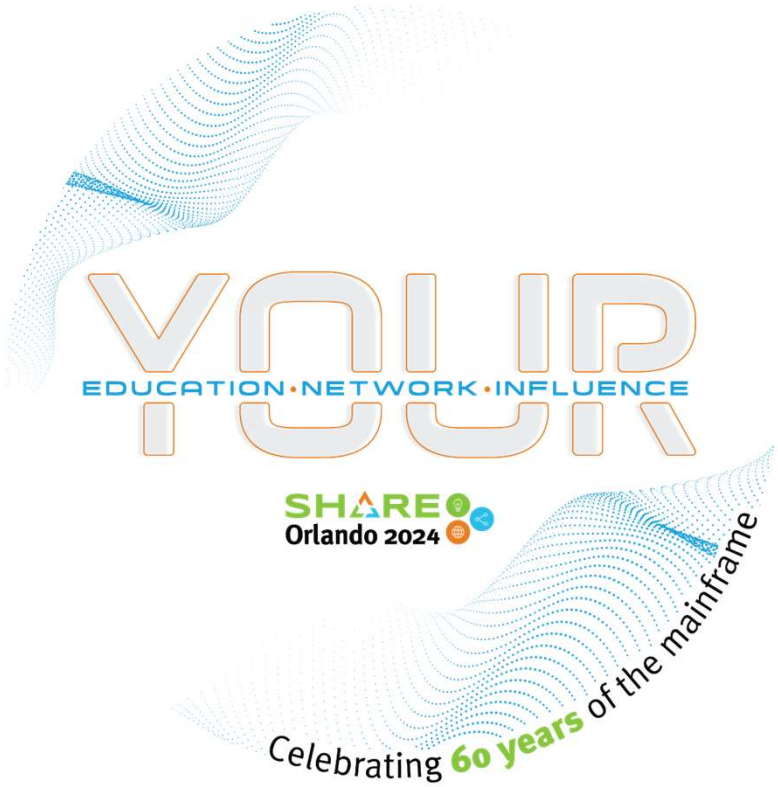


Customizing z/OSMF Using Plugins

Jerry Spencer
DTS Software LLC

Session 73329
Tuesday March 5, 2024
3:30 PM
Room: Salon 21

About the Presenter – Jerry Spencer

SHARE
EDUCATE ‣ NETWORK ‣ INFLUENCE

- Started as an application programmer in 1975
- Became a System Programmer in 1978
- My career pre-dates:
  - The Internet (as we know it)
  - PC's (laptops/desktops/tablets)
  - Cell Phones
  - WiFi
- Passion for sharing knowledge and experience

DTS
SOFTWARE, LLC

AGENDA

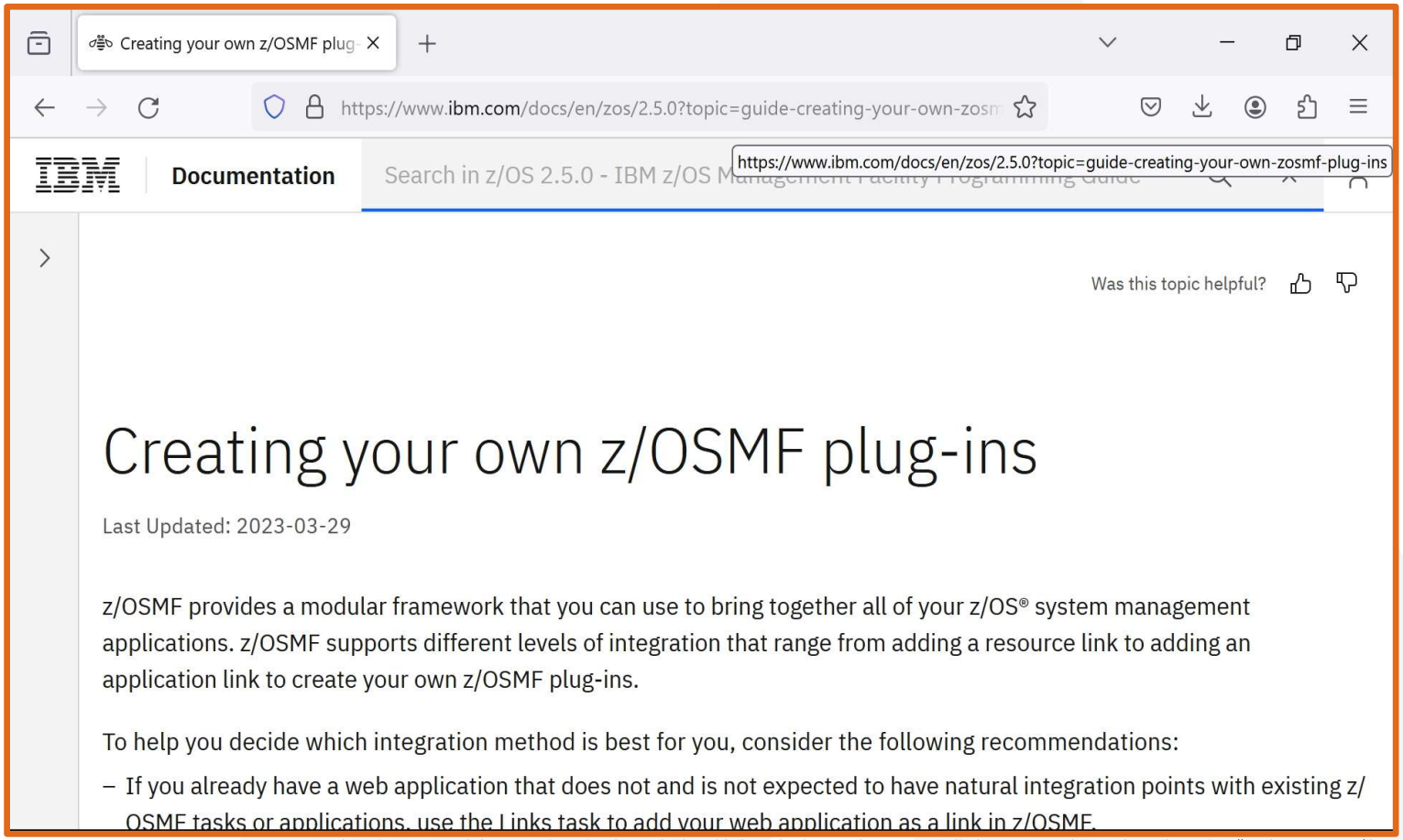
AGENDA

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

- DOWNLOAD THE IBM SAMPLE PLUGIN
- INSTALL THE IBM SAMPLE PLUGIN
- OVERCOME MY PERSONAL ISSUES
- UNDERSTAND THE IBM SAMPLE PLUGIN
- CREATE YOUR OWN PLUGIN
- WRAP UP

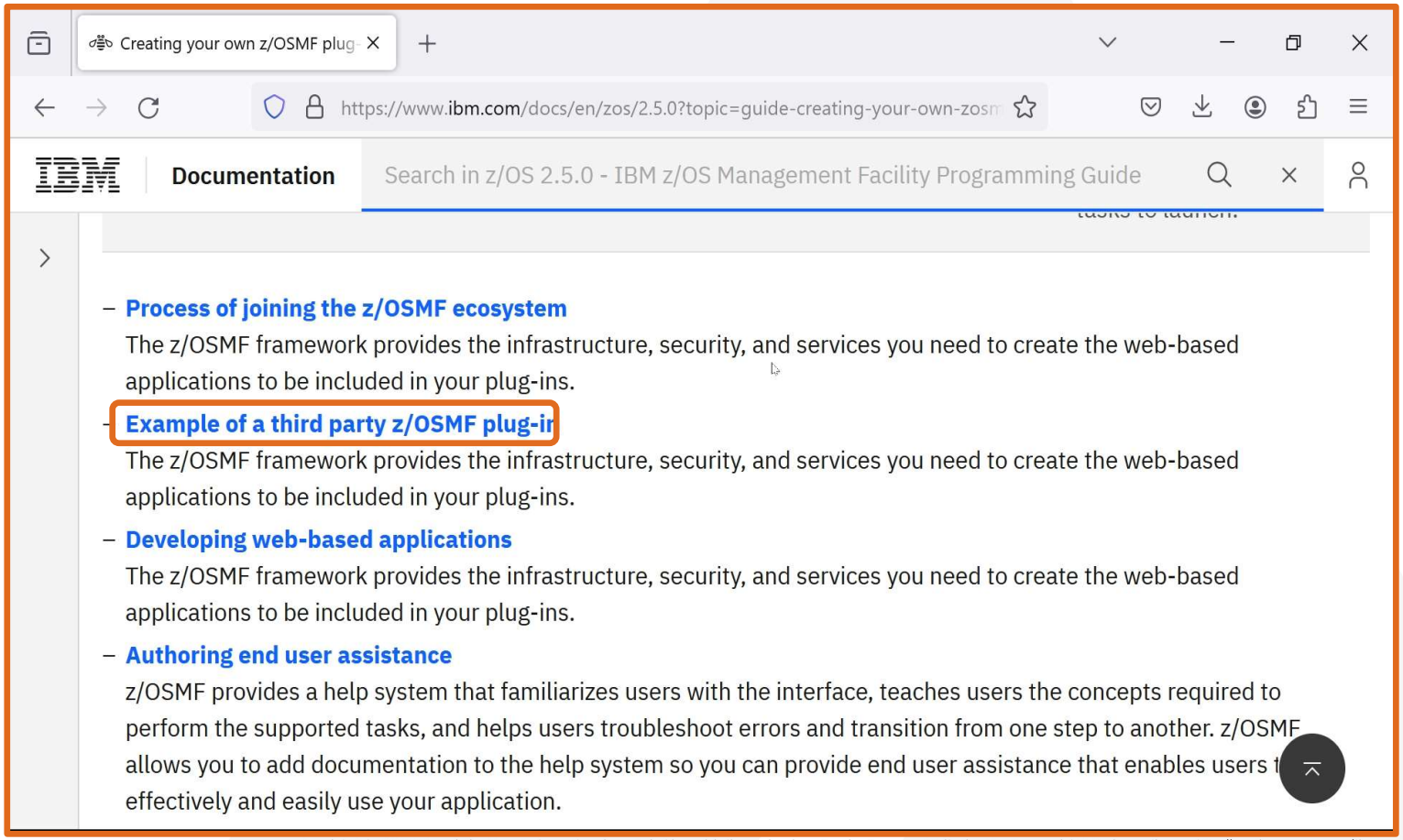
DOWNLOAD THE IBM SAMPLE PLUGIN

DOWNLOAD THE IBM SAMPLE PLUGIN



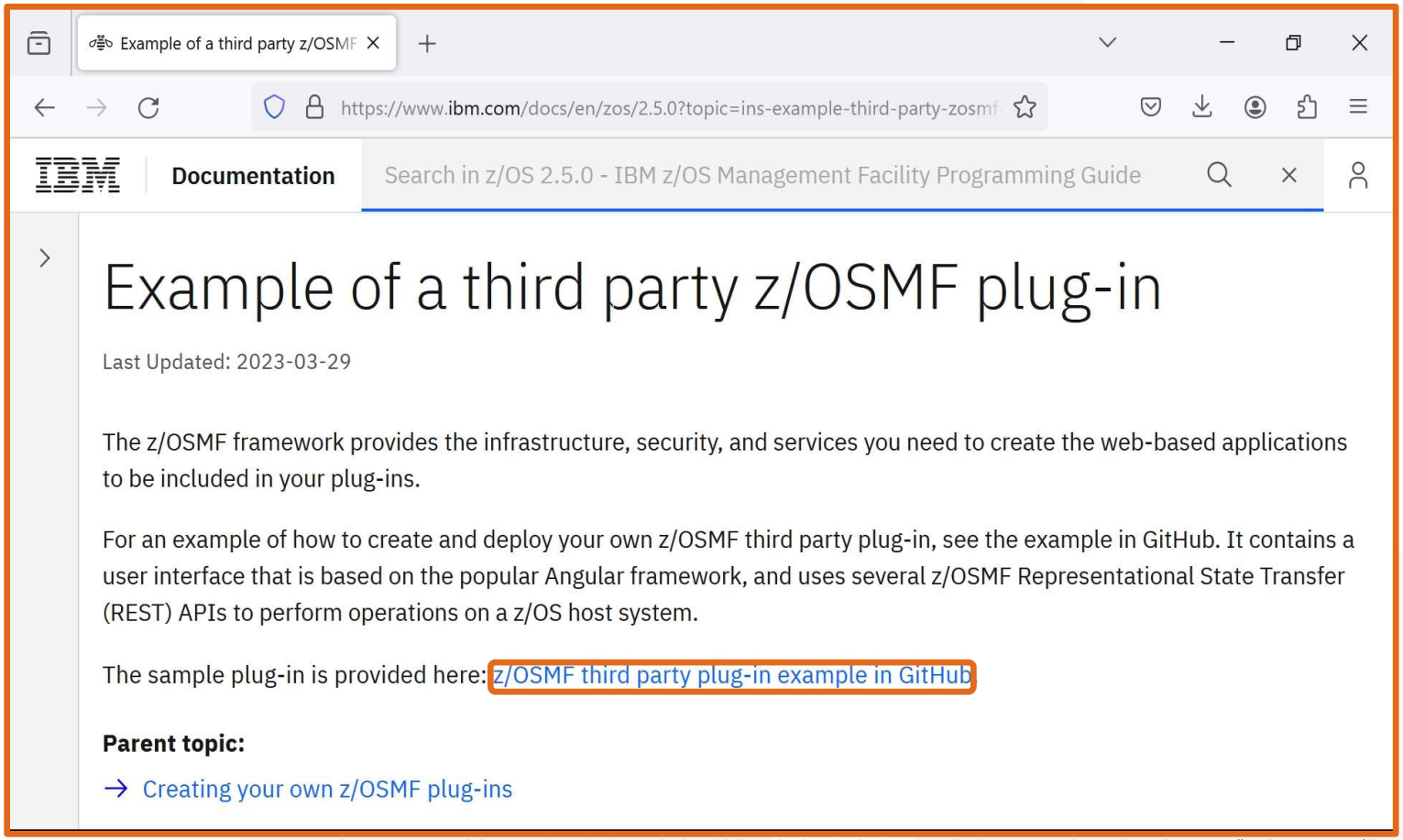
IBM provides a sample plugin to provide some assistance for you to create your own plug-in

