Setting Up A SharePoint Framework Dev Environment with a SSIS backend support

Tables to request access:

* \*LAW.RQLOC
* \*LAW. REQUESTER
* \*LAW. ACACTIVITY
* \*LAW.RQACCTU

1. Setting Up A SharePoint Framework Dev Environment
   1. Setup an online test server
      1. Setup Office 365 Tenant
      2. Create app catalog site
      3. Create a new site collection
   2. Setup a local dev server – install developer tools
      1. Install Visual Studio Code
      2. Install JDK
      3. Install NodeJs
      4. Install Yeoman and gulp
      5. Install Yeoman Sharepoint generator
      6. Install SQL sever

* See- [Set up development environment](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/set-up-your-development-environment)

1. Setting a SSIS backend support
   1. Setup a project
   2. Create a solution
   3. Configure solution.param
   4. Configure staging.connmgr
   5. Configure lawson.connmgr
   6. Create SSIS package flow for each dataset
      1. Create and test SQL query
      2. Create the extract and staging step
      3. Create the data transform and transfer to sharepoint server

# Setting up Set up your Office 365 tenant

1. Administrator :
   1. Username: hutchl
   2. Pass: umaine pass – Mballa&2020
   3. Email: [hutchl@jaxspdev.onmicrosoft.com](mailto:hutchl@jaxspdev.onmicrosoft.com)
2. User:
   1. Lucie Hutchins
   2. User name: LucieHut
   3. Email: [LucieHut@jaxspdev.onmicrosoft.com](mailto:LucieHut@jaxspdev.onmicrosoft.com)
3. **Admin User details**
   1. Display name: LucieHut
   2. Username: LucieHut@jaxspdev.onmicrosoft.com
   3. Password: Mballa&2020
4. Domain Prefix: jaxspdev
5. Domain: jaxspdev.onmicrosoft.com
6. Create Authentication for your App – after you setup your solution dev env – see create webpart project

Useful Links:

* Admin Center: <https://admin.microsoft.com/Adminportal/Home?source=applauncher#/homepage>
* Developer Profile - Note: User Jax login to view the browse the developer profile

1. <https://developer.microsoft.com/en-us/office/profile/>

## Create app catalog site

* <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/set-up-your-developer-tenant#create-app-catalog-site>

Steps:

* Go to https:// jaxspdev-admin.sharepoint.com
* Then follow the steps:

1. **Create a new site collection**

* [**https://docs.microsoft.com/en-us/sharepoint/dev/spfx/set-up-your-developer-tenant#create-a-new-site-collection**](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/set-up-your-developer-tenant#create-a-new-site-collection)

Steps:

* Go to https://jaxspdev-admin.sharepoint.com
* Then follow the steps:

