An Introduction to Development with the SharePoint Framework (SPFx)

Rob Windsor
rwindsor@paitgroup.com
@robwindsor
About Me

▪ Lead SharePoint Consultant at PAIT Group
▪ Chief Imagineer at extaCloud
▪ Technical Contributor to the Pluralsight On-Demand Library
▪ Microsoft MVP, MCPD, MCT Alumni
▪ Founder and Past-president of the North Toronto .NET UG
▪ Co-author of Prof. Visual Basic 2012 and .NET 4.5 (Wrox)
SharePoint Framework (SPFx)

Modern client-side development

Lightweight web and mobile

Powers our own experiences

Backward compatible

Supports open source tools and JavaScript web frameworks

SharePoint framework

Microsoft Graph

LOB Systems and Cloud Services
Open Source Tooling

Build Process & Tooling
- Yeoman Templates
- Gulp-based Build Process
- NodeJS/NPM
- SystemJS
- Webpack
- SASS
- TypeScript

JavaScript Frameworks
- React
- Angular
- Knockout
- and more ... your choice!

Code Editors
- Visual Studio Code
- Atom
- Sublime
- and more ... your choice!
Server-side Tool Comparison

IIS Express

.NET

NuGet

MSBuild

C#

Visual Studio Project Templates

node

npm

Gulp

TypeScript

Yeoman
Visual Studio?

- Visual Studio extension for SharePoint Framework projects
  - [https://github.com/SharePoint/sp-dev-fx-vs-extension](https://github.com/SharePoint/sp-dev-fx-vs-extension)
Client-side Web Parts

They are still web parts!

Built for the modern, JavaScript-driven web.

Runs directly inside a SharePoint Page.
SharePoint Workbench

Developer tool to debug web parts
Local development experience
Test your changes locally
SharePoint Framework available offline
Available in SharePoint Online as well
Developer Preview - SharePoint Framework Extensions

- **Application customizer**
  - Embed HTML into well-known elements
  - Embed JavaScript
  - Embed done at site collection, site, or list level

- **Command Set customizer**
  - Custom actions
  - Add menu items to command bar or ECB menu

- **Field customizer**
  - JSLink like functionality
Developer Preview - SharePoint Framework Extensions

- Application Customizer
- Command Set customizer
- Field customizer
Setting Up Your Development Environment

- You don’t need to have SharePoint installed locally
- Setup steps for Windows:
  - Install NodeJS
  - Install Yoeman and Gulp
  - Install the Yoeman SharePoint Project Generator
  - Install Visual Studio Code
- Full details at https://dev.office.com/sharepoint/docs/spfx/set-up-your-development-environment
Setting Up Your Office 365 Developer Tenant

- Create an App Catalog
- Create a site collection based on the Developer Site template

- No longer need to deploy SharePoint Workbench
  - It’s now in the layouts folder
  - /_layouts/workbench.aspx

- Full details at https://dev.office.com/sharepoint/docs/spfx/set-up-your-developer-tenant
Create a Client-side Web Part

- Create the web part project
  - yo @microsoft/sharepoint
- Preview the web part
  - gulp serve
public render(): void {
    this.domElement.innerHTML = `"<div class="${styles.helloWorld}"
        <div class="${styles.container}"
            <div class="ms-Grid-row ms-bgColor-themeDark ms-fontColor-white ${styles.row}"
                <div class="ms-Grid-col ms-u-lg10 ms-u-xl8 ms-u-xlPush2 ms-u-lgPush1"
                    <span class="ms-font-xl ms-fontColor-white">Welcome to SharePoint!</span>
                    <p class="ms-font-l ms-fontColor-white">Customize SharePoint experiences using Web Parts.</p>
                    <p class="ms-font-l ms-fontColor-white">${escape(this.properties.description)}</p>
                    <a href="https://aka.ms/spfx" class="${styles.button}"
                        <span class="${styles.label}">Learn more</span>
                    </a>
                </div>
            </div>
        </div>";
Context Object

- Provides access to information about current context
- Greatly extended version of _spPageContextInfo
private getLists(): Promise<ISPLists> {
    let result: Promise<ISPLists>;

    if (Environment.type === EnvironmentType.Local) {
        let listData: ISPLists = {
            value: [
                { Title: "List 1", Id: "1" },
                { Title: "List 2", Id: "2" },
                { Title: "List 3", Id: "3" }
            ]
        };
        result = new Promise<ISPLists>((resolve) => {
            resolve(listData);
        });
    } else {
        let webUrl = this.context.pageContext.web.absoluteUrl;

        result = this.context.spHttpClient.get(webUrl + "/_api/web/lists?$filter=(Hidden eq false)", SPHttpClient.configurations.v1)
            .then((response) => {
                return response.json();
            });
    }
}
public render(): void {
    this.domElement.innerHTML = `...`;

    let webPart: HelloWorldWebPart = this;
    let button: Element = this.domElement.querySelector('#getListsButton');
    button.addEventListener('click', () => { webPart.getListsButtonClickHandler(); });
}

private getListsButtonClickHandler(): void {
  var call = this.getLists();
  call.then((response) => {
    this.renderLists(response);
  });
}
**Custom Web Part Properties**