DOWNLOAD THE IBM SAMPLE PLUGIN



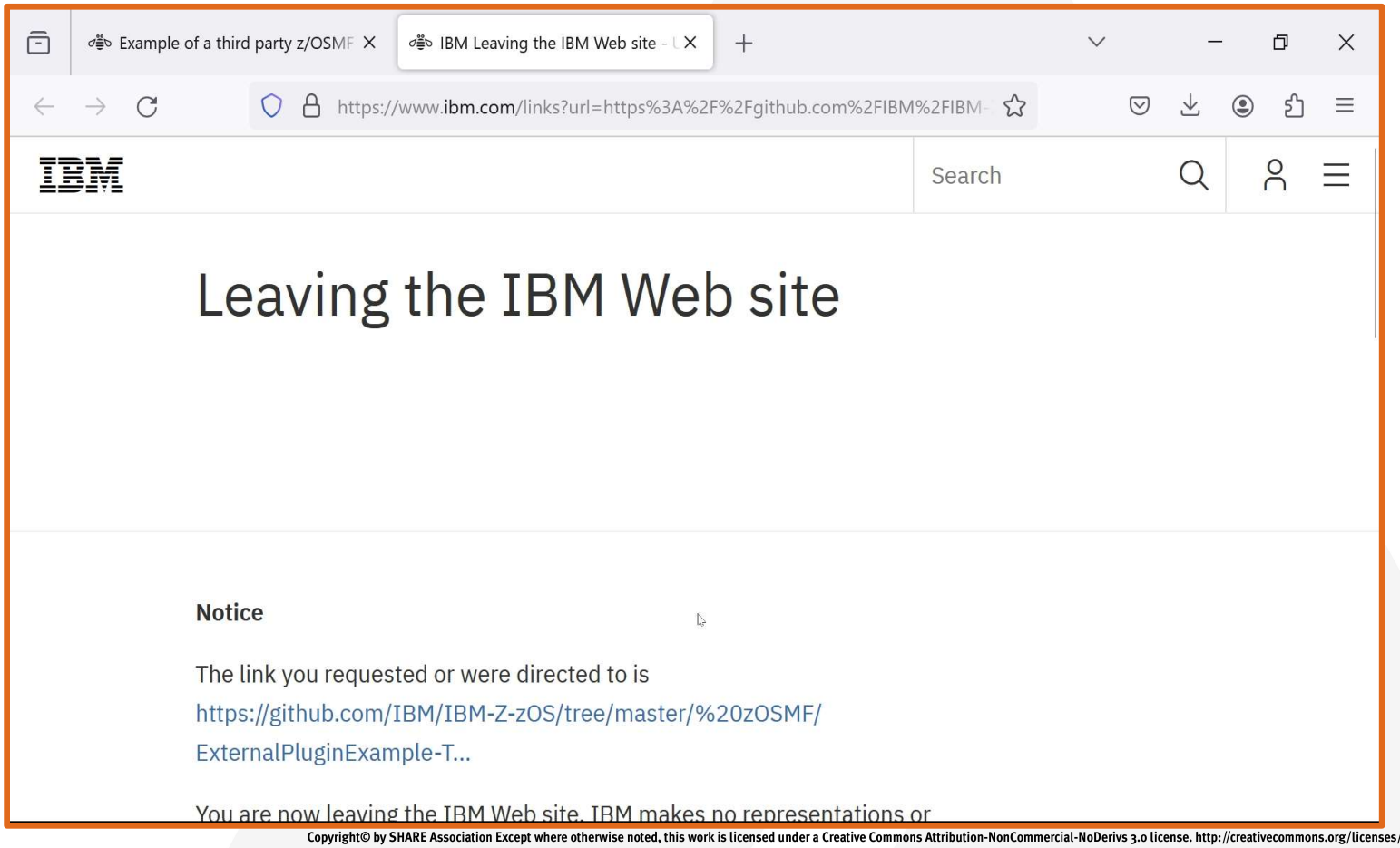
Click on Example of a third party z/OSMF plug-in

DOWNLOAD THE IBM SAMPLE PLUGIN



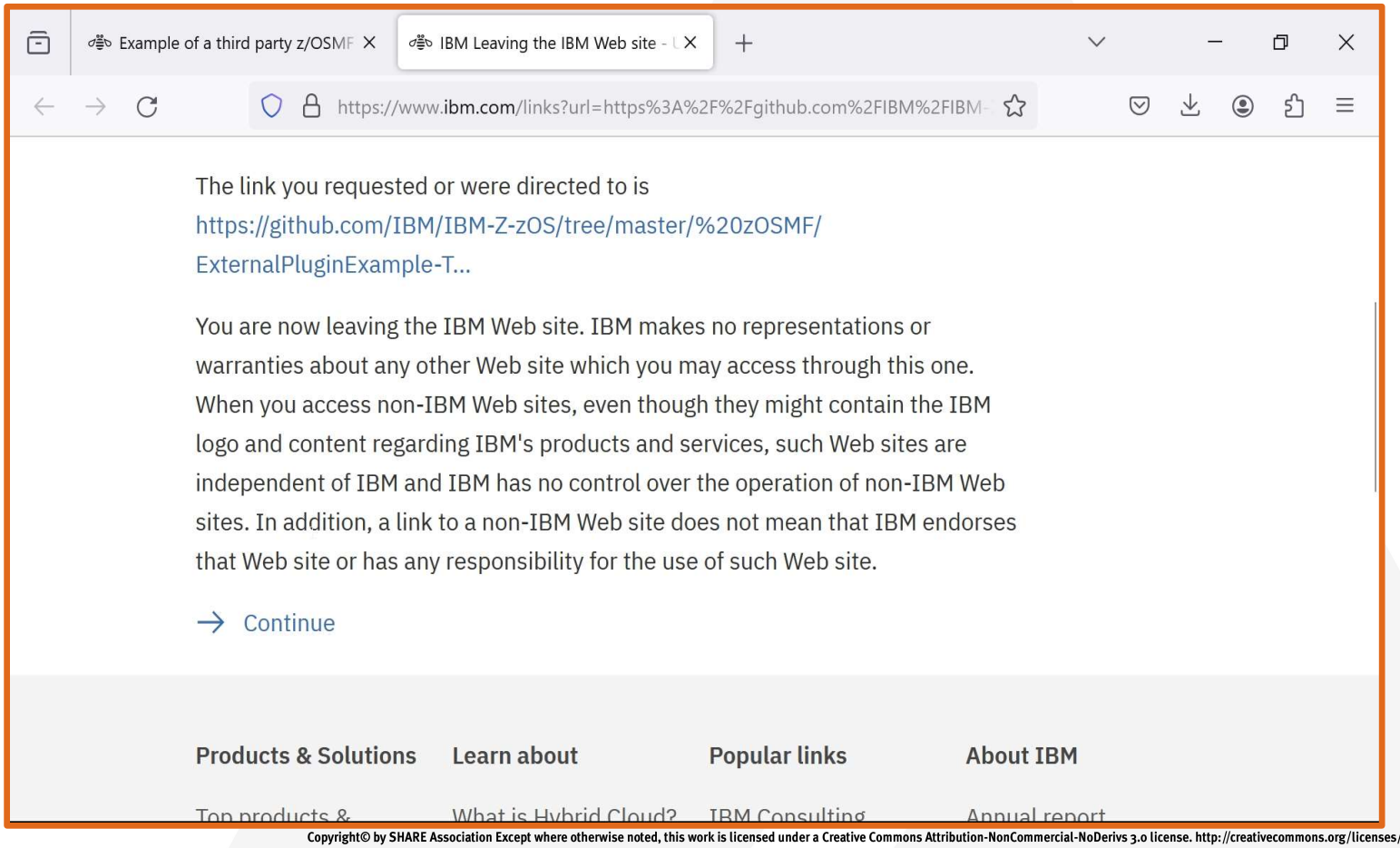
Click on z/OSMF third party plug-in example in GitHub