1. SetUp SharePoint Framework Development

* Install Node – version 10.19.0
  1. Use package installed
* Install Gulp
  1. Sudo npm install -g yo gulp
* Install Yeoman -
  1. npm install -g @microsoft/generator-sharepoint
* Install SharePoint REST API
  1. *npm install @pnp/logging @pnp/common @pnp/odata @pnp/sp –save*
  2. *See: https://www.c-sharpcorner.com/article/single-page-app-in-sharepoint/*
* Install Visual Code – then Install code in your PATH:
  1. open Visual studio Code editor
  2. Press “shift” + “command” + P
  3. On the shell prompt type “Install code in PATH” and hit enter
* Create a sharePoint project –
  1. Create a directory for your web-part project
     1. Mkdir mywp
     2. Cd to mywp and run command: yo @microsoft/sharepoint
     3. When prompted:
  + Accept the default **mywp** as your solution name, and then select Enter.
  + Select **SharePoint Online only (latest)**, and select Enter.
  + Select **Use the current folder** for where to place the files.
  + Select **N** to allow the solution to be deployed to all sites immediately.
  + Select **N** on the question if solution contains unique permissions.
  + Select **WebPart** as the client-side component type to be created.
    1. The next set of prompts ask for specific information about your web part:
  + Accept the default **HelloWorld** as your web part name, and then select Enter.
  + Accept the default **HelloWorld description** as your web part description, and then select Enter.
  + Select React **framework** as the framework you would like to use, and then select Enter.
* Once a project has been created with the Yeoman generator for the SharePoint Framework, execute the following command from within the root folder of the project. -- the self-signed developer certificate
  1. gulp trust-dev-cert
* To launch the new site:
  1. Cd to myproject
  2. Run: gulp serve
* Register and configure your application with Microsoft Azure AD using your tenant (Authorization, API, SCOPE) -
  1. Log onto <https://portal.azure.com/#home>
  2. Click on “View” under the **Manage Azure Active Directory section**
  3. On the left menu, click on “App registration”
  4. On the App registration page, click “New registration”
     1. Enter the name of your app “RQC Access Request App”
     2. Select “Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)”
     3. Set the Web field to <https://localhost:4321> – under the Redirect URI section
     4. Click Register
     5. In this case we got
        1. Client id: 21954173-f0ec-4d81-943a-f9d91d322713
        2. Tenant id: b56bc27e-6da9-4384-9cad-73117b11b0b3
     6. Go to App registrations then click on your new app
        1. Click on “Authentication” on the left menu
           1. On the “Authentication” page, under the “Implicit grant”, select the two check boxes (access tokens, and ID tokens)
        2. Click on “Certificates & secrets”
           1. Click “New client secret” then enter a name for the secret key
           2. Click “Add” to generate the ticket
           3. Copy and save the generated key for later use
        3. Set API permissions to application level – this requires admin consent
           1. Microsoft graph - <https://docs.microsoft.com/en-us/graph/>
           2. Sharepoint
        4. Grant Admin consent - Requesting Admin consent:
           1. Paste this link on your browser: <https://login.microsoftonline.com/b56bc27e-6da9-4384-9cad-73117b11b0b3/adminconsent?client_id=21954173-f0ec-4d81-943a-f9d91d322713&state=12354&redirect_uri=http%3A%2F%2Flocalhost%3A4321>
           2. Then click access

Redirected link: http://localhost:4321/?admin\_consent=True&tenant=b56bc27e-6da9-4384-9cad-73117b11b0b3&state=12354

https://login.microsoftonline.com/b56bc27e-6da9-4384-9cad-73117b11b0b3/oauth2/authorize?client\_id=21954173-f0ec-4d81-943a-f9d91d322713&response\_type=code&redirect\_uri=http%3A%2F%2Flocalhost%3A4321&response\_mode=query&resource= 21954173-f0ec-4d81-943a-f9d91d322713&state=12345

To get the access token: https://login.microsoftonline.com/b56bc27e-6da9-4384-9cad-73117b11b0b3/oauth2/authorize?client\_id=21954173-f0ec-4d81-943a-f9d91d322713&response\_type=code&redirect\_uri=http%3A%2F%2Flocalhost%3A4321& state=12345

http://localhost:4321/?code=AQABAAIAAAAm-06blBE1TpVMil8KPQ41jr3GSMKFqIqzBtXareMY\_ryZuC75Oa6E0BwyfARMDdBqQCZ3DmOwyTBPQ19etFphQBnGkUwJpUFxBiUIbkP4GRaL\_1KRvHwvDcCWmV32HSLd0P2jPfyu2lOwYGMTVHElkvoS\_tyH\_9boijqSEDj8Kq0xxIY0pDTqKLohQIfoPwZQ3gUkIsgVI8euqA7NOh1rjTAOrxzOFLGzKs70-WvH8BnurPvNJZ13dqMH8KaqSbU-bVHbq8UqpwbUQBhDXiZZjtlOpLHXxpKaP\_7A1UdINFfWqcgtI8dhgmsP-DLNF1MUwnDWJ-TBDOFcbwiD2rI6F1A7Wmt7bRueDmYMAkPPoDno54MVehjTWkDXU1MEqkBmwMEKKpjiZexF4r\_78DstgEVHtqy42xQC2obo2jcjG94y3g5s1zdVu36Sj5pX\_Gm0fyXuc\_O-nQsURN8Hl-PrYjiz-iH9g70hxFYfcGHfaIw1us82xFAa8cQhJEVc6xQSVUZXfO76iff945nK8FDYVUtr9Iu3KUBXpIQUDDJUbfzUxfVdWaZuA2uSRdoe\_fkk\_McrAj3pmy-3IL90jpWxl5dy5qOB4TAJHqZA9BtrbesOw7kMlvu7egoP6i9UVB2CnniYrvKsfS-EKvV1kpuUcBxAGEbWqMa2ERyRh45u7Dm2BthmIf2FkJLUMMxumHiUqSccGZyRIGgSl07myb6KFwh7Tbxm6SdhT\_aTKm4UQmWRL2Eq23GnlZjBHA6zeRY2iQH37l87Zb6ZFqp4-S3dIAA&state=12345&session\_state=ae7a2bc9-8af7-4653-bf58-73e4a9df6b5e

