



**SharePoint
CONFERENCE
NORTH AMERICA**

Advanced SharePoint Framework Strategies

Beyond the basics

Eric Overfield



Eric Overfield

President, PixelMill
Microsoft Regional Director
Microsoft MVP – Office Servers and Services

@ericoverfield
ericoverfield.com

Top priorities

Versioning and ALM

Graph client and Azure Functions

Beyond bundling

Scoped deployment

Best practices and advise

A photograph of a large tree from a low angle, looking up at its canopy. The tree has many branches, some bare and some bearing clusters of leaves. The leaves are a mix of vibrant red and bright green, suggesting a transition between seasons or a healthy, diverse ecosystem. The background is a clear, pale blue sky.

Embrace the Change

Controlling nodeJS versions

SPFx based on nodeJS LTS

As SPFx keeps evolving, so too will your version of nodeJS

What about existing SPFx solutions?

Existing SPFx projects may not work with new nodeJS

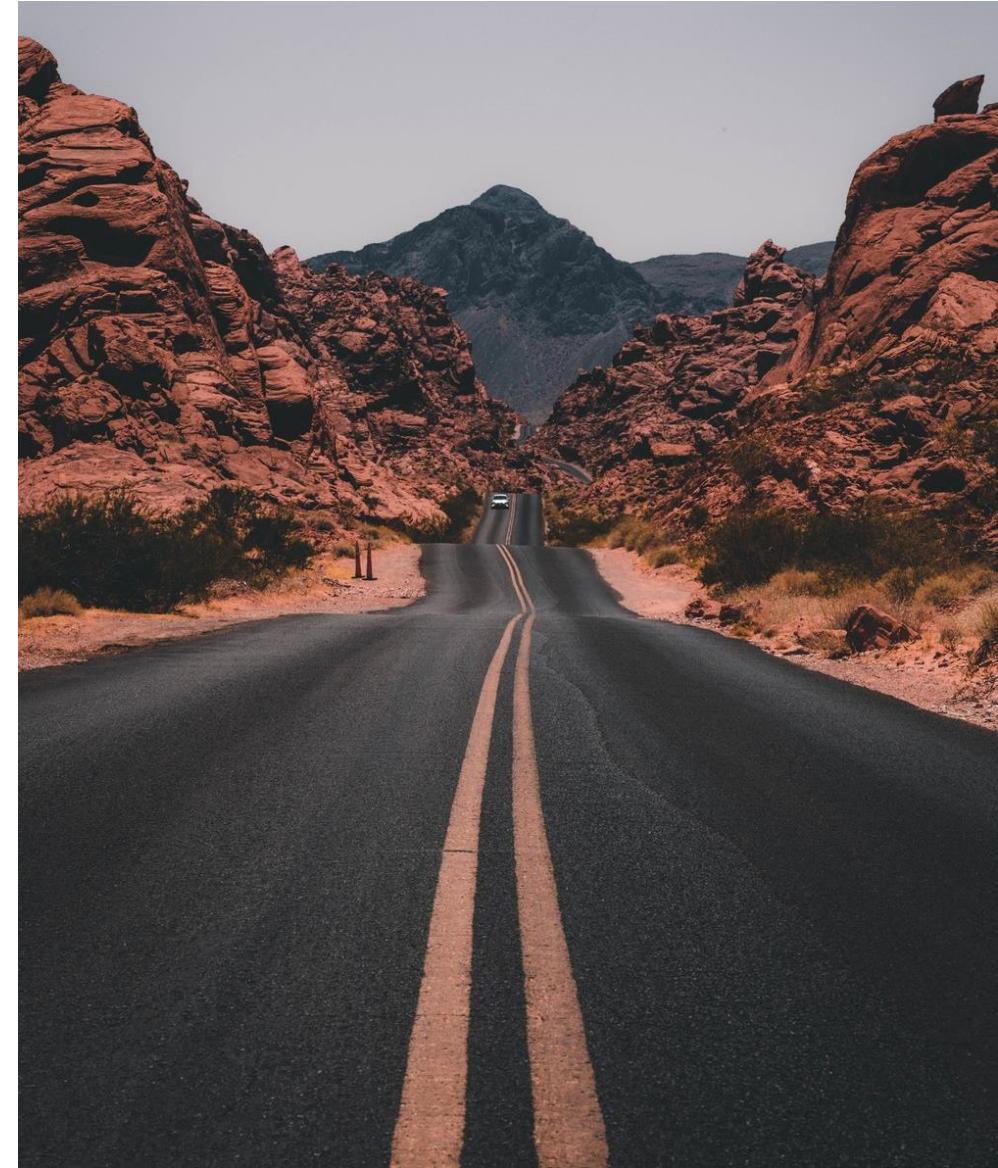
Or other way around

Solution: Maintain multiple versions

Use tools such as nvm to maintain multiple versions of nodeJS

<https://github.com/creationix/nvm> (Linux/Mac)