DOWNLOAD THE IBM SAMPLE PLUGIN



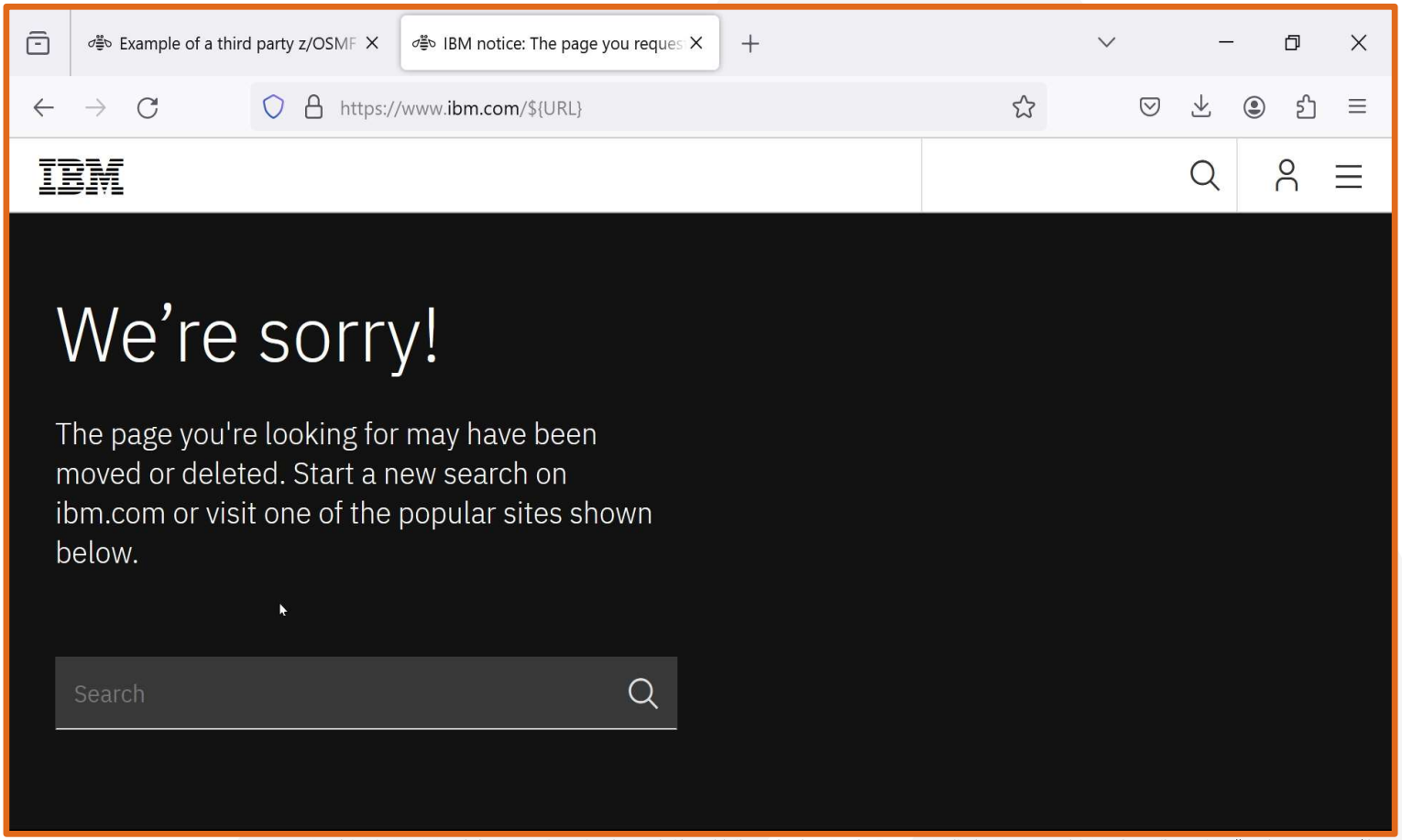
We're on our way to github

DOWNLOAD THE IBM SAMPLE PLUGIN



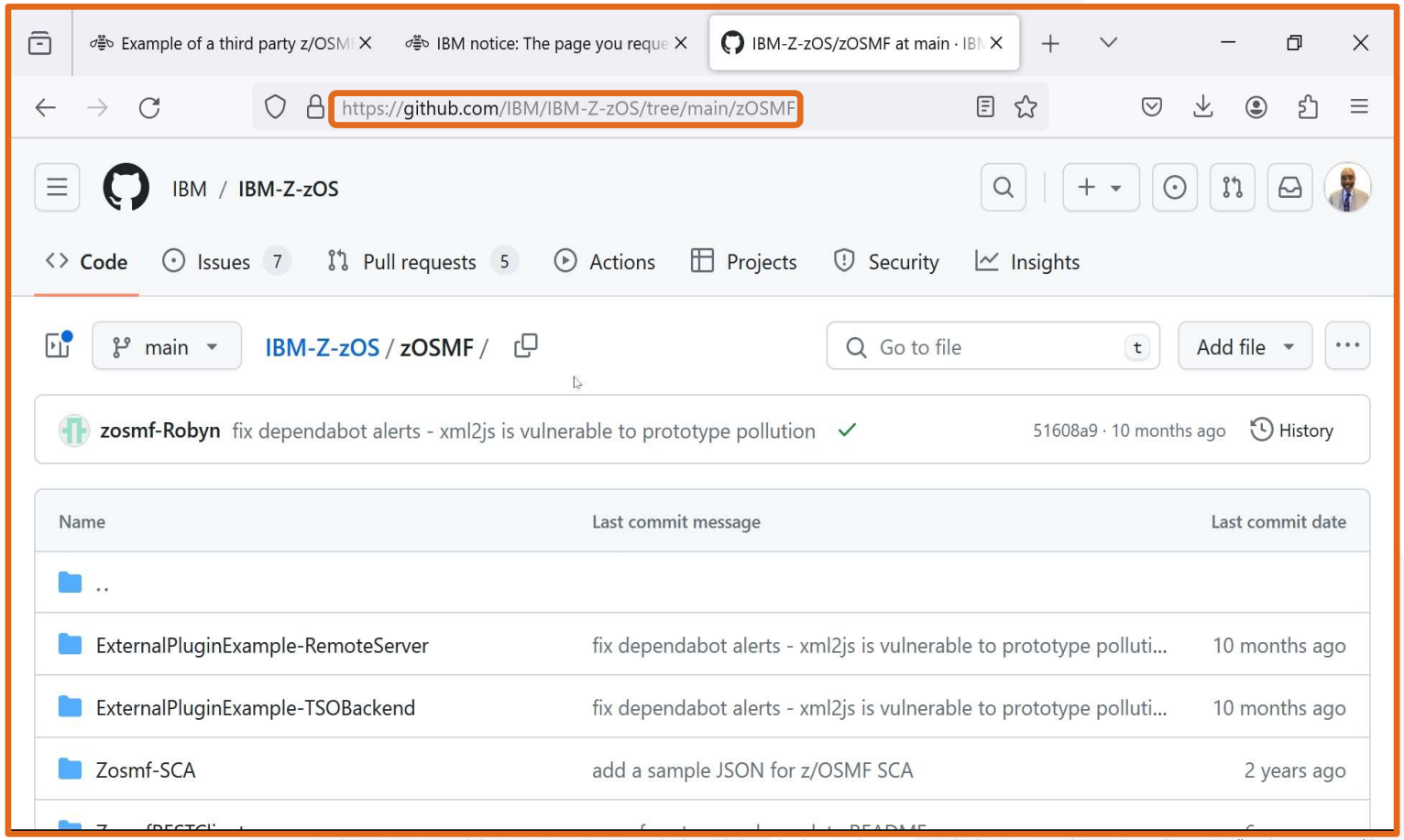
Click on Continue

DOWNLOAD THE IBM SAMPLE PLUGIN



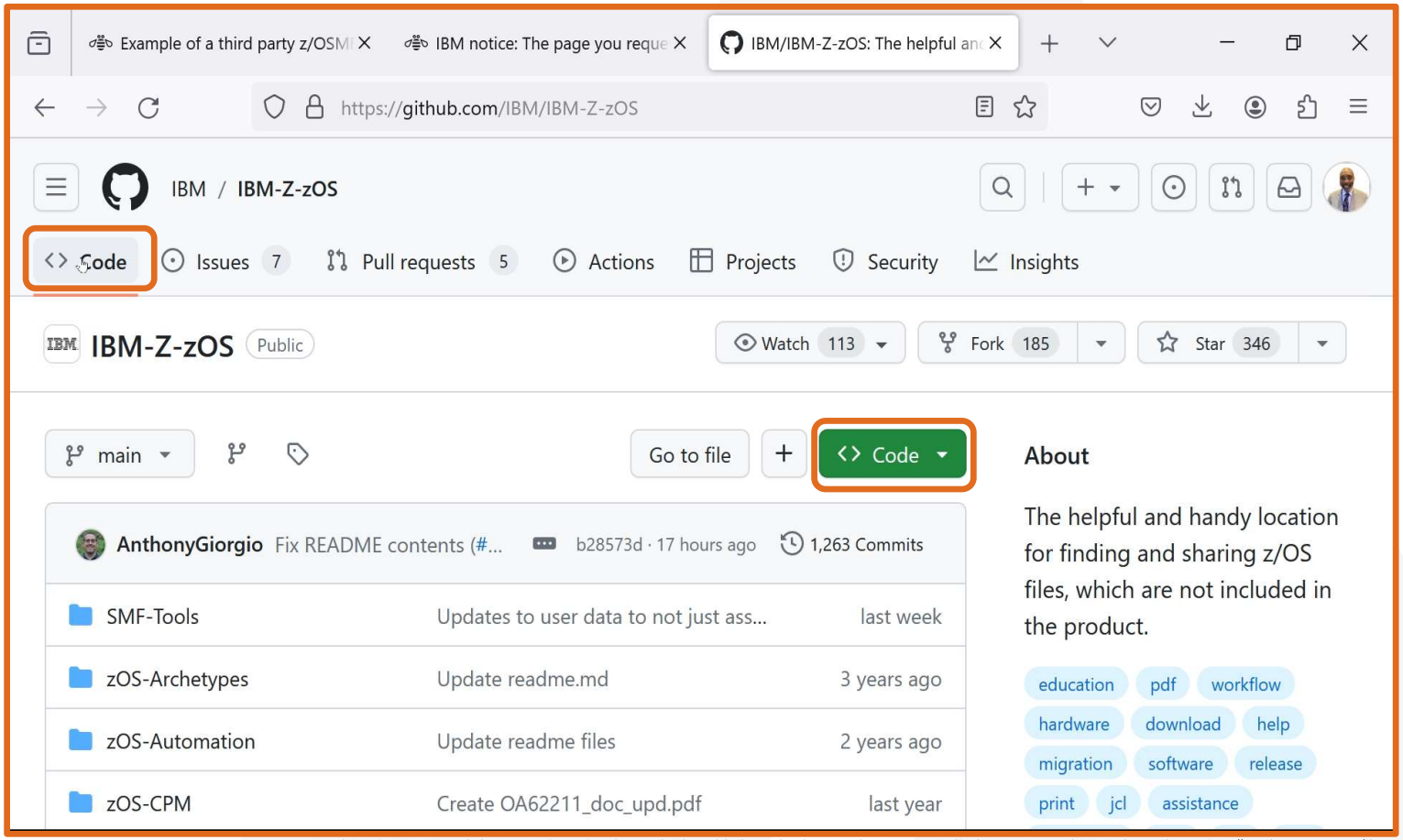
Oops. There's a glitch in the documentation.

DOWNLOAD THE IBM SAMPLE PLUGIN



I went to github on my own and found the zOSMF page

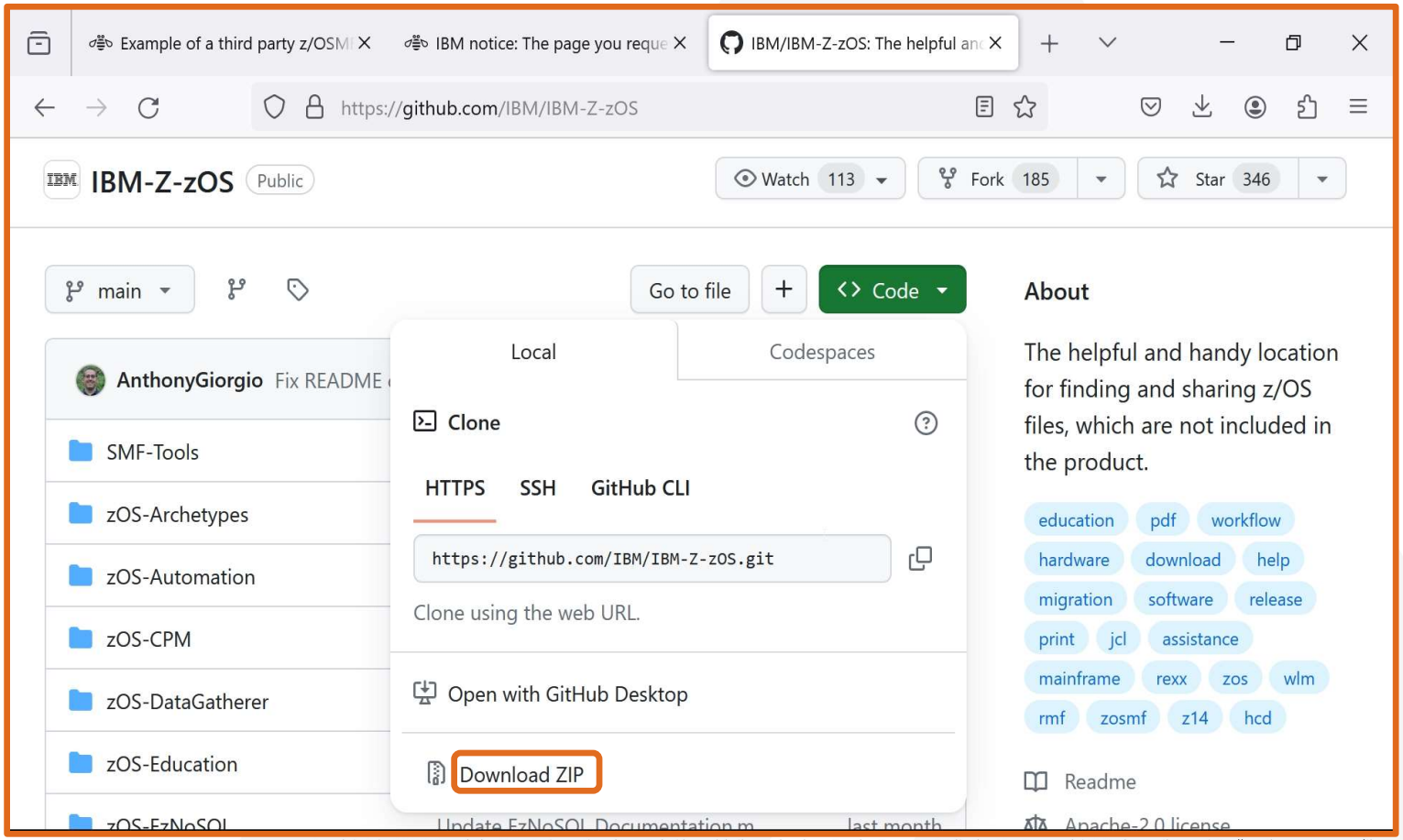
DOWNLOAD THE IBM SAMPLE PLUGIN



I am a mainframe true believer thrown into a non-mainframe world. I floundered around and finally clicked the <>Code button.

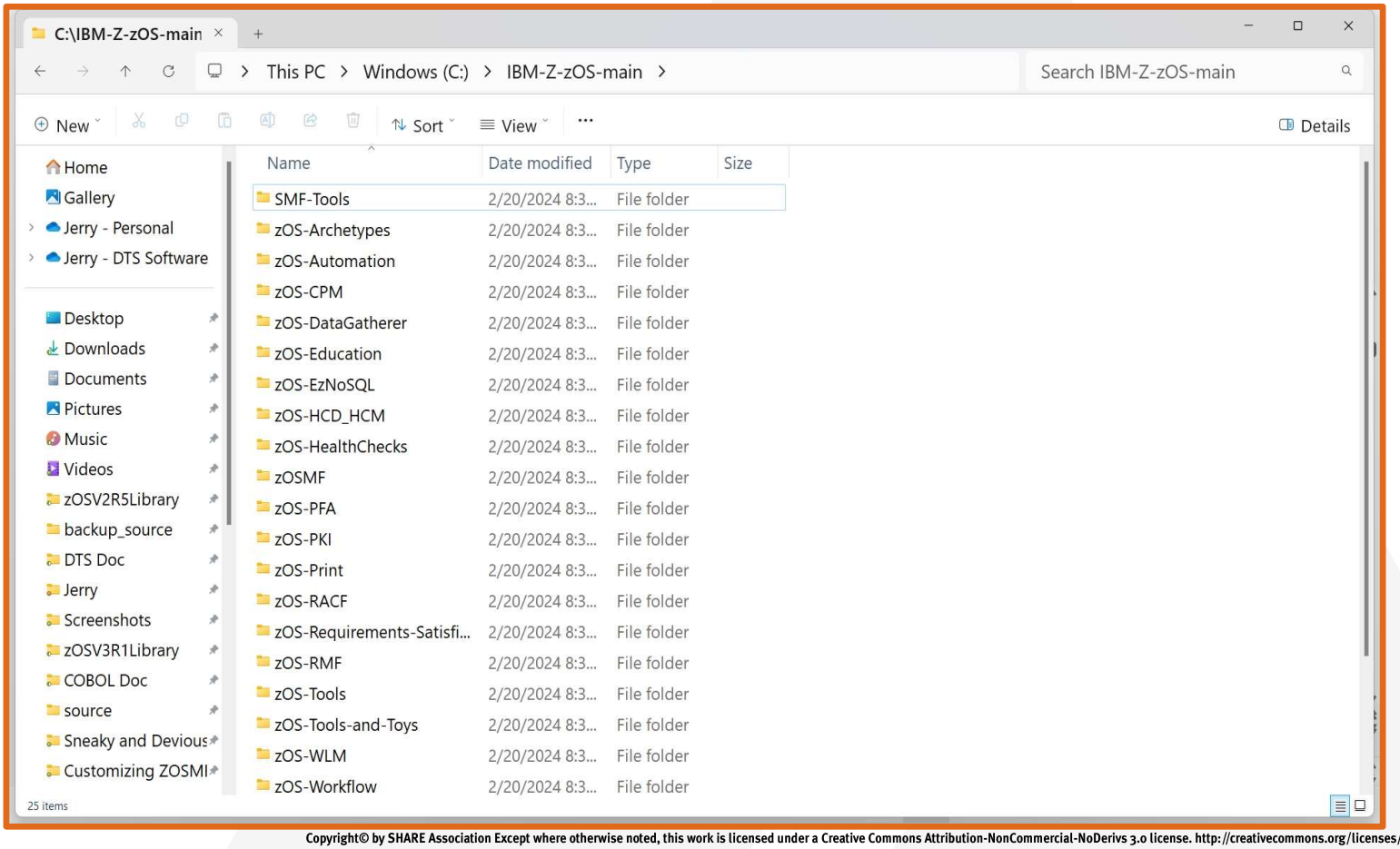
This took me to the base directory and lit up the green <> Code button.

DOWNLOAD THE IBM SAMPLE PLUGIN



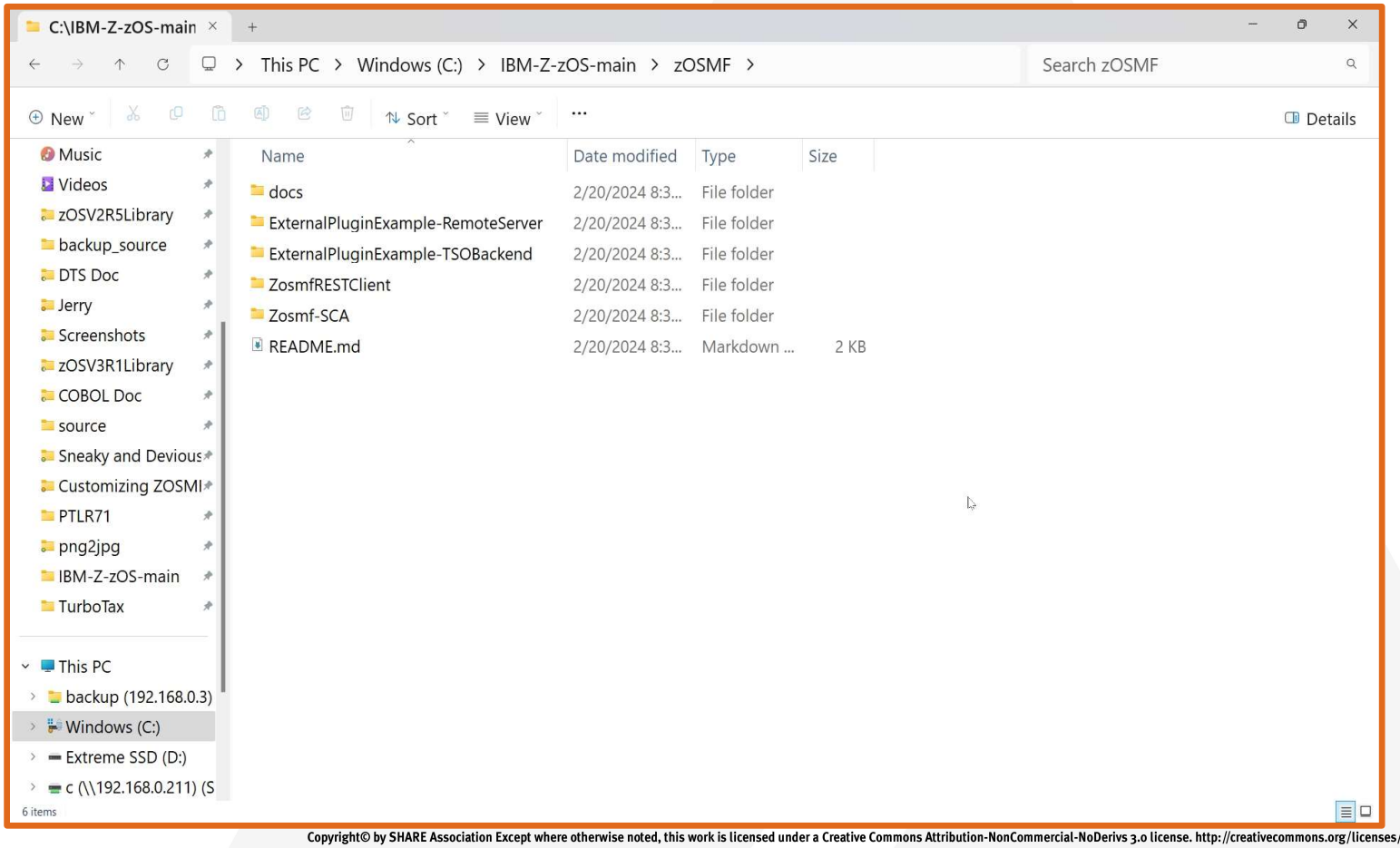
Clicking the green <>Code button allows me to Download ZIP.

