**Sharepoint Frame work**

**Ref :-**[**http://www.qipoint.com/blog/getting-started-with-sharepoint-framework-spfx/**](http://www.qipoint.com/blog/getting-started-with-sharepoint-framework-spfx/)

SharePoint Framework brings the new SharePoint UX, using open source tools like Yeoman scaffolding, Gulp and REST-based APIs for building SharePoint web parts. It allows any Java script frameworks like React JS, Angular or Knockout JS.

It provides:

* Full client-side development
* Development using open source technology
* Your choice of host
* SPFx components run in the current user context, which improves the load time,SharePoint Framework app are already authorized with the currently logged in users credentials and permissions.
* Supports development on modern pages as well as traditional classic and publishing pages
* Full-trust mode
* Responsive design
* Node.js—Node.js is an open source, cross-platform runtime environment to develop applications. You won’t code in Node.js, but you will use NPM (Node package manager) to deploy the various packages in your project. You’ll use NPM to get the pre-built packages to use in your application like Angular, Jquery, and other similar apps.
* Yeoman—Yeoman is a template engine that runs on top of Node.js. You’ll use the SharePoint framework just to create your Sharepoint project structure and then forget about it. The out of the box SharePoint Yeoman Generator for SharePoint Framework has two templates one for React and another for Knockout.
* Gulp—Gulp is a task runner to compile Typescript or lesser files and to run the automated processes. Again, this is also built on top of node.js. You use this to compile Typescript files, bundle the files into a package and upload to SharePoint or CDN.
* Webpack—Webpack is the build tool that bundles your JS, CSS, Image and fonts file into a single big file, which helps to increase performance and manageability. SharePoint framework uses Webpack to bundle the source code files. Like Yeoman, once installed, you can ignore Webpack since you don’t have to explicitly install it. It will remain available as default scaffolding from the Yeoman template.
* Typescript—Typescript is an open source language Microsoft created. Typescript executes in Node.js and is suitable for the client- and server-side applications. Typescript is a super-set of JavaScript. So a valid JavaScript code is a valid Typescript code but not the other way around. Typescript is something you have to be very well versed in since most of  development uses Typescript.

   
You can also choose any of the JavaScript frameworks like React, AngularJS, KnockoutJS, among others. **Microsoft doesn’t recommend which framework to use**. **However, Microsoft uses ReactJS for its customizations in SharePoint**. The only one that doesn’t work well is Angular 2 because of its architecture. It works with one web part on the page but not on multiple web parts on the page, which is more of an Angular 2 issue and not a Microsoft issue.

SET UP

Ref video :- [**https://www.youtube.com/watch?v=\_fxYexlUhe0**](https://www.youtube.com/watch?v=_fxYexlUhe0)

**Install Node JS**[**https://nodejs.org/en/**](https://nodejs.org/en/)

Check version npm –v 3.10.10 (package manager for nodejs)

**Install Visual StudioCode (source code editor- development tooling)**

<https://code.visualstudio.com/docs/?dv=win>

**Install Yoemen**(template manager for the templates) and **gulp** (task manager)

npm install –g yo gulp

**Spframework specific tooling** pull down all spfx client side dev specific tooling, templates , dependencies, pulling the need set up and getting installed in the machine

npm install –g @microsoft/generator-sharepoint

UPDATING CONFIGURATION PANE  AND TAKING WITH SHAREPOINT (Using REST API)

Ref Videos:

HowTo Create a new Project:

<https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/get-started/build-a-hello-world-web-part>

<https://www.youtube.com/watch?v=QbDtsMg88Js>

[https://www.youtube.com/watch?v=9VMwjb2pbQ8](https://www.youtube.com/watch?v=9VMwjb2pbQ8&t=14s)

Run Sharepoint template system

yo @microsoft/sharepoint

Install a certificate to this machine, to be able to serve content form https

gulp trust-dev-cert

Then we can test the webpart by giving below

gulp serve

 it will start a browser in the local host using the local workbench

<https://localhost:4321/temp/workbench.html>

Cntrl + C to stop (node.js is no longer serving)

To look at code open visual studio code by giving the below command.

code .

Online work bench can be accessed in the below url

<https://hansaplusnet.sharepoint.com/sites/jajdev/_layouts/15/workbench.aspx>

When talking with SharePoint  , Local work bench don’t have access to sharepoint data so we have to mock data

HOST TO SHAREPOINT PAGE

Creating a package gulp package-solution

The package extension is .sppkg

Drag and drop to App catalog.

DEPLOY TO SHAREPOINT/AZURE

Ref Videos :- <https://www.youtube.com/watch?v=dxD0lgYUCI4>

We need to enable Office 365 Public CDN and create a Public CDN (Administrator enables Office 365 Public CDN for the tenant using SharePoint Online Management Shell.)

Steps can be found in the link:-

<https://dev.office.com/blogs/general-availability-of-office-365-cdn>

**For hansaplusnet below  public CDN is created,**

<https://publiccdn.sharepointonline.com/hansaplusnet.sharepoint.com/sites/cdn/cdn/>

* Update the public cdn in the manifest.json
* Then gulp --ship making solution ready for packaging
* gulp package-solution --ship
* Copy the files in the path 'webpart path\temp\deploy' to the cdn library
* Copy the package in the path 'webpart path\sharepoint\solution\\*.sppkg' to the appcatalog and trust

So after each change and packaging you get a new version of the js file in the deploy folder, copy all of them to the cdn library and new pacakge to appcatalog.

For AZURE , we will use an **Azure Storage account integrated with CDN** to deploy our assets.