<https://github.com/coreybutler/nvm-windows> (Windows)



Localize Yeoman SharePoint Generator

Which generator did you use?

SPFx Yeoman generator normally installed globally

Control which version of generator used project to project

Install local generator

```
npm init
npm install @microsoft/generator-sharepoint --save-dev
npm install @microsoft/generator-sharepoint@1.3.4 --save-dev
```

SharePoint Yeoman Generator

<https://www.npmjs.com/package/@microsoft/generator-sharepoint>

```
λ yo @microsoft/sharepoint

  _-----_
  |  _  _  |
  | - (o) - |
  |  \_ `U` _ /
  | / __A__ \ \
  | |   ~   | |
  | .   .   . |
  | | o | ' Y |
  |-----|-----|
```

Welcome to the
SharePoint Client-side
Solution Generator

Let's create a new SharePoint solution.

- ? What is your solution name? app-extension
- ? Which baseline packages do you want to target?
- ? Where do you want to place the files? Use the default path
- ? Do you want to allow the tenant admin the ability to manage or adding apps in sites? No
- ? Which type of client-side component to create?
- ? Which type of client-side extension to create?

Demo: Versioning nodeJS and Yeoman Generator

Dependency Management with Shrink Wrap

Get comfortable with package.json

Project dependencies

Maintained by package.json

Client execution dependencies

"dependencies" section

```
npm install --save --save-exact
```

Develop / build dependencies

"devDependencies" section

```
npm install --save-dev --save-exact
```

Dependencies management

```
npm install
```



```
{} package.json x
1  {
2    "name": "hello-world-web-part-demo",
3    "version": "0.0.1",
4    "private": true,
5    "engines": {
6      "node": ">=0.10.0"
7    },
8    "scripts": {
9      "build": "gulp bundle",
10     "clean": "gulp clean",
11     "test": "gulp test"
12   },
13   "dependencies": {
14     "react": "15.6.2",
15     "react-dom": "15.6.2",
16     "@types/react": "15.6.6",
17     "@types/react-dom": "15.5.6",
18     "@microsoft/sp-core-library": "~1.4.1",
19     "@microsoft/sp-webpart-base": "~1.4.1",
20     "@microsoft/sp-lodash-subset": "~1.4.1",
21     "@microsoft/sp-office-ui-fabric-core": "~1.4.1",
22     "@types/webpack-env": ">=1.12.1 <1.14.0"
23   },
24   "devDependencies": {
25     "@microsoft/sp-build-web": "~1.4.1",
26     "@microsoft/sp-module-interfaces": "~1.4.1",
27     "@microsoft/sp-webpart-workbench": "~1.4.1",
28     "gulp": "~3.9.1",
29     "@types/chai": ">=3.4.34 <3.6.0",
30     "@types/mocha": ">=2.2.33 <2.6.0",
31   }
32 }
```

Control dependency versions with shrinkwrap

Dependency float an issue with production projects

--save-exact only addressed top-level dependencies

Do not check in / commit node_modules!

Shrinkwrap project as soon as created

/>npm shrinkwrap

Creates npm-shrinkwrap.json

Locks down node_modules

Provides ability for 100% reproducible project / build

Manage npm-shrinkwrap.json

Commit this file once per release at minimum

Update by re-running />npm shrinkwrap

The screenshot shows a portion of the npmjs.com website. At the top, there's a red header bar with the 'npmjs.com' logo. Below it, the main content area has a white background. On the left, there's a sidebar with links to 'Getting started', 'Private packages', 'Troubleshooting', 'Using npm', 'CLI commands', 'Configuring npm' (which is currently selected, indicated by a red border), and 'folders', 'npmrc', 'package-lock.json', 'package-locks', 'package.json', and 'shrinkwrap.json'. Below this is a section for 'npm, Inc.' with a link to 'view all on one page'. The main content area to the right is titled 'npm-shrinkwrap.json' and has a subtitle 'A publishable lockfile'. It includes a 'Description' section with a note about the file being a lockfile and its relationship to package-lock.json. There's also a 'See Also' section at the bottom.

npmjs.com

npm

Getting started
Private packages
Troubleshooting
Using npm
CLI commands
Configuring npm
folders
npmrc
package-lock.json
package-locks
package.json
shrinkwrap.json

npm, Inc.