<http://localhost:4321/?code=>

AQABAAIAAAAm-06blBE1TpVMil8KPQ41PL37uy1AjiCIjbZrw8Uch-Yo5p7NDP\_YQV-98lz9BxDb7vHi6UwIcrKKw1Tk-qJGb3metfGv-Q6bltYBP4\_KXSS\_uF5DXCfVodQ\_RzSRtZv3F\_dNYT\_EOihy6VEPpJXvhwi3rFjWM2c2IefgKoZqWjF1uFjOVkiud2T9FrKi\_hczOVLx1ELZZoqnrH88t4DEalMp8kHLjUuGk1VJlfpnbVuTR7yk7RdKVgSn2EqGebrTJKCtkvhKq6MWShzdYmV0K4dPa7\_HSHWdanSWx2qsO3IBuCO4ZkQXCoE79mqlcwtQHzijqUtlPOm2T83oBDoFKpo96ArE4l-aXPH-vByzkcAlJUPXIgLoaAVPEfa1kpYCGRMWEuff2VUr4NFOKVTev80bhrpQqcWoEDSIePS6WebSog3CWHP5IegkKD3oB9VIPm38M\_t3ntmk2g0hR3l0hBHkNDh8koXkBJuH6InuuLe9cv5mIRqPtrcp3uLe2chyvoF474soaq6iBCC4FQ3SrPAkDIgGwGfIZkgfyYeNhSAA&state=12345&session\_state=8f846f2a-9a5e-498f-a0e1-5dda57ce60c8

* Microsoft Graph:
  1. Site\_id: "jaxspdev.sharepoint.com,71ce3e47-95cf-47fa-8483-c1ed8e3b17eb,acda6010-0212-45bc-9b10-cc3b27d57e3c",
  2. App Client id: 1ffb5f51-f2ff-482c-bd76-578ebb88d392
  3. App Name: My React App
  4. Queries: on graph explorer - <https://developer.microsoft.com/en-us/graph/graph-explorer>
     1. Get the site ID
        1. https://graph.microsoft.com/v1.0/sites/jacksonlaboratory.sharepoint.com:/sites/FinancialServices
     2. List all the document libraries under the Finance Services site:
        1. <https://graph.microsoft.com/v1.0/sites/jacksonlaboratory.sharepoint.com:/sites/FinancialServices:/drives>