DOWNLOAD THE IBM SAMPLE PLUGIN



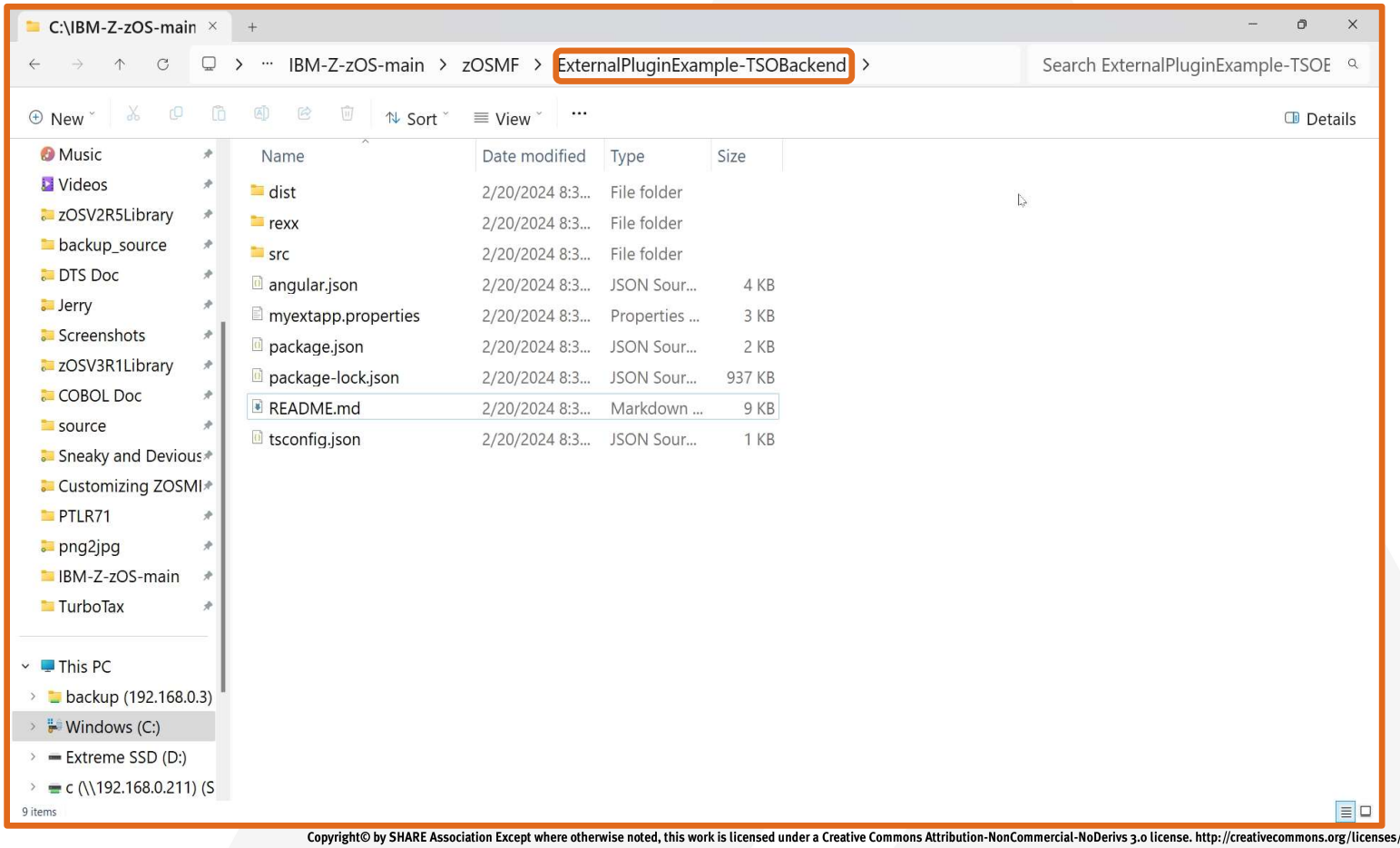
I have extracted all of the files.

DOWNLOAD THE IBM SAMPLE PLUGIN



I navigate to the zOSMF directory.

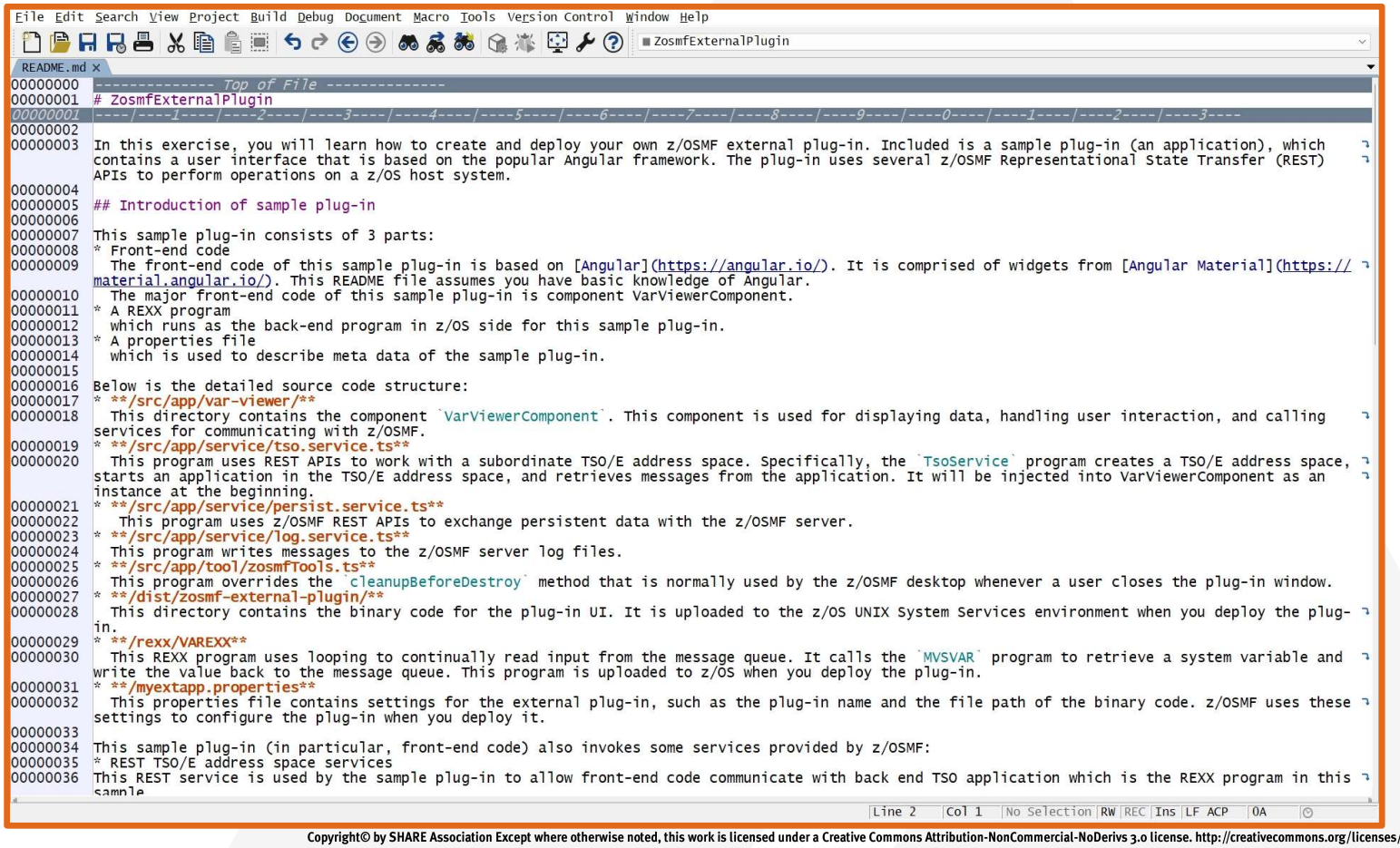
DOWNLOAD THE IBM SAMPLE PLUGIN



And finally I arrive at the ExternalPluginExample-TSOBackend directory

DOWNLOAD THE IBM SAMPLE PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE



The Readme.md file contains the installation instructions.

INSTALL THE IBM SAMPLE PLUGIN

INSTALL THE IBM SAMPLE PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

Step # 1

Deploy the external plug-in into z/OSMF
1. Verify the parameters for starting the TSO/E address space
The sample plug-in uses the following parameters to start a TSO/E address space. Ensure that these settings are suitable for your environment. If necessary, you can modify the parameters at *_line 20_* in `src/app/service/tso.service.ts`. Alternatively, the parameter `proc` can also be specified in the UI by the plug-in user.

```javascript
param = {
    proc: "IZUFPROC", // this parameter can also be specified from the UI
    acct: "IZUACCT",
    chset: 697,
    cpage: 1047,
    rows: 204,
    cols: 160,
    rsize: 50000 }
```

INSTALL THE IBM SAMPLE PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

Step # 2

2. Upload the REXX file to z/OS
Upload `rexx/VAREXX` to z/OS. The name of the PDS that contains the REXX program is hard-coded at *_line 39_* of `src/app/service/tso.service.ts`. If you upload the REXX program to another location, you must change the source code, too. Also, if you want to rename the REXX program, you must change *_line 40_* of `TsoService`.

```javascript
rexxLib = "ZOSMF.EXTERNAL.REXX";
rexx = "VAREXX"; ```

Remember to **catalog** the PDS.
```

INSTALL THE IBM SAMPLE PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

Step # 3

3. Build the UI and upload it to z/OS

This example uploads the UI binary files to the directory `/dist/zosmf-external-plugin/`. However, if you changed the source code, you might need to build the binary files again. To do so, use the following command. The binary files will reside in the another directory `/dist/ExternalPluginExample-TSOBackend/`.

```shell
ng build --base-href='./'
```

In this example, you must upload the binary files from the source code directory `/dist/zosmf-external-plugin/` or `/dist/ExternalPluginExample-TSOBackend`(if you re-build the project) to the z/OS directory `/usr/lpp/myextapp/dist/`.

INSTALL THE IBM SAMPLE PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

Step # 4

4. Prepare the properties file
Upload the properties file `/myextapp.properties` to the directory `/usr/lpp/myextapp/`. Included in this file are the following properties:

```shell
# below properties are for the plug-in
izu.externalapp.local.context.root=myextapp
izu.externalapp.code.root=zosmf-external-plugin
pluginId=MYEXTAPP
# below properties are for the UI task
taskId1=APPTSO
taskSAFResourceName1=ZOSMF.IBM_MYEXTAPP.APPTSO.VARVIEWER
taskNavigationURL1=index.html
```

INSTALL THE IBM SAMPLE PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

Step # 5

5. Authorize users to the task

On a system with RACF as the security manager, you can authorize users by entering the following RACF commands. Note that the RDEFINE command creates a SAF profile with the format `<safPrefix>.<taskSAFResourceName>`. `<safPrefix>` is configured in z/OSMF; by default it is `IZUDFLT`. `<taskSAFResourceName>` is configured in the `myextapp.properties` file in last step of this procedure.

```
RDEFINE ZMFAPLA IZUDFLT.ZOSMF.IBM_MYEXTAPP.APPTSO.VARVIEWER UACC(NONE)
PERMIT IZUDFLT.ZOSMF.IBM_MYEXTAPP.APPTSO.VARVIEWER CLASS(ZMFAPLA) ID(IZUADMIN) ACCESS(CONTROL)
PERMIT IZUDFLT.ZOSMF.IBM_MYEXTAPP.APPTSO.VARVIEWER CLASS(ZMFAPLA) ID(IZUUSER) ACCESS(READ)
SETROPTS RACLIST(ZMFAPLA) REFRESH
```

INSTALL THE IBM SAMPLE PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

Step # 6

6. Import the plug-in into z/OSMF

Complete your work by importing the plug-in into z/OSMF.

Do the following:

1. Log into z/OSMF
2. Access the z/OSMF Import Manager task. If you selected the z/OSMF classic view, click the `Import Manager` task in the `z/OSMF Administration` category. Otherwise, if you selected the z/OSMF desktop view, click the `Import Manager` icon on the desktop.
3. In the `Import Manager` task, select the `Import` tab and specify the full file path and name of the property file that you created: `/usr/lpp/myextapp/myextapp.properties`.
4. Click `Import`. A message is displayed to indicate whether the plug-in was added.

INSTALL THE IBM SAMPLE PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

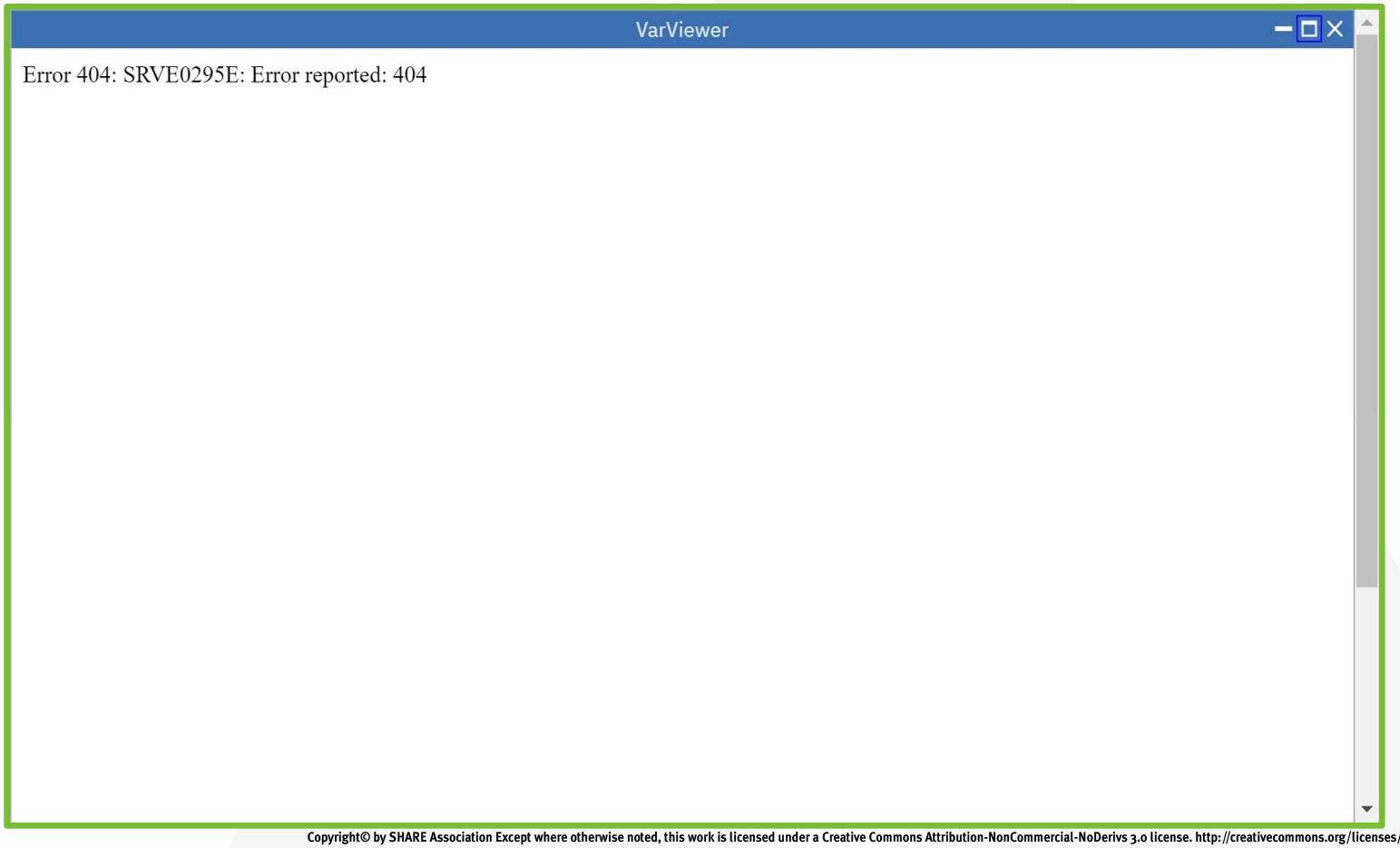
Step # 7

Try the plug-in

Find your plug-in in z/OSMF. In the z/OSMF classic view, expand the `Configuration` category and select `VarViewer`. In the z/OSMF desktop view, select the icon `VarViewer` on the desktop.