[view all on one page](#)

npm-shrinkwrap.json

A publishable lockfile

Description

npm-shrinkwrap.json is a file created by [npm-shrinkwrap](#).json, with one major caveat: Unlike package-lock.json, it may be included when publishing a package. The recommended use-case for npm-shrinkwrap.json is through the publishing process on the registry: for example, tools intended as global installs or devDependencies. Library authors to publish this file, since that would prevent over-transitive dependency updates.

Additionally, if both package-lock.json and npm-shrinkwrap.json are present in a package root, package-lock.json will be ignored.

For full details and description of the npm-shrinkwrap.json file, see the manual page for [package-lock.json](#).

See Also

Demo: package.json and
npm shrinkwrap

The New SPFx Graph Client

And how to use it

CONVERSATIONS

FILES

GROUPS

NOTES

TASKS

TASKS

NOTES



memberOf

workingWith

createdBy

EVENTS

USER

trendingAround



manager

MESSAGES

MESSAGES



Shared with me

workingWith

DirectReports

FILES



createdBy

public



modifiedBy



USER



trendingAround

FILES



SPFx v1.4.1 Includes MSGraphClient in Preview

Native access to Microsoft Graph with expanded permissions

MSGraphClient (in preview ~May 2018) found in @microsoft/sp-client-preview

AadHttpClient found in @microsoft/sp-http (Access any AAD secured endpoint)

Requesting permissions to Graph API

Set required permissions in webApiPermissionRequests properties in ./config/package-solution.json

Currently still in preview ~May 2018

Setting permissions does not mean permission is granted

Tenant administrator must approve requested permissions to Graph from SPFx during deployment

Use VSCode to debug REST calls

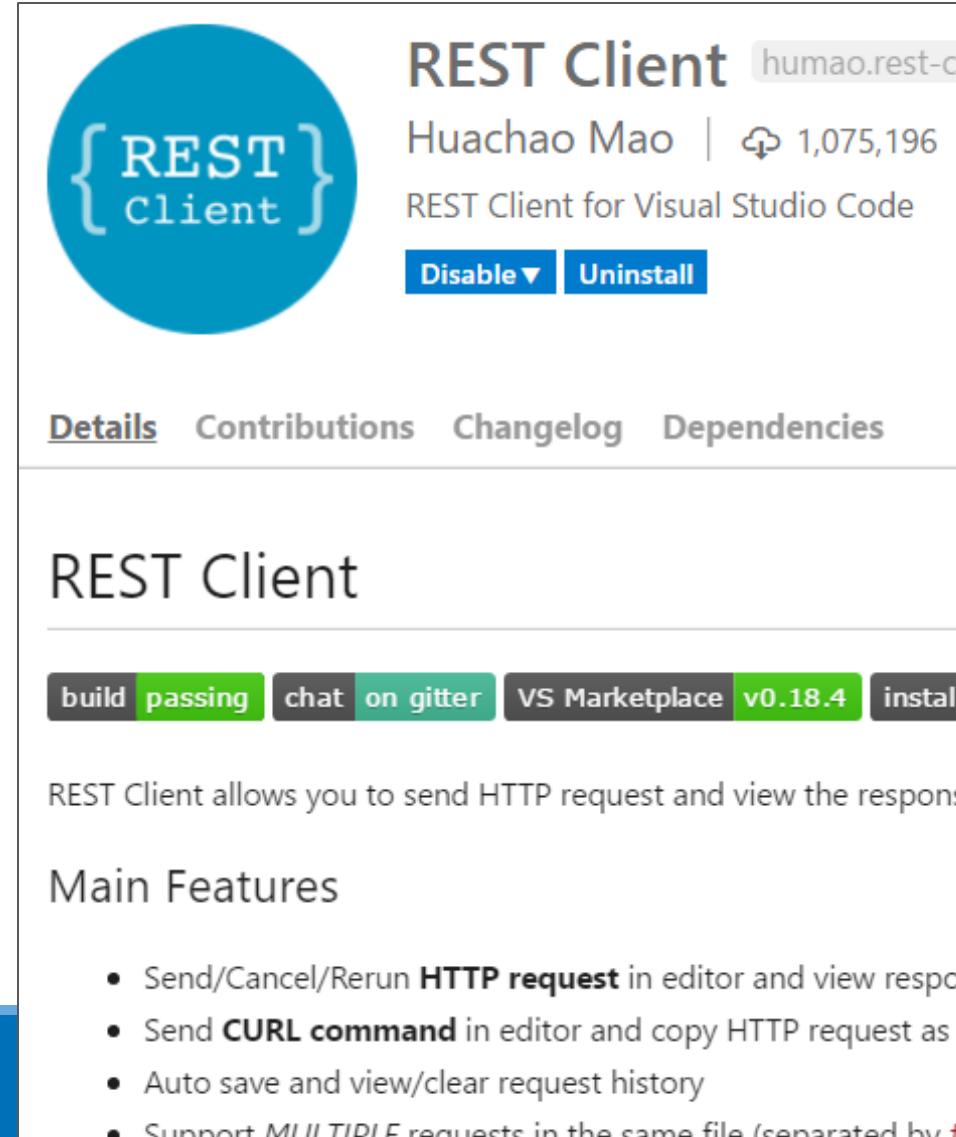
Debug SharePoint REST quickly with VSCode