* + 1. <https://graph.microsoft.com/v1.0/sites/jaxspdev.sharepoint.com:/sites/rqcsite:/drives> list all the document libraries:
       1. "name": "Shared Document",
       2. "webUrl": "https://jaxspdev.sharepoint.com/sites/rqcsite/Shared%20Document",
       3. "driveType": "documentLibrary",
       4. "id": "b!c6\_dRSgaCEWhDp\_ooYpZOSf9PSmooXVOtU4pSfpT8BHL-TGXBnbORoN2\_jb9dEYj",
    2. <https://graph.microsoft.com/v1.0/sites/jaxspdev.sharepoint.com/drives/b!Rz7Occ-V-keEg8HtjjsX6xBg2qwSArxFmxDMOyfVfjwfxVW1KVqCS4YpTePYBj0I/root/children>
       1. List all the children /Shared Documents/
    3. <https://graph.microsoft.com/v1.0/sites/jaxspdev.sharepoint.com/drives/b!Rz7Occ-V-keEg8HtjjsX6xBg2qwSArxFmxDMOyfVfjwfxVW1KVqCS4YpTePYBj0I/items/01XXZICP5ZB7U62JV3Q5FKD7GZT3YRBSHQ/children> or <https://graph.microsoft.com/v1.0/sites/jaxspdev.sharepoint.com/drives/b!Rz7Occ-V-keEg8HtjjsX6xBg2qwSArxFmxDMOyfVfjwfxVW1KVqCS4YpTePYBj0I/root:/models:/children>
       1. List all the children under /Shared Document/models/ -- models id 01XXZICP5ZB7U62JV3Q5FKD7GZT3YRBSHQ
    4. <https://graph.microsoft.com/v1.0/sites/jaxspdev.sharepoint.com/drives/b!Rz7Occ-V-keEg8HtjjsX6xBg2qwSArxFmxDMOyfVfjwfxVW1KVqCS4YpTePYBj0I/root:/models/active_au.json>
    5. Active\_au.json id=01XXZICP3LDPNUSDG5BBHIJU2S3VLXZHUH
    6. Path: /drives/b!Rz7Occ-V-keEg8HtjjsX6xBg2qwSArxFmxDMOyfVfjwfxVW1KVqCS4YpTePYBj0I/root:/models
    7. To get a file content : https://graph.microsoft.com/v1.0/sites/jaxspdev.sharepoint.com/drives/b!Rz7Occ-V-keEg8HtjjsX6xBg2qwSArxFmxDMOyfVfjwfxVW1KVqCS4YpTePYBj0I/items/01XXZICP7HNAS6LDRRPZGLJLIR6IGY5KDM?select=microsoft.graph.downloadUrl
  1. {
  2. "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#sites('jaxspdev.sharepoint.com')/drives('b%21Rz7Occ-V-keEg8HtjjsX6xBg2qwSArxFmxDMOyfVfjwfxVW1KVqCS4YpTePYBj0I')/items/$entity",
  3. "@odata.etag": "\"{E52568E7-318E-4C7E-B4AD-11F20D8EA86C},1\""
  4. }

Notes: To use MS graph api calls that reads file content stored in MS SP document library on the cloud, your APP should implement a two-step process:

1. Use graph api to get the downloadWebURL of the target file
2. Use axios to get the content of the file using the link returned in step 1

Example: to fetch the content of test.req\_loc.json from <https://jaxdev.sharepoint.com/Shared%20Documents/models> -

* Given SP driveId of “Shared Documents” folder is “b!Rz7Occ-V-keEg8HtjjsX6xBg2qwSArxFmxDMOyfVfjwfxVW1KVqCS4YpTePYBj0I”
* Call <https://graph.microsoft.com/v1.0/sites/jaxspdev.sharepoint.com/drives/b!Rz7Occ-V-keEg8HtjjsX6xBg2qwSArxFmxDMOyfVfjwfxVW1KVqCS4YpTePYBj0I/root:/models/test.req_loc.json>
  1. It returns a json response with @microsoft.graph.downloadUrl field which you can use to call axios

{

"@odata.context": "https://graph.microsoft.com/v1.0/$metadata#sites('jaxspdev.sharepoint.com')/drives('b%21Rz7Occ-V-keEg8HtjjsX6xBg2qwSArxFmxDMOyfVfjwfxVW1KVqCS4YpTePYBj0I')/root/$entity",

"@microsoft.graph.downloadUrl": "https://jaxspdev.sharepoint.com/\_layouts/15/download.aspx?UniqueId=8bce2e5c-f6ef-451f-91b9-27e51433d9a2&Translate=false&tempauth=eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0..dXhmM2tPVWN0UWR2bXZHY1hPVUlHakpoOWNRODdRMy9mMDFFVjB5S0h4Zz0&ApiVersion=2.0",

"createdDateTime": "2020-04-25T14:55:32Z",

## Authentication and authorization steps

The basic steps required to configure a service and get a token from the Microsoft identity platform endpoint that your service can use to call Microsoft Graph under its own identity are:

1. Register your app.
2. Configure permissions for Microsoft Graph on your app.
3. Get administrator consent.
4. Get an access token.
5. Use the access token to call Microsoft Graph.
6. See: <https://docs.microsoft.com/en-us/graph/auth-v2-service>
7. <https://docs.microsoft.com/en-us/graph/tutorials/react>
8. <https://github.com/microsoftgraph/msgraph-training-reactspa>
9. <https://github.com/AzureAD/microsoft-authentication-library-for-js/tree/dev/samples/react-sample-app/src>
10. <https://docs.microsoft.com/en-us/azure/active-directory/azuread-dev/v1-protocols-oauth-code>
11. Jwt.ms : access token viewer

