failure to include the above assets will reject the app submission request.

1. App Approval Procedures



Figure 2 - App Approval Procedure

This section will give an overview of the app submission procedures required by the developer. Members of the approvers group will be able to approve or reject the application submitted for deployment.

* 1. App Approval Requirements

The app approval process will require a site collection to be created with the site collection app catalog enabled on it. This site will be used to complete testing of the application after the initial approval of the app submission.

* 1. Review App Information

The app information, defined in section 3.1, will need to be reviewed and validated against the policies listed in sections 2.1 and 2.2.

* + 1. App Justification

The app justification will need to be reviewed to ensure it will provide value to the organization. Understanding the app and what is does will also ensure duplicate apps will not exist in the application catalog.

* 1. Review App Security Justification

The SPFx app will run under the context of the user by default. Any apps requesting additional permissions to execute requests against the Graph API or custom API, will need further review. The more permissions being requested by an app, the more it should be scrutinized. The app should only have access to the site(s)/list(s) required by it. The justification for the permissions will need to be reviewed and approved.

* 1. Review App Functionality

The app will need to be validated against the policies defined in section 2.5. The app information and deployment to the test site should be straightforward and error free. The information displayed in the app catalog should be complete.

* + 1. Deploy to Test Site

A site collection will need to be created and used as a dedicated area for testing applications. The site collection app catalog will need to be enabled at this level. As part of the functionality testing, the approver will need to upload the package file and deploy the solution to a sub-site. This will be required as part of the policy validation defined in section 2.5.