OVERCOME MY PERSONAL ISSUES

OVERCOME MY PERSONAL ISSUES



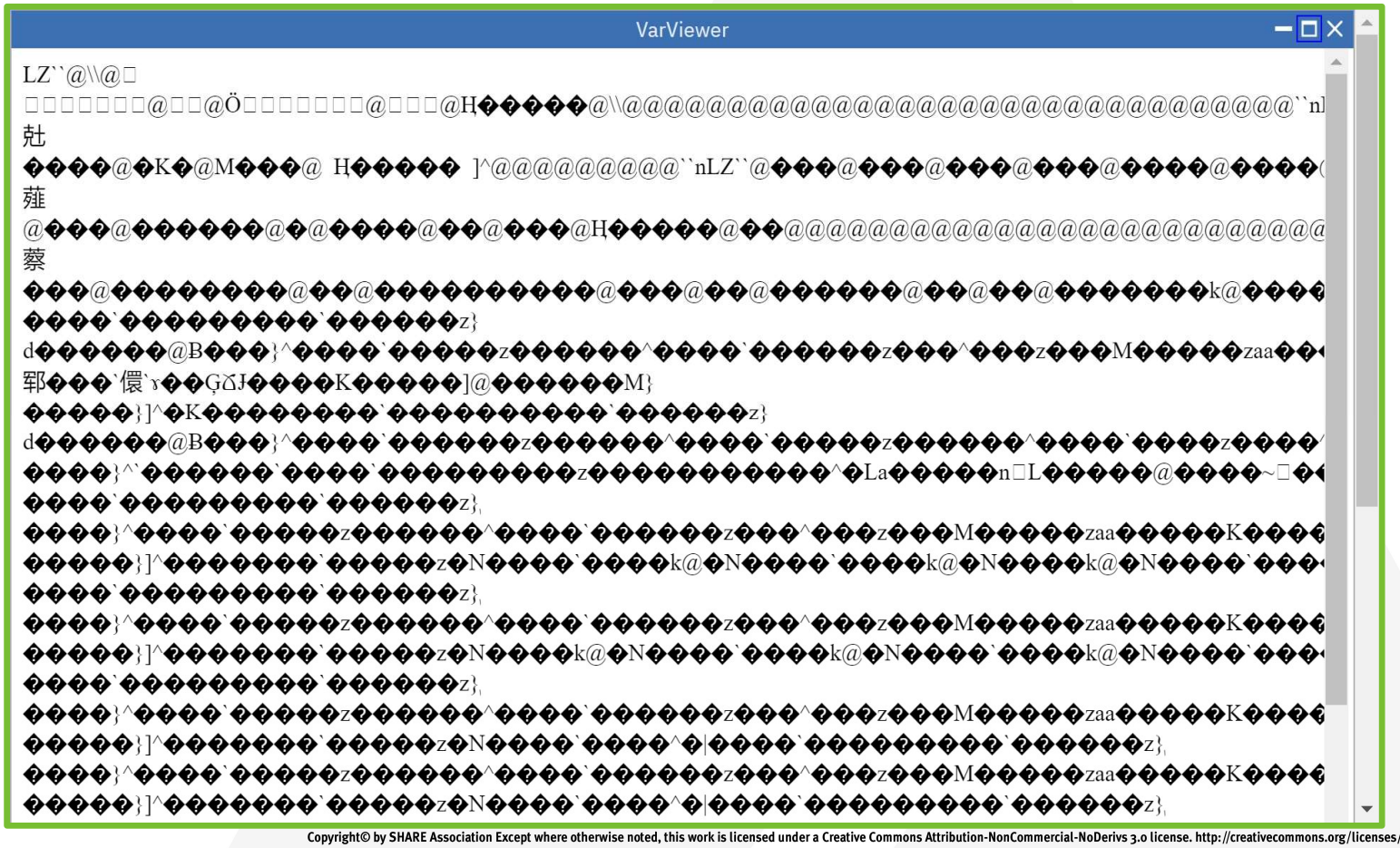
I followed the steps precisely (???!??)

Screenshot from 1st attempt at loading the plug in

After discussing at a staff meeting realized my permissions were wrong

[Hint 1: Talk to your co-workers]

OVERCOME MY PERSONAL ISSUES



Again… I followed the steps precisely (?)

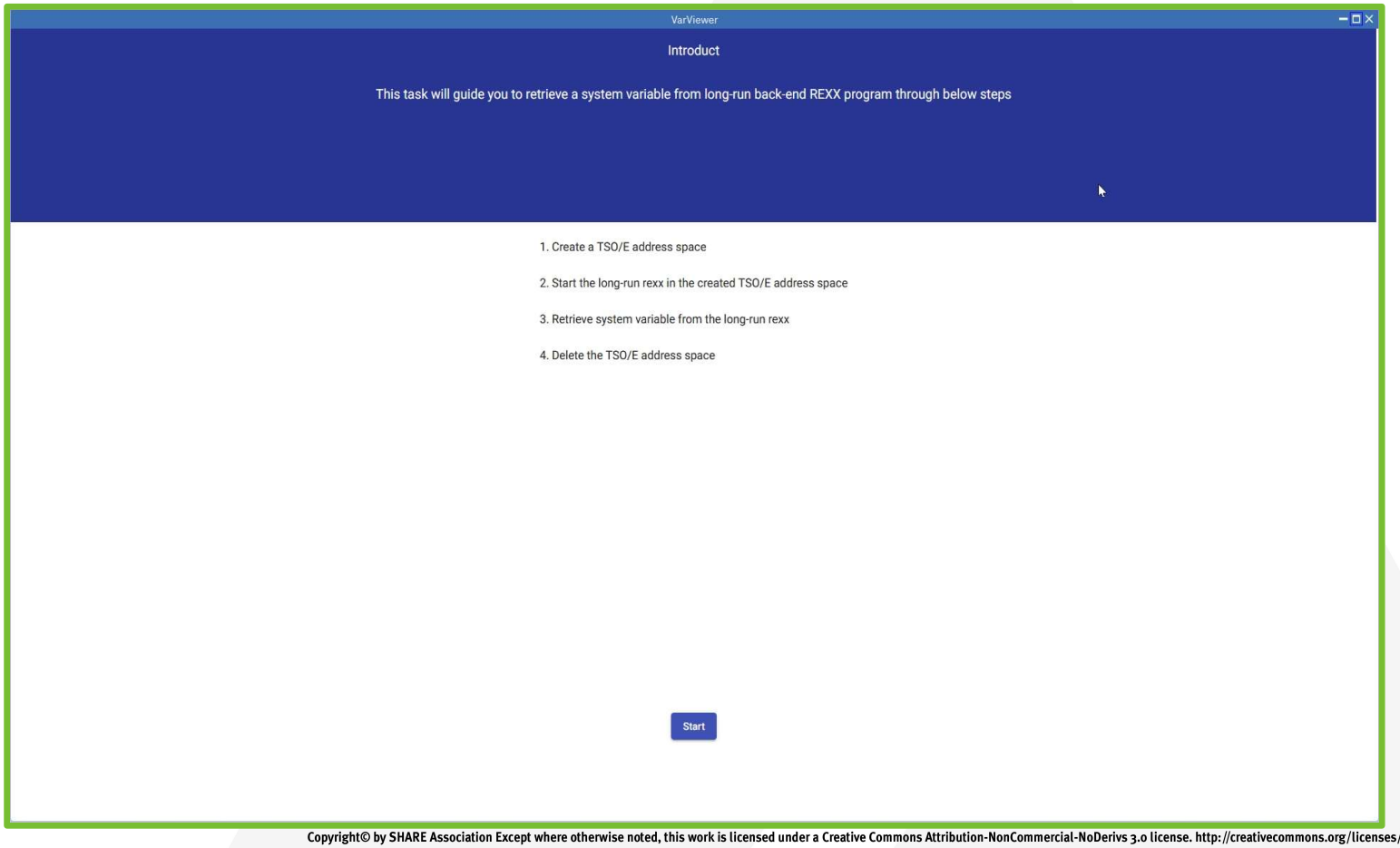
Screenshot from 2nd attempt at loading the plug in (corrected permissions)

It turns out that I outsmarted myself

Binary members on my PC appeared as characters.

[Hint 2 – What you see is not always what you get]

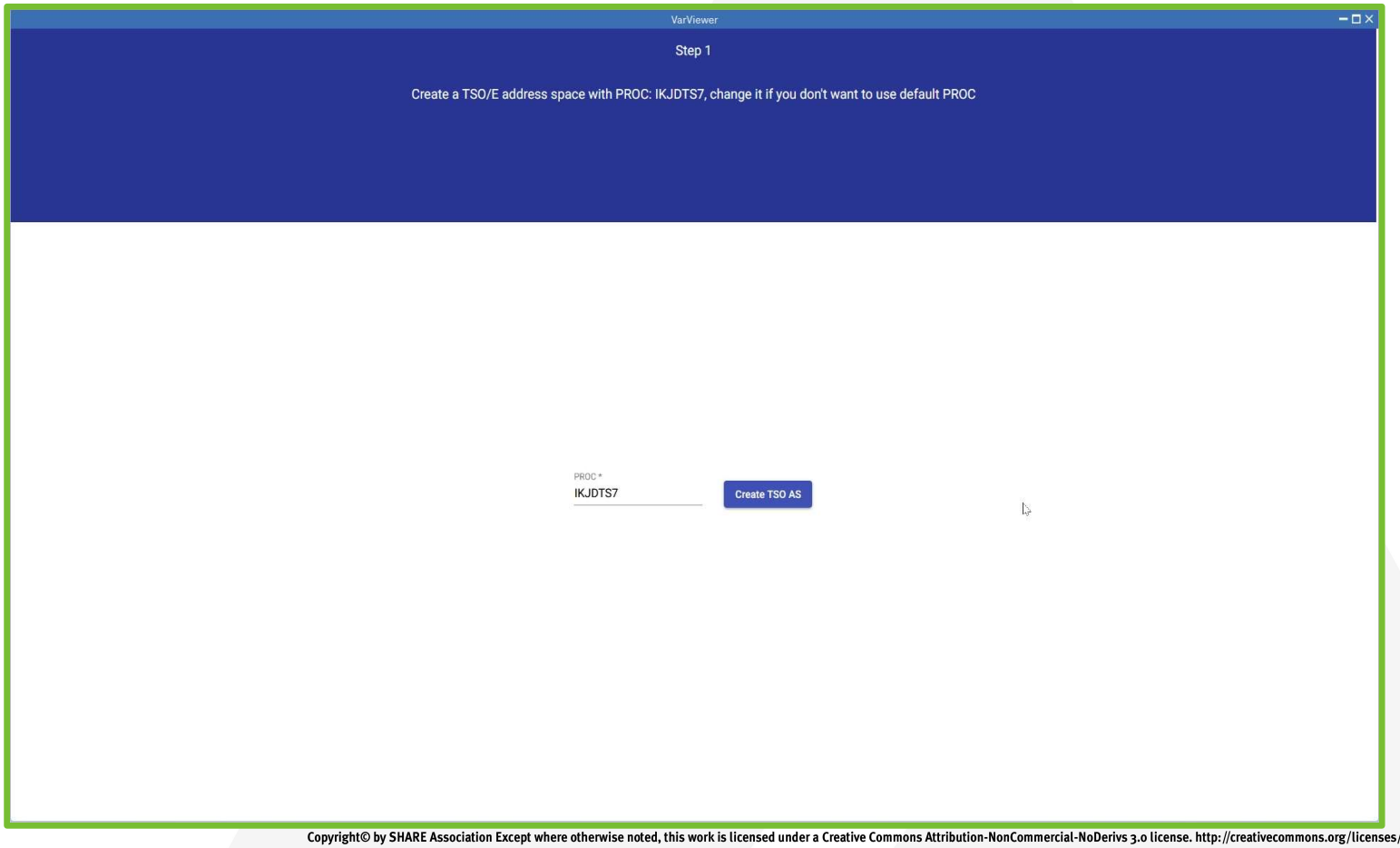
OVERCOME MY PERSONAL ISSUES



Screenshot from 3rd attempt at loading the plug in (corrected permissions & text)

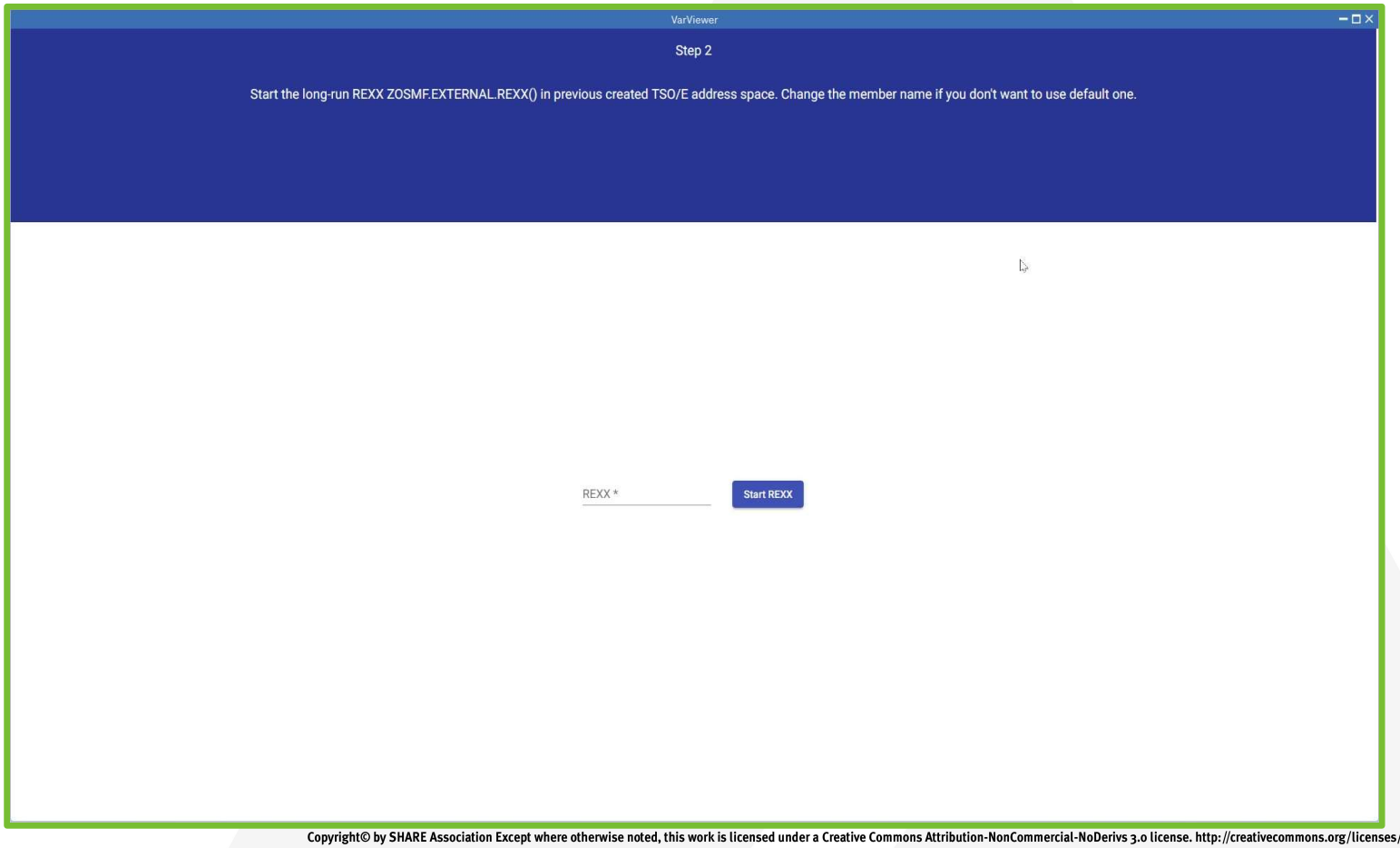
[Hint 3 – 3rd time is the charm… but keep going even if you're still struggling after 3 times.]