CREATING WEBPARTS

Cd to the newly created project directory. In this tutorial, I have created a webpart project called rqcsite. Under the project root directory, we have the following structure:

* README.md
* Config/
  1. config.json
  2. copy-assets.json
  3. deploy-azure-storage.json
  4. **package-solution.json**
  5. serve.json -- this is where you configure connections to local server - port
  6. write-manifests.json
* Dist/
* gulpfile.js
* lib/
* node\_modules/
* package-lock.json
* package.json
* src/
  1. index.ts
  2. webparts/rqcSite
     1. RqcSiteWebPart.manifest.json -- to set the default values of properties
     2. RqcSiteWebPart.ts -- code base for the webpart component
        1. Import property panes that will be used by the webpart fields

import {

IPropertyPaneConfiguration,

PropertyPaneTextField,

PropertyPaneDropdown

} from ‘@microsoft/sp-property-pane’;

* + - * 1. For multi-select fields:
        2. npm install @pnp/spfx-property-controls --save --save-exact
        3. add –

import { PropertyFieldMultiSelect } from '@pnp/spfx-property-controls/lib/PropertyFieldMultiSelect';

* + - 1. Configure the interface of your webpart – set all the fields to associated datatype

export interface IRqcSiteWebPartProps {

description: string;

requestedby: object;

action: string;

requestor: object;

acctunit: string[];

…….

}

* + - 1. Set your interface properties

protected getPropertyPaneConfiguration(): IPropertyPaneConfiguration {

…

groups: [

{

groupName: strings.BasicGroupName,

groupFields: [

PropertyPaneTextField('description', {

label: strings.DescriptionFieldLabel

}),

PropertyPaneDropdown('action', {

label: 'Action',

options: [

{key:'1',text:'New Persmissions'},

{key:'2',text:'Additional Permissions'},

{key:'3',text:'Modification to Permissions'},

{key:'4',text:'Removal of Permissions'}

]

}),

PropertyPaneDropdown('requestor', {

label: 'RQC Requestor'

}),

…

]

}]

}

* + 1. Components
       1. IRqcSiteProps.ts --
       2. RqcSite.tsx
       3. RqcSite.module.scss overview.tsx
       4. RqcSite.module.scss.ts
    2. loc
* teams/
* temp/
* tsconfig.json
* tslint.json

SharePoint Active Sites:

* + <https://jaxspdev.sharepoint.com/sites/rqcsite>
  + <https://jaxspdev.sharepoint.com/sites/rqcsite/_layouts/15/workbench.aspx>
  + <https://portal.azure.com/#home>
  + https://aad.portal.azure.com

Setting up the SSIS backend:

* Install Microsoft Remote Desktop Connection agent from the App store
* Have Chris assign a remote desktop server
* Have Chris setup the initial SSIS project in teams
* Connect to the RD server
  + Mapp the teams project to local H: drive (in team explorer)
  + Open the project’s solution
  + Set the passwords for SQL sever and Oracle db connections
    - SQL Finance\_Integrations staging : svcsql-finance\_etl: LeQ#dT%BMb1x
    - <https://dirtburger.jax.org/SecretView.aspx?secretid=6741>
    - Oracle InforLawson: c0xllawo: v4s\_JLWL
    - <https://dirtburger.jax.org/SecretView.aspx?secretid=3706>

|  |
| --- |
|  |

* + Update the solution.connection manager with the above creadentials
* Create a ssis package for each dataset pipeline
  + Create an SQL task for each dataset to be extracted from lawson - In the task configuration editor, set
    - General -> ResultSet to “Full result set”
    - ResultSet -> ResultName to 0
    - ResultSet -> VariableName.Name to your dataset name
    - ResultSet -> VariableName.Value type to object
  + CREATE A DATA FLOW TASK – then edit to add:
    - A Script Component
      * Inputs and Outputs -> Click on Output 0 and rename it to something relevant
      * Add and configure output columns as needed
      * Then edit the script as needed -See - <https://www.timmitchell.net/post/2015/04/20/using-the-ssis-object-variable-as-a-data-flow-source/>
      * add the following lines:
        + In the header --- using System.Data.OleDb;
        + In the CreateNewOutputRow :