**IHelloWorldWebPartProps.ts**

```typescript
export interface IHelloWorldWebPartProps {
    description: string;
    color: string;
}
```

**HelloWorldWebPart.ts**

```typescript
protected getPropertyPaneConfiguration(): IPropertyPaneConfiguration {
    return {
        pages: [{
            header: { description: strings.PropertyPaneDescription },
            groups: [{
                groupName: strings.BasicGroupName,
                groupFields: [
                    PropertyPaneTextField('description', {
                        label: strings.DescriptionFieldLabel
                    }),
                    PropertyPaneDropdown('color', {
                        label: "Color",
                        options: [
                            { key: "Red", text: "Red" },
                            { key: "Green", text: "Green" },
                            { key: "Blue", text: "Blue" }
                        ]
                    })
                ]
            }]
        }]
    }
}
```

**HelloWorldWebPart.manifest.json**

```json
{
    "$schema": "../../../node_modules/@microsoft/sp-module-interfaces/lib/manifestSchemas/jsonSchemas/clientSideComponentManifestSchema.json",
    "id": "4ff90fbd-2134-4fa7-a595-f292f3ce7b9b",
    "alias": "HelloWorldWebPart",
    "componentType": "WebPart",
    "version": "0.0.1",
    "manifestVersion": 2,
    "preconfiguredEntries": [{
        "groupId": "4ff90fbd-2134-4fa7-a595-f292f3ce7b9b",
        "group": { "default": "Under Development" },
        "title": { "default": "HelloWorld" },
        "description": { "default": "HelloWorld description" },
        "officeFabricIconFontName": "Page",
        "properties": {
            "description": "HelloWorld",
            "color": "Blue"
        }
    }]
}
```
Debugging

- Browser debugging tools support TypeScript debugging

- Debugging in Visual Studio Code
  - Requires extension and configuration
  - Only works with Chrome
  - [https://dev.office.com/sharepoint/docs/spfx/debug-in-vscode](https://dev.office.com/sharepoint/docs/spfx/debug-in-vscode)
Using jQuery (or other External Libraries)

- Ensure Typings TypeScript Definition Manager is installed
  - npm install typings --global

- Add TypeScript definition for jQuery
  - typings install dt~jquery --global --save

- Add external reference in config/config.json
  - "jquery": "https://code.jquery.com/jquery-2.1.1.min.js"

- Import jQuery in your web part
  - import * as jQuery from 'jquery';

- For full details see https://dev.office.com/sharepoint/docs/spfx/web-parts/basics/add-an-external-library
Deploying Your Web Part: Configuring a CDN

- Your web part files will need to be deployed to a CDN
- Using a SharePoint Online document library
  - I’m using this method in this talk
- Using Windows Azure
  - https://dev.office.com/sharepoint/docs/spfx/web-parts/get-started/deploy-web-part-to-cdn
Deploying a Client Web Part - 1

- Add CDN URL to your web part project
  - https://publiccdn.sharepointonline.com/<tenant name>/sites/site/library/folder
  - config/write-manifest.json
- Update the package name (optional)
  - config/package-solution.json
- Create the deployment bundle
  - Gulp bundle --ship
- Copy deployment files to CDN
  - temp/deploy
Deploying a Client Web Part - 2

- Create deployment package (app package)
  - gulp package-solution -ship
- Add app package to App Catalog
  - sharepoint/solution
- Add the App to root site of site collection
  - Adds web part to web part gallery
- Add the web part to a page
Deleting a Project Folder

- Node module folder names become very long due to nesting
- Length of these folder names can be problematic on Windows machines
- Windows will not fully delete project folder

- You can use rimraf to delete node module folders

- For full details see http://www.nikola-breznjak.com/blog/javascript/nodejs/how-to-delete-node_modules-folder-on-windows-machine/
Resources

- **Office Developer Center**
  - https://dev.office.com/sharepoint

- **GitHub**
  - http://github.com/sharepoint

- **Microsoft Ignite 2016 videos**
  - https://www.youtube.com/channel/UCrhJmfAGQ5K81XQ8_od1iTg/search?query=sharepoint+framework

- **Build 2017 videos**

- **Waldek Mastykarz blog**
  - https://blog.mastykarz.nl/
Thank You

- Big thanks to the organizers, sponsors and you for making this event possible
- Please fill out your evaluation
- Please keep in touch

rwindsor@paitgroup.com
@robwindsor
blogs.msmvps.com/windsor