OVERCOME MY PERSONAL ISSUES



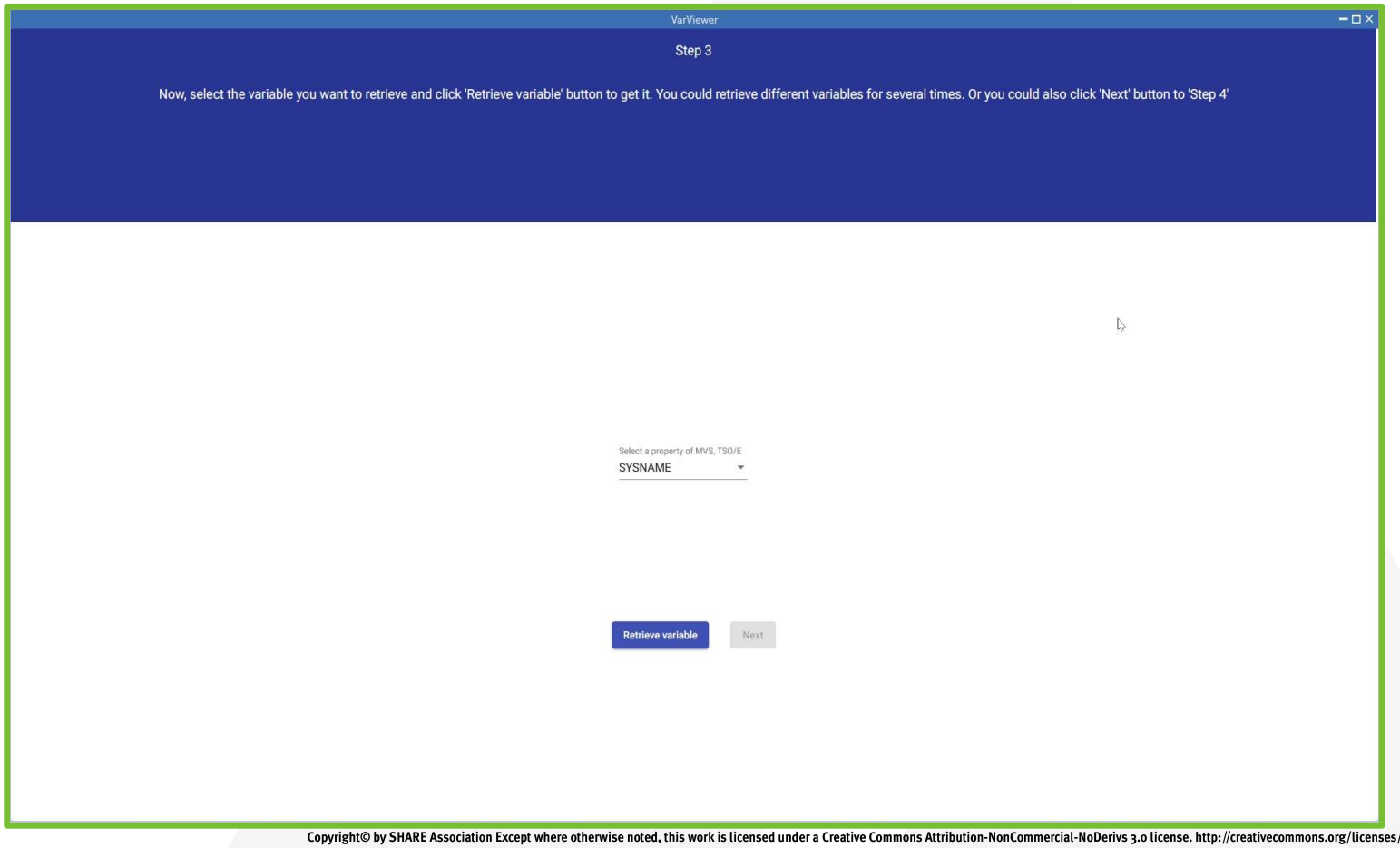
I did not change the default PROC, but the plugin remembered the value I specified the first time through.

OVERCOME MY PERSONAL ISSUES



The next step executes the REXX routine

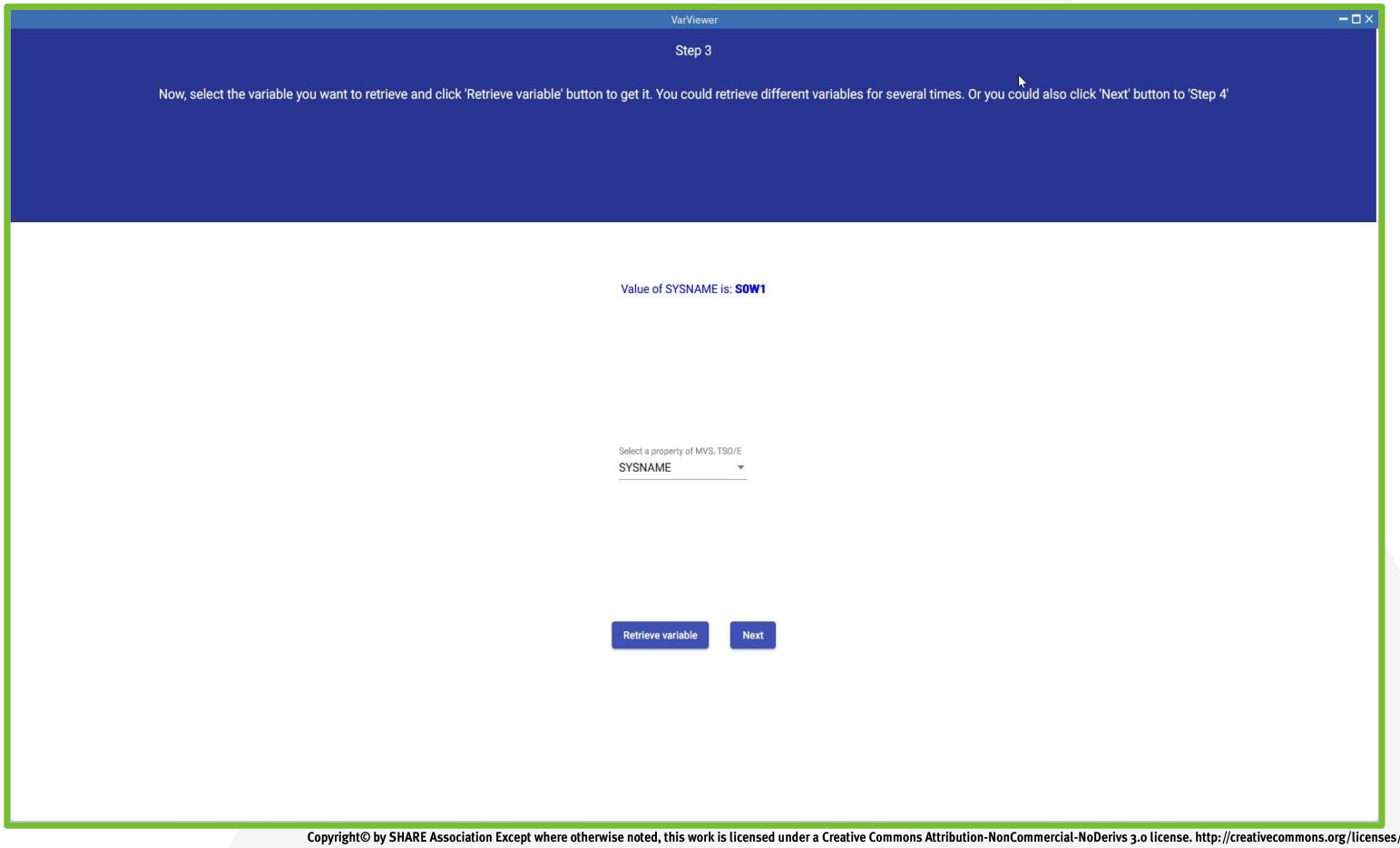
OVERCOME MY PERSONAL ISSUES



I specified I wanted the value for SYSNAME.

Once again, the plugin remembers the value you specified but there's a dropdown that gives you all the options.

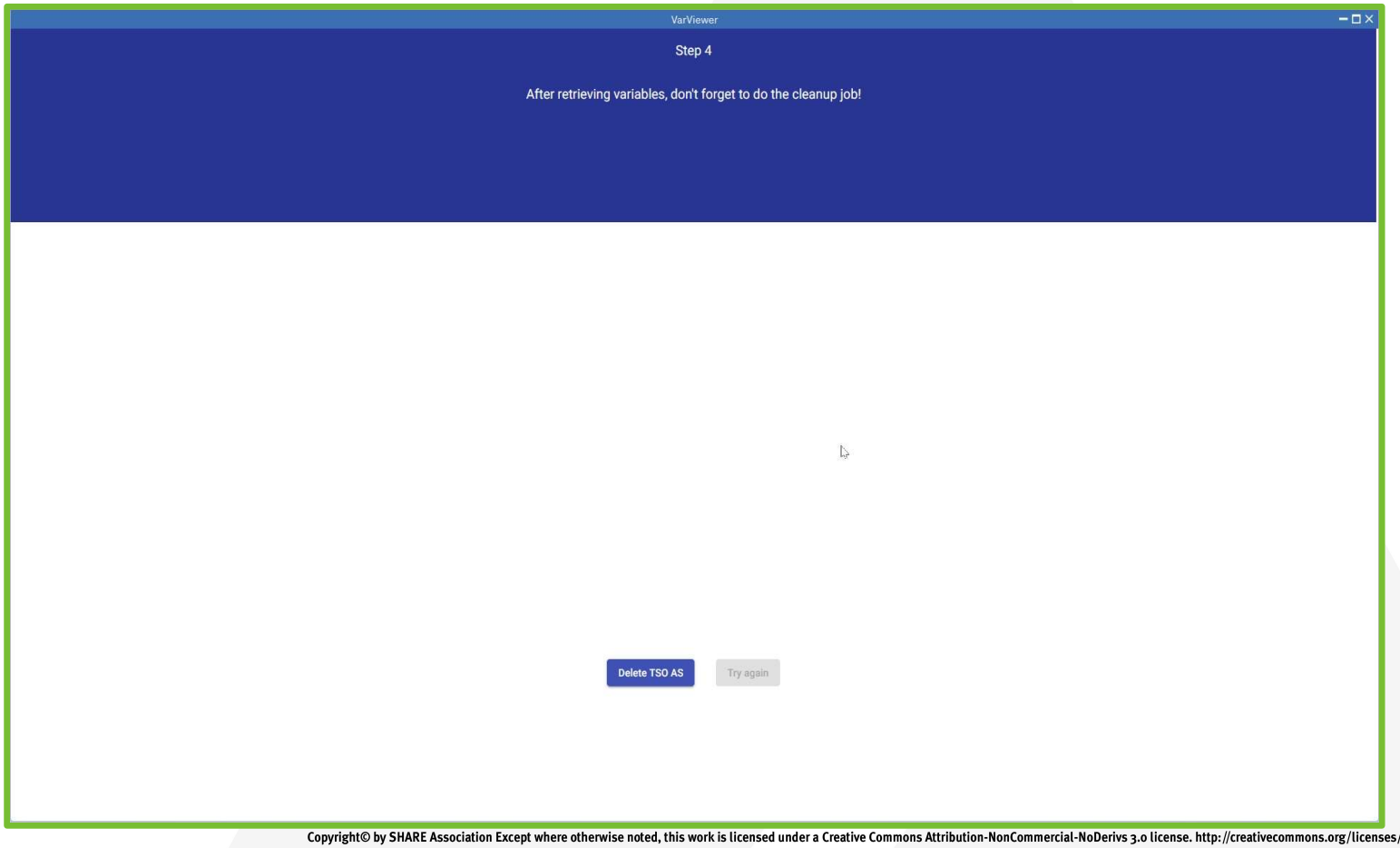
OVERCOME MY PERSONAL ISSUES



The plugin returned the value of SYSNAME (S0W1) and gives me a choice of retrieving a variable or going to the next step