OleDbDataAdapter da = new OleDbDataAdapter();

DataTable dt = new DataTable();

// Extract the data from the object variable into the table

da.Fill(dt, Variables.AccountingUnits);

// iterate over each row in the DataTable, creating a new row in our Data Flow buffer for each

foreach (DataRow dr in dt.Rows)

{

// Create a new, empty row in the output buffer

AuOutputBuffer.AddRow();

// Now populate the columns

AuOutputBuffer.objid = int.Parse(dr["obj\_id"].ToString());

AuOutputBuffer.parentobjid = int.Parse(dr["parent\_obj\_id"].ToString());

AuOutputBuffer.acctunit = dr["acct\_unit"].ToString();

AuOutputBuffer.description = dr["description"].ToString();

* + - * + }
    - An OLE DB Destination component
      * Follow

<https://jaxspdev.sharepoint.com/sites/rqcsite/Lists/Requester%20Access%20Approval/AllItems.aspx>

https://jaxspdev.sharepoint.com/sites/rqcsite/Lists/Requester%20Access%20Approval/DispForm.aspx?ID=1&e=P8nJW7

Setting up my Mac:

* Map my network H: drive on to my mac
  + Open “finder” -> Go -> Connect to Server
  + Enter “smb://bhhdrive.jax.org/hutchl”
  + Then click “connect”
  + If not already logged in - Enter your login details and password then click OK to mount the network drive
* Make your map persistent after reboot
  + Hit the Apple menu, then System Preferences > Users & Groups
  + From here, select Login Items and click + to add a new item
  + Find your network drive and click Add, then close the window
* Install Oracle SQL developer
  + <https://developer.oracle.com>
    - [Lucie.hutchins@jax.org](mailto:Lucie.hutchins@jax.org)
    - Nkou&2004
  + Install:
    - <https://docs.oracle.com/cd/E12151_01/doc.150/e12153/install.htm#CIHFHIAH>

See:

1. <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/set-up-your-development-environment>
2. <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/get-started/build-a-hello-world-web-part>
3. Starting apache - <https://medium.com/better-programming/install-apache-mysql-php-macos-mojave-10-14-b6b5c00b7de>
4. **Check for open TCP ports using Network Utility on Mac**
   1. <https://support.apple.com/lt-lt/guide/mac-help/mchlp1710/mac>
   2. <https://wilsonmar.github.io/ports-open/>
5. <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/get-started/connect-to-sharepoint>
6. <https://docs.microsoft.com/en-us/office/developer-program/build-office-365-solutions>
7. <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/set-up-your-developer-tenant>
8. <https://social.msdn.microsoft.com/Forums/sqlserver/en-US/60baba9d-6dab-43c2-8738-a80c448c3b3c/forum-faq-how-do-i-send-multiple-rows-returned-by-execute-sql-task-as-email-content-in-sql-server?forum=sqlintegrationservices>
9. <https://docs.microsoft.com/en-us/sql/integration-services/map-query-parameters-to-variables-in-an-execute-sql-task?view=sql-server-2014>
10. <https://www.sqlshack.com/execute-sql-tasks-in-ssis-output-parameters-vs-result-sets/>
11. <https://support.microsoft.com/en-us/help/2009672/you-may-get-unable-to-prepare-the-ssis-bulk-insert-for-data-insertion>
12. <https://docs.microsoft.com/en-us/sql/integration-services/extending-packages-scripting-data-flow-script-component-types/creating-a-source-with-the-script-component?view=sql-server-ver15>
13. <https://www.timmitchell.net/post/2015/04/20/using-the-ssis-object-variable-as-a-data-flow-source/>
14. <https://www.cozyroc.com/ssis/sharepoint>
15. <https://www.c-sharpcorner.com/article/single-page-app-in-sharepoint/>

Tutorials:

1. Web-part – hello world 1 - <https://www.youtube.com/watch?v=_O2Re5uRLoo&feature=emb_rel_end>
2. Web-part – hello world 2 - <https://www.youtube.com/watch?v=4F65TmsHucY>
3. Web-part – hello world 3 - <https://www.youtube.com/watch?v=CMJEp8TS4aU>
4. Web-part – hello world 4 - <https://www.youtube.com/watch?v=wl2ur1SBp3Q>
5. SharePoint Framework Training - Using React and Office UI Fabric React Components
   1. <https://www.youtube.com/watch?v=TlSGdDZmrTM>
   2. <https://docs.microsoft.com/en-us/learn/paths/m365-sharepoint-associate/>
   3. <https://github.com/SharePoint/sp-dev-training-spfx-react-fabric/tree/master/Demos/01-webpart>
6. **SharePoint Framework Tutorial - Using Microsoft Graph APIs**
   1. <https://www.youtube.com/watch?v=tHzbh5JoC-A>
   2. <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/get-started/using-microsoft-graph-apis>
7. <https://sharepoint.github.io/sp-dev-fx-property-controls/controls/PropertyFieldMultiSelect/>
8. <https://sharepoint.github.io/sp-dev-fx-property-controls/controls/PropertyFieldListPicker/>
9. <https://n8d.at/how-to-use-bootstrap-in-sharepoint-framework-projects/>
10. <https://docs.microsoft.com/en-us/graph/overview>
11. <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/get-started/connect-to-sharepoint>
12. <https://docs.microsoft.com/en-us/azure/active-directory/develop/msal-js-initializing-client-applications>
13. <https://github.com/pnp/sp-dev-fx-webparts>
14. <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/use-aad-tutorial> (Consume the Microsoft Graph in the SharePoint Framework
15. )

TITLE: COZYROC SSIS+

------------------------------

QC REPORTS

* List of location and active requesters ( requester, location)
* List of active requester and associated location

App’s Specs clarification.

**Accounting Units: What is the Valid range ?**

1. Include AU with/without activities –
2. The AU query should include income statement account=2 ( chart\_section=2) condition
3. The accounting units visibility:
   * **If an AU does not have a budget manager, then don’t list**

* ~~Supervisor – not a Budget Manager or FA~~
  + ~~Only AUs not associated with a Budget manager?~~
* **Budget manager** 
  + **Only the AU they are assigned**
* **Financial Analyst**
  + **Only the AUs they are assigned to**

1. Is there a quota on the number of requestors per:
   * Location ?
   * Accounting unit?
   * …?

**Requestor eligibility:**

1. Only requestors who are employees – found in employee table?
2. Employee should be in
   * Category –
     1. CO-OP ASSOCIATE
     2. COLLABORATOR
     3. CONTINGENT WORKER
     4. JAX SHANGHAI
     5. HHMI STAFF
     6. POST BAC RESEARCHER
     7. CONTRACTOR
     8. EMERITUS
     9. POSTDOC
     10. RETIREE
     11. STUDENT
     12. TRAINEE
     13. **EMPLOYEE**
     14. JOINT FACULTY
     15. SUMMER STUDENT
     16. PREDOC
   * Status
     1. MIL LEAVE-ACTIVE
     2. FOREIGN PAYROLL
     3. JW MED LEAVE
     4. ~~INACTIVE~~
     5. ~~TERMINATED~~
     6. ~~TRAINING ENDED~~
     7. GRADUATED
     8. ~~WC LEAVE~~
     9. **ACTIVE**
     10. ~~EM LEAVE~~
     11. ADJUNCT STAFF
     12. ~~EMERITUS - RETIRED~~
     13. ~~TERM FINAL~~
     14. ~~DECEASED~~
     15. ~~RETIRED~~
     16. ~~PER LEAVE~~
     17. ~~TERMINATED FINAL~~
   * Type
     1. **REGULAR FULL TIME**
     2. PART TIME HOURLY
     3. ON-SITE AFFILIATE
     4. TEMPORARY
     5. REGULAR PART TIME
     6. REMOTE AFFILIATE
3. Physical location: Should we remove this?
4. Start and End date fields will be added
5. Other requester
6. At submission =>
   * if activity send email to [spa@jax.org](mailto:spa@jax.org)
   * Store the info