REST Client extension enabled quick REST calls

Native support of .http and .rest file extensions

REST Client extended with SP REST Client

SP Rest Client extension assists with SharePoint specific Rest calls

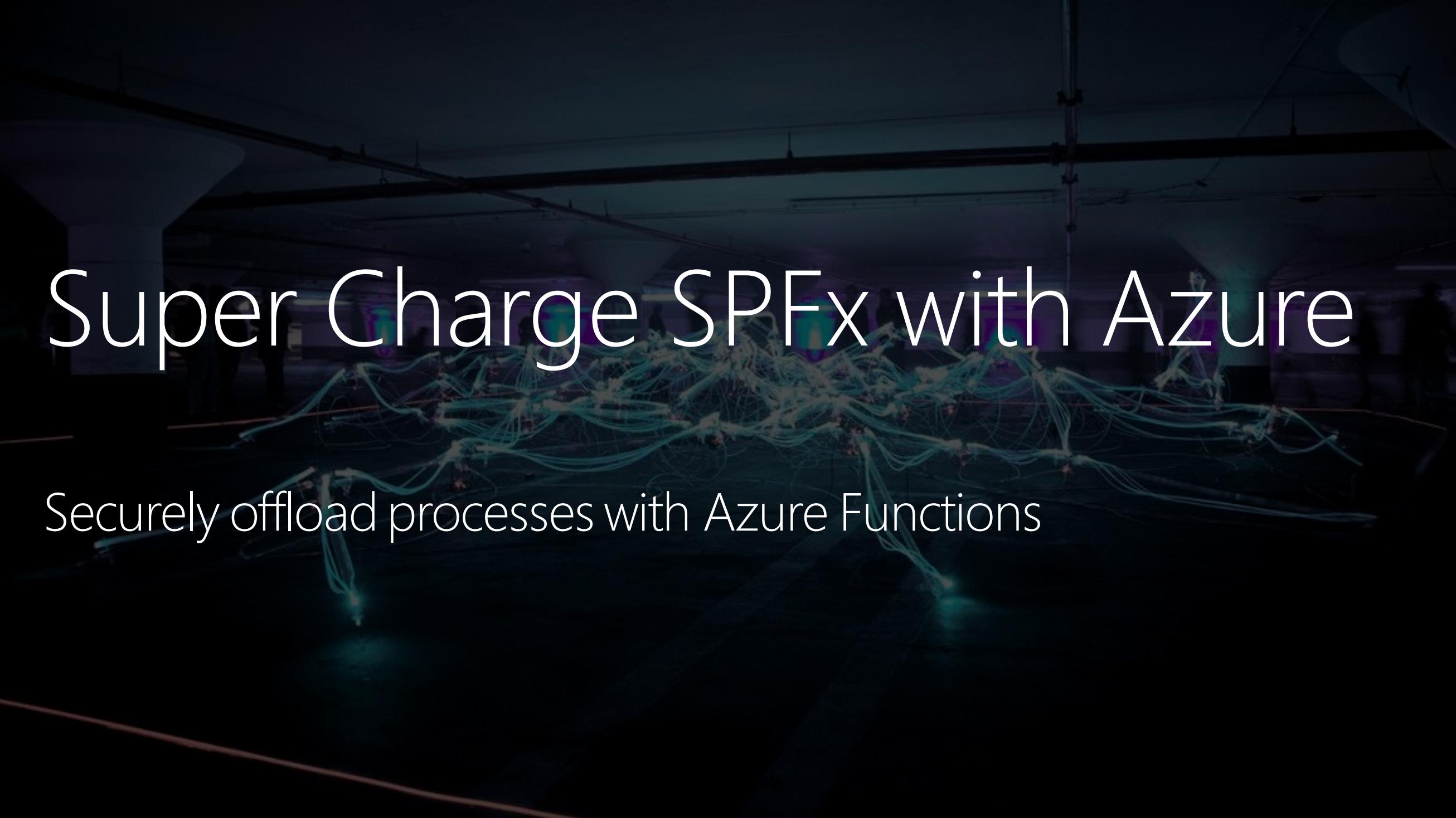


The screenshot shows the REST Client extension page in the VS Code Marketplace. The extension icon is a blue circle with white text '{ REST Client }'. The title is 'REST Client' by 'humao.rest-client'. It has 1,075,196 installs. Below the title, there are buttons for 'Disable' and 'Uninstall'. At the bottom, there are links for 'Details', 'Contributions', 'Changelog', and 'Dependencies'. The main content area is titled 'REST Client' and includes status indicators for 'build' (passing), 'chat' (on gitter), 'VS Marketplace' (v0.18.4), and 'install'. A description states: 'REST Client allows you to send HTTP request and view the response in the editor.' The 'Main Features' section lists:

- Send/Cancel/Rerun **HTTP request** in editor and view response
- Send **CURL command** in editor and copy HTTP request as curl command
- Auto save and view/clear request history
- Support **MULTIPLE** requests in the same file (separated by blank lines)