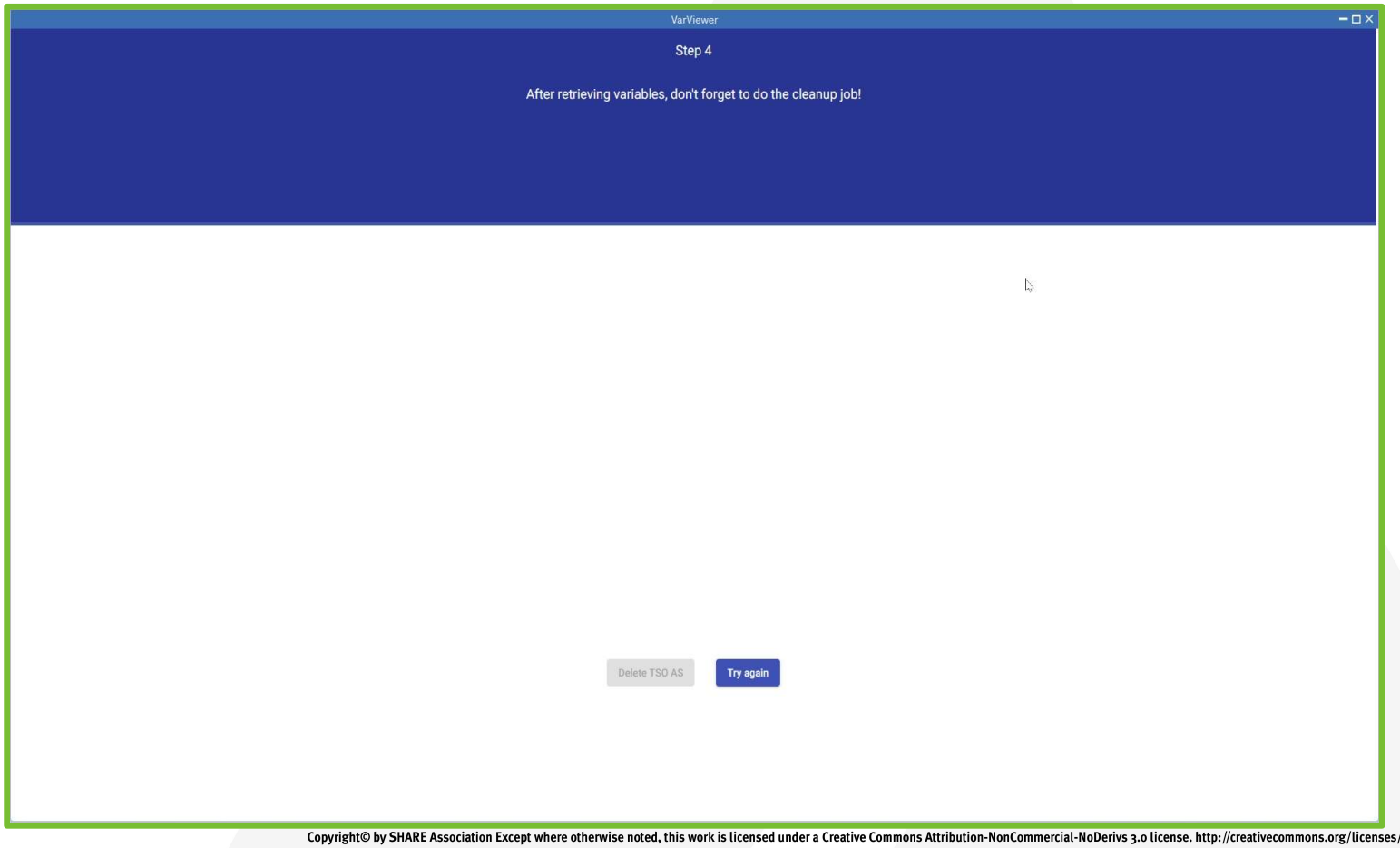
I opted to go to the next step

OVERCOME MY PERSONAL ISSUES



Step 4 allows me to delete the TSO address space

OVERCOME MY PERSONAL ISSUES



Once the address space is deleted, I can go back to step 1 or terminate the plugin.

UNDERSTAND THE IBM SAMPLE PLUGIN

UNDERSTAND THE IBM SAMPLE PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

"Any sufficiently advanced technology is indistinguishable from magic"

Arthur C. Clarke

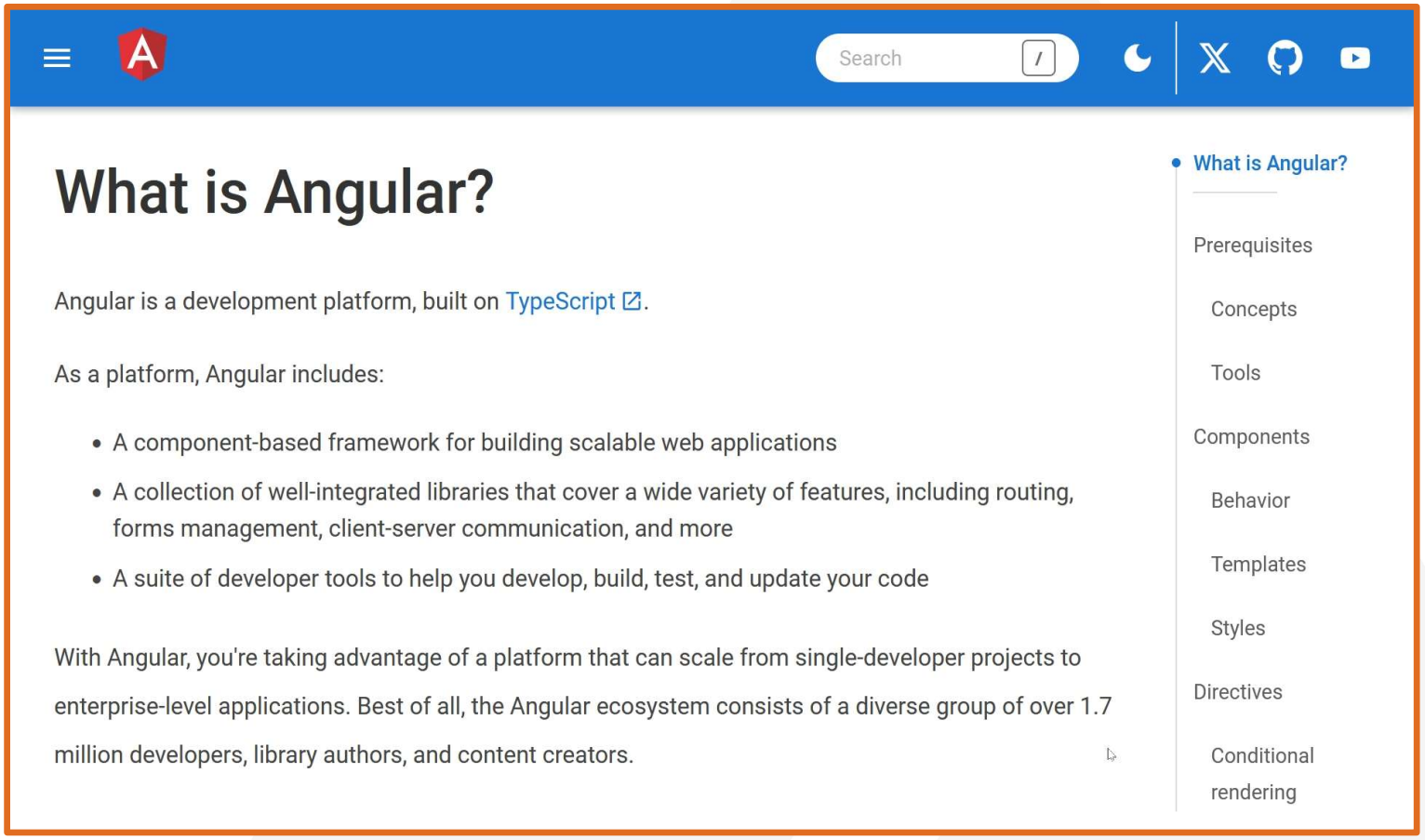
UNDERSTAND THE IBM SAMPLE PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

"... You ain't gonna never
understand it
Till you been all down in it
And find out what it's all about"

Bill Withers

UNDERSTAND THE IBM SAMPLE PLUGIN

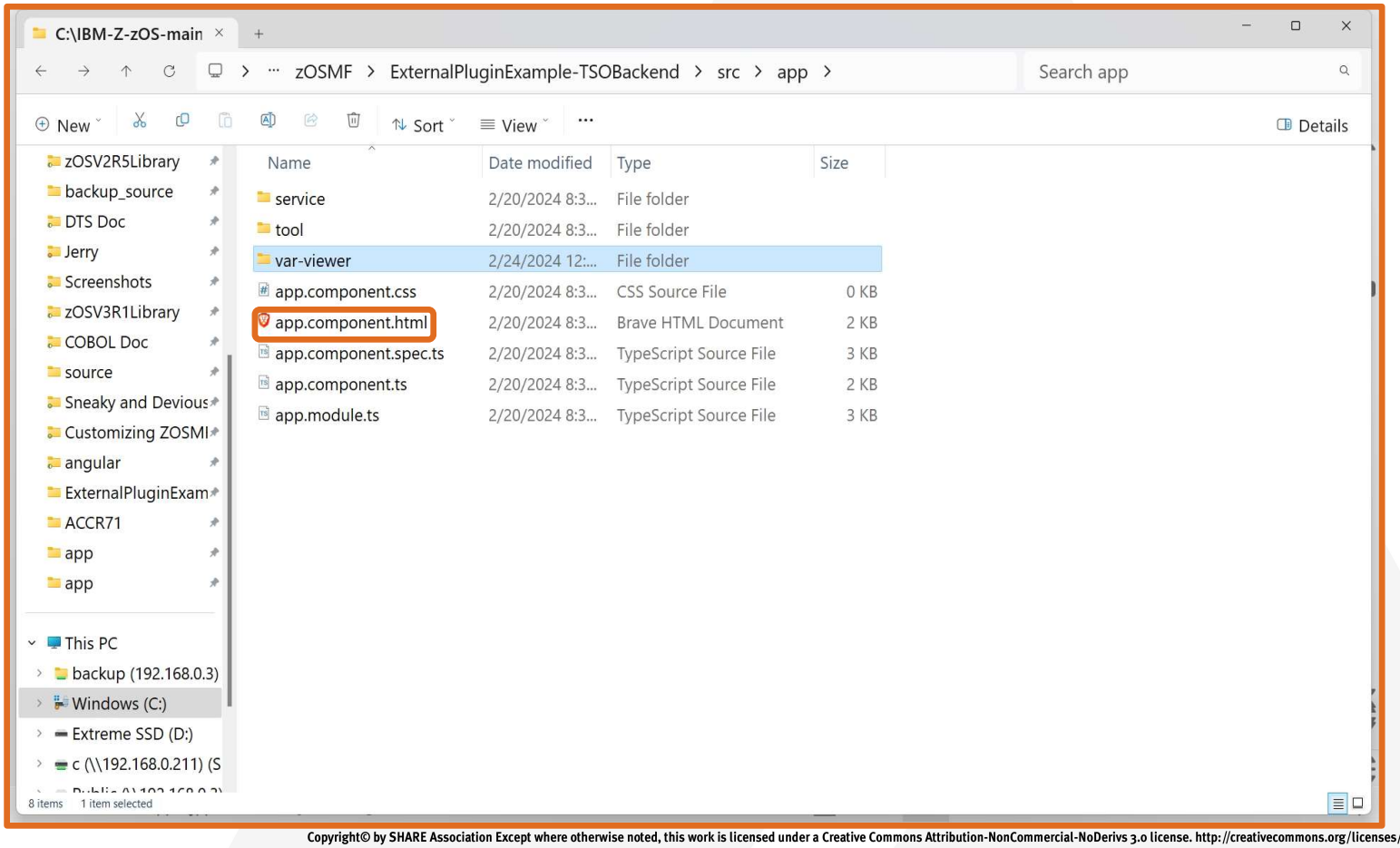


The sample plugin is written using Angular. So, of course, I googled Angular, given I had no idea what it was or how to use it.

I found a course titled “Learn Angular in 5 minutes”

After 10 minutes I considered myself an expert, ready to understand the sample plugin.

UNDERSTAND THE IBM SAMPLE PLUGIN



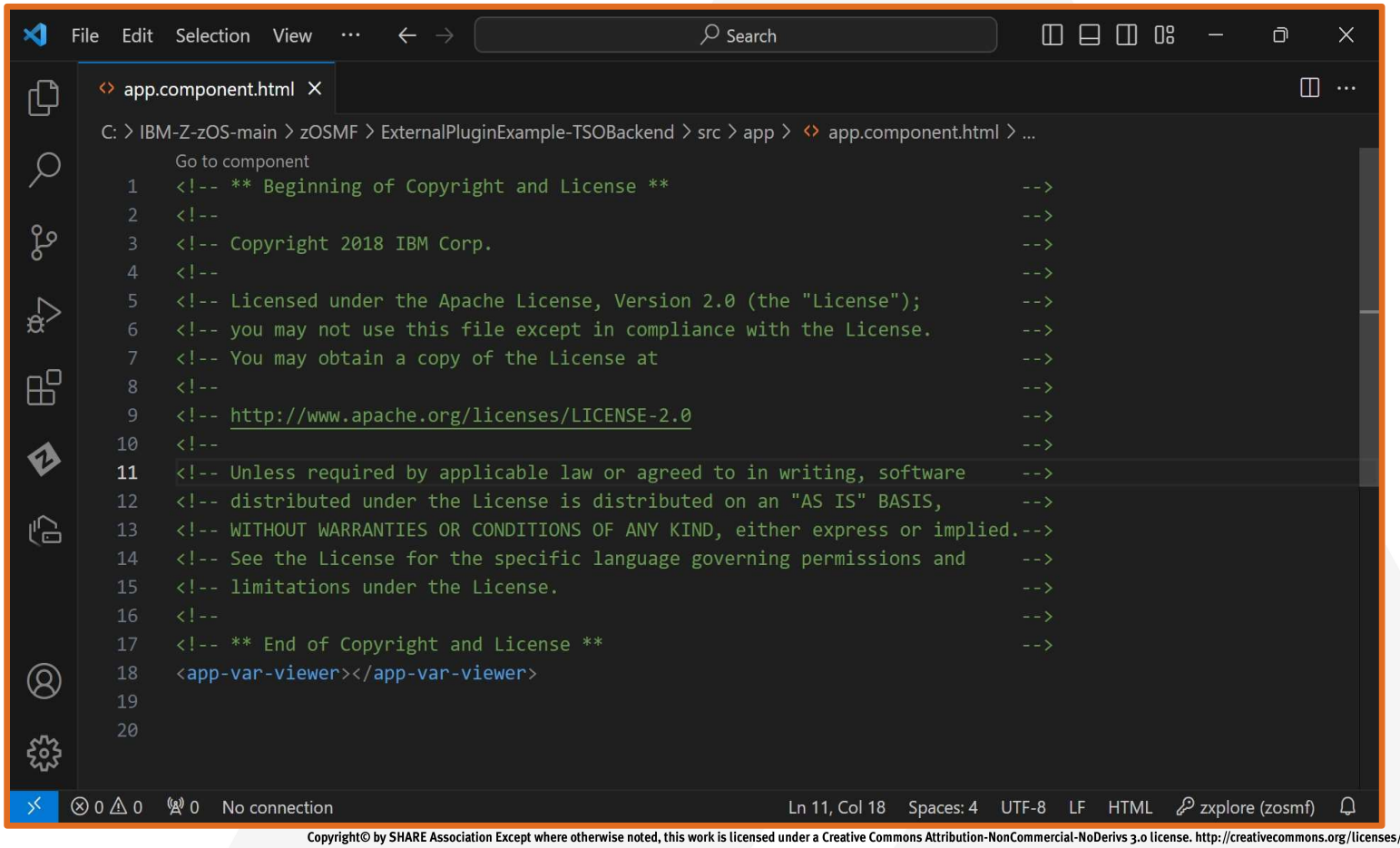
Time for my expert knowledge in action

app.component defines the elements of the web page.

I eagerly opened app.component.html

UNDERSTAND THE IBM SAMPLE PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE



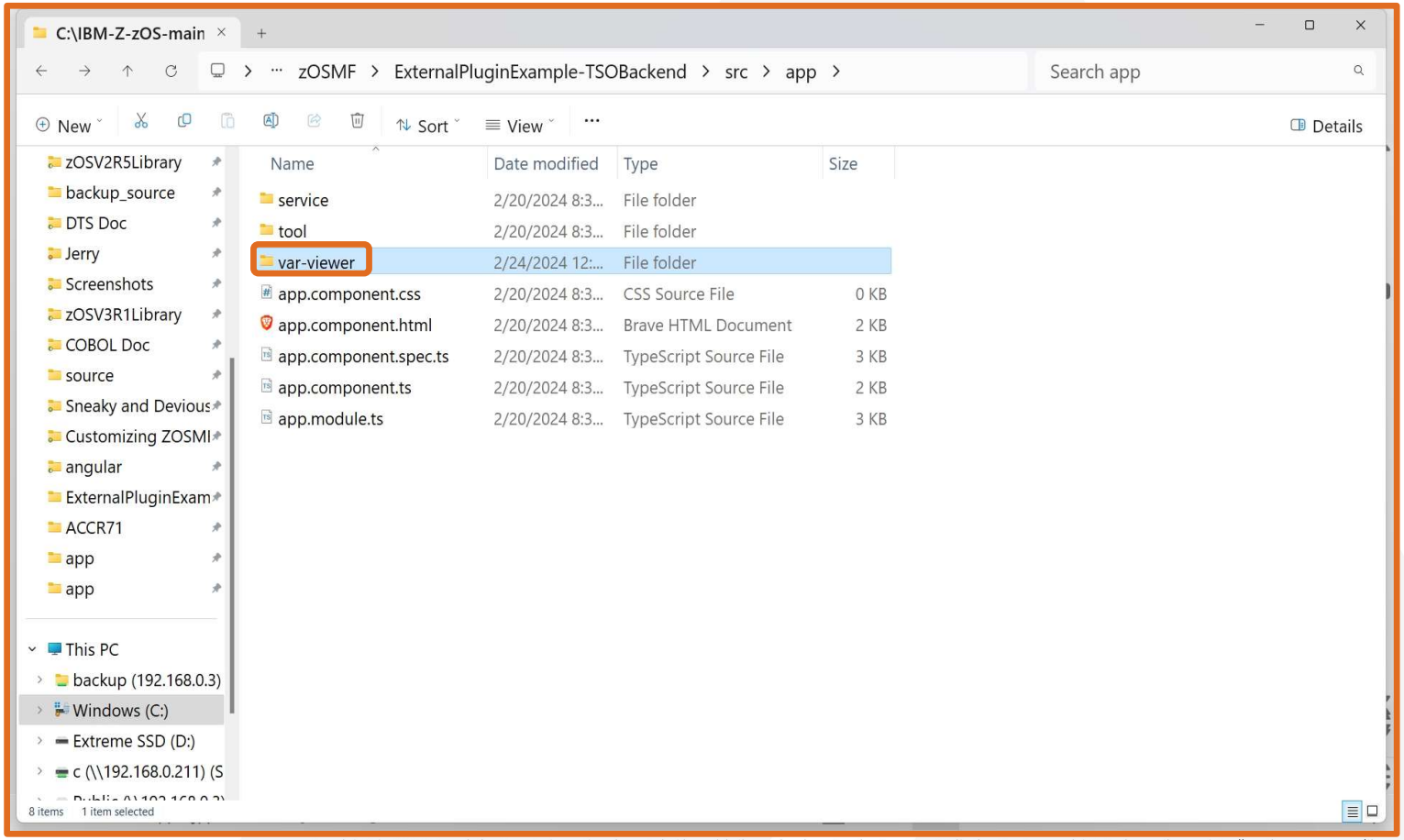
This is not what I was expecting to see.

There's nothing here.

I need more knowledge.

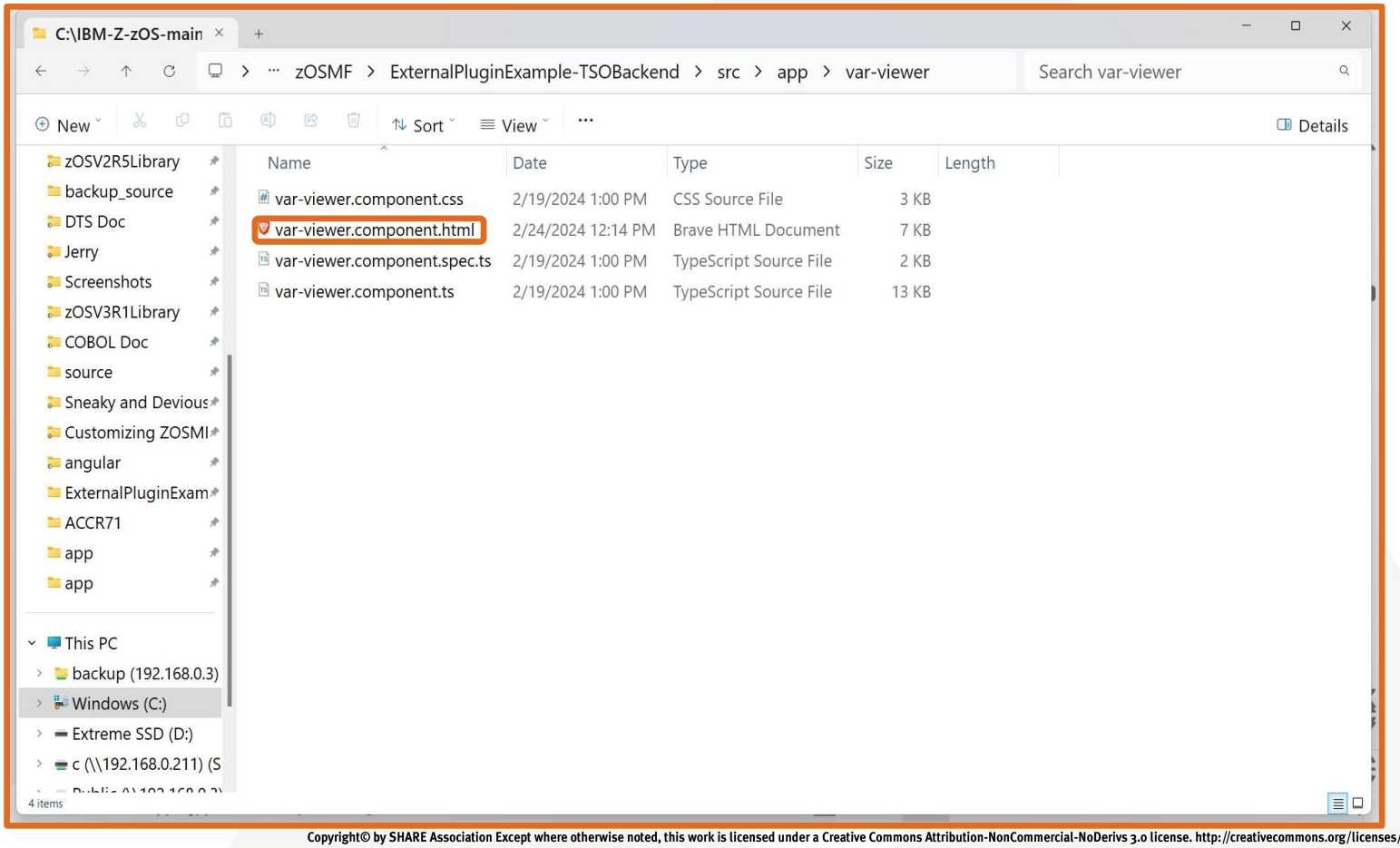
I started a Udemy course on Angular. It explained a lot and I was ready to take the next step.

UNDERSTAND THE IBM SAMPLE PLUGIN



Now I understand. The app is called var-viewer. That's the folder where all of the elements reside

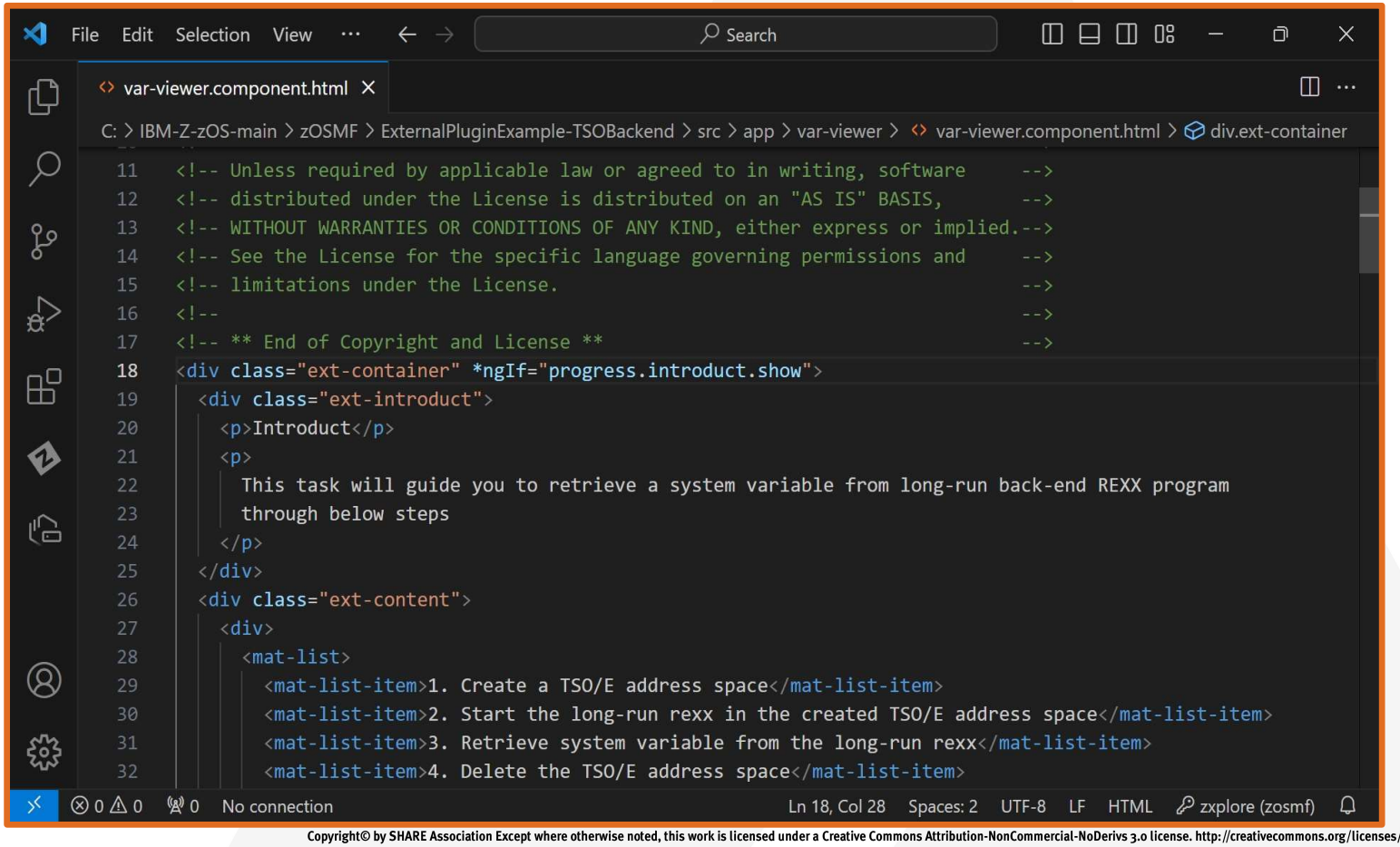
UNDERSTAND THE IBM SAMPLE PLUGIN



This is the html that defines the elements that appear on the screen.

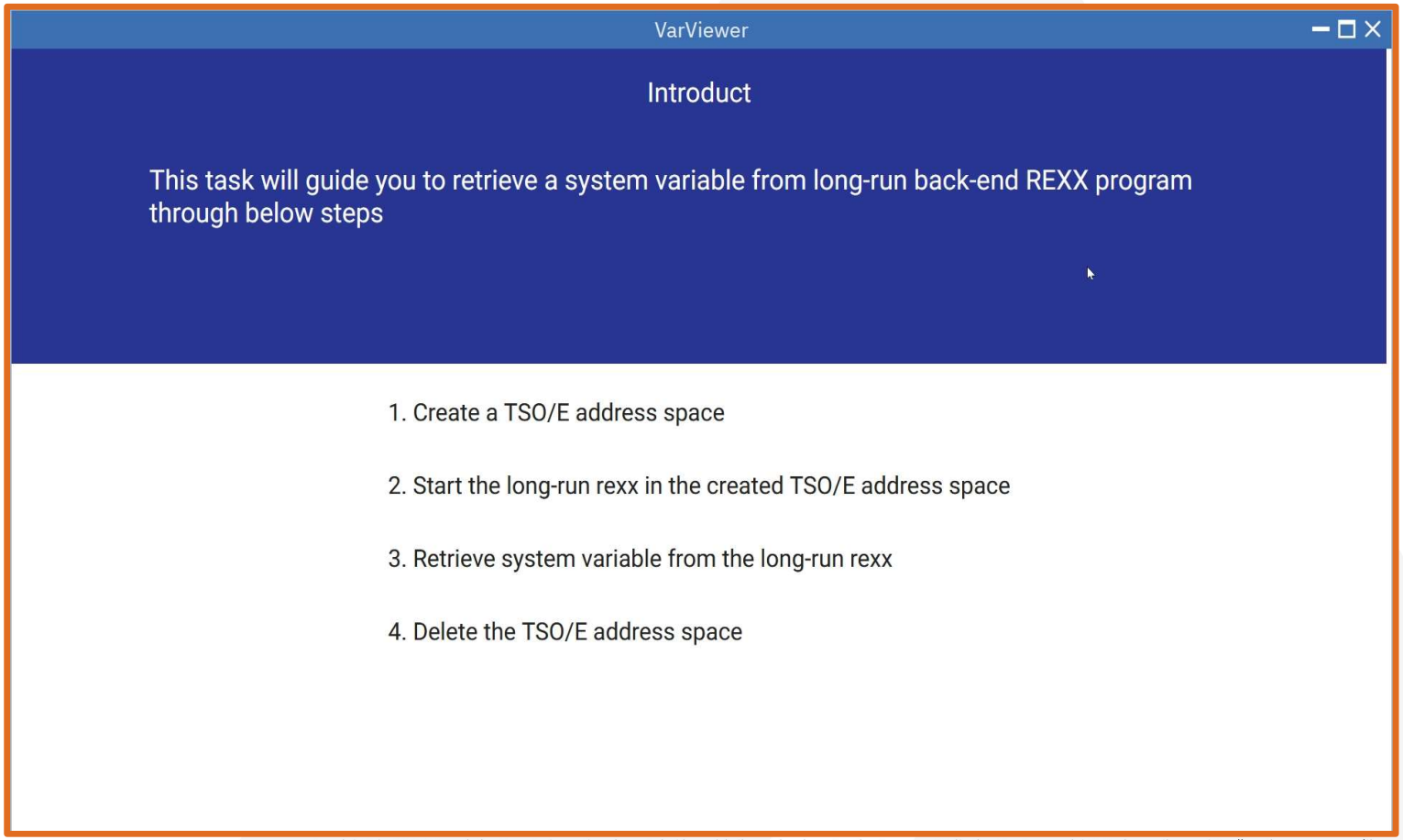
UNDERSTAND THE IBM SAMPLE PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE



This is the source code that generates the first screen.

UNDERSTAND THE IBM SAMPLE PLUGIN