Demo: GraphClient and
Debug REST calls with VSCode

Super Charge SPFx with Azure



Securely offload processes with Azure Functions

Securely enable backend LOB processes

Secure custom simple to complex process with AAD

Security model to access custom coded components

Integrate with external data

Possible migration path for full trust solutions to SharePoint Online

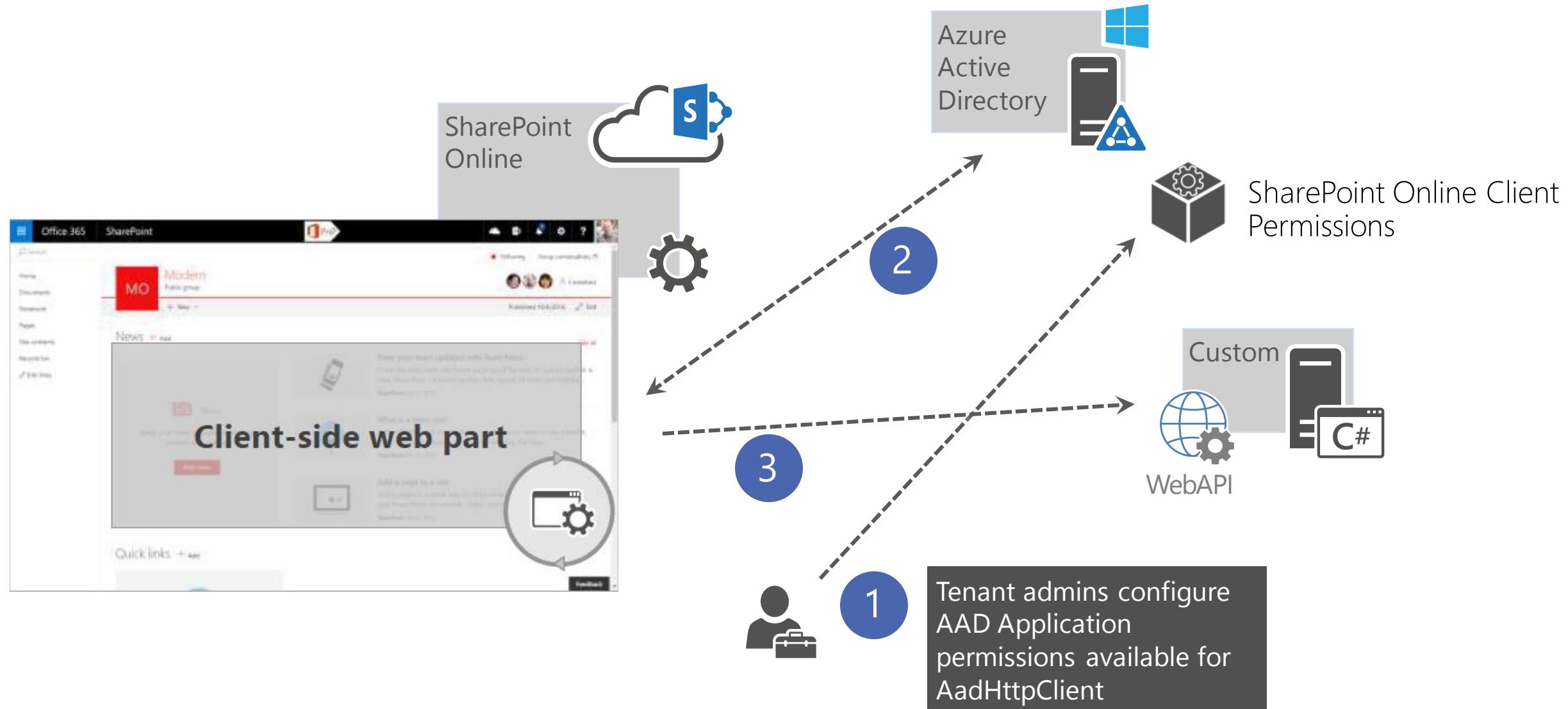
Refactor full trust code to Azure / Azure Functions / Azure based Web API

Secured by AAD

API's hosted in different AAD requires admin consent

CORS also needs to be configured for cross domain requests.

Calling external processes with AAD



Demo: Securely calling an Azure Function

Bundling

Control your solution deployments



Production builds require proper bundling

Default build targets DEBUG. Target RELEASE with --ship parameter
Minifies bundle, places assets in ./temp/deploy

Load 3rd party libraries from CDN's as often as possible

By default, 3rd party libraries are added to bundle

Each bundle loaded once per webpart on the page. 5 of the same webparts on a page == 5 bundles

Update ./config/config.json to use CDN's

Use Webpack Bundle Analyzer to gain insights into bundle

```
/>npm install webpack-bundle-analyzer --save-dev
```

Must also update gulpfile.js to create bundle analysis

Demo: Optimize Bundles for Production

Scope Deployments

Using Site Collection App Catalogs and
SharePoint Online Management Shell

Scope SPFx to specific site collections

Limit SPFx bundles to specific site collections

Control who has access to what

Use case includes live debugging while hiding from other site collections

Requires Site Collection specific app catalog

Use SharePoint Online Management Shell

```
Add-SPOSiteCollectionAppCatalog -Site URL-TO-TARGET-SITE-COLLECTION
```

Demo: Site Collection App
Catalogs and Scoped SPFx

Borrow from the Best

<https://docs.microsoft.com/sharepoint>

<https://github.com/SharePoint/sp-dev-fx-webparts>

<https://github.com/SharePoint/sp-dev-fx-extensions>

<https://github.com/SharePoint/sp-dev-solutions>

SPFx Roadmap CY 2018

Spring 2018

Asset packaging GA

Site Collection App Catalog GA

Native Graph access from SharePoint Framework preview

API and dev support for many OOTB features

Summer / Fall

Native Graph access from SPFx GA

Enterprise solutions for SPFx to Teams

Global deployment for SPFx

Dynamic data in SPFx components

SPFx in SharePoint 2019 - including modern experiences



Thank You

Eric Overfield

@ericroverfield

ericroverfield.com

docs.microsoft.com/sharepoint

github.com/SharePoint/sp-dev-fx-webparts

github.com/SharePoint/sp-dev-fx-extensions

github.com/SharePoint/sp-dev-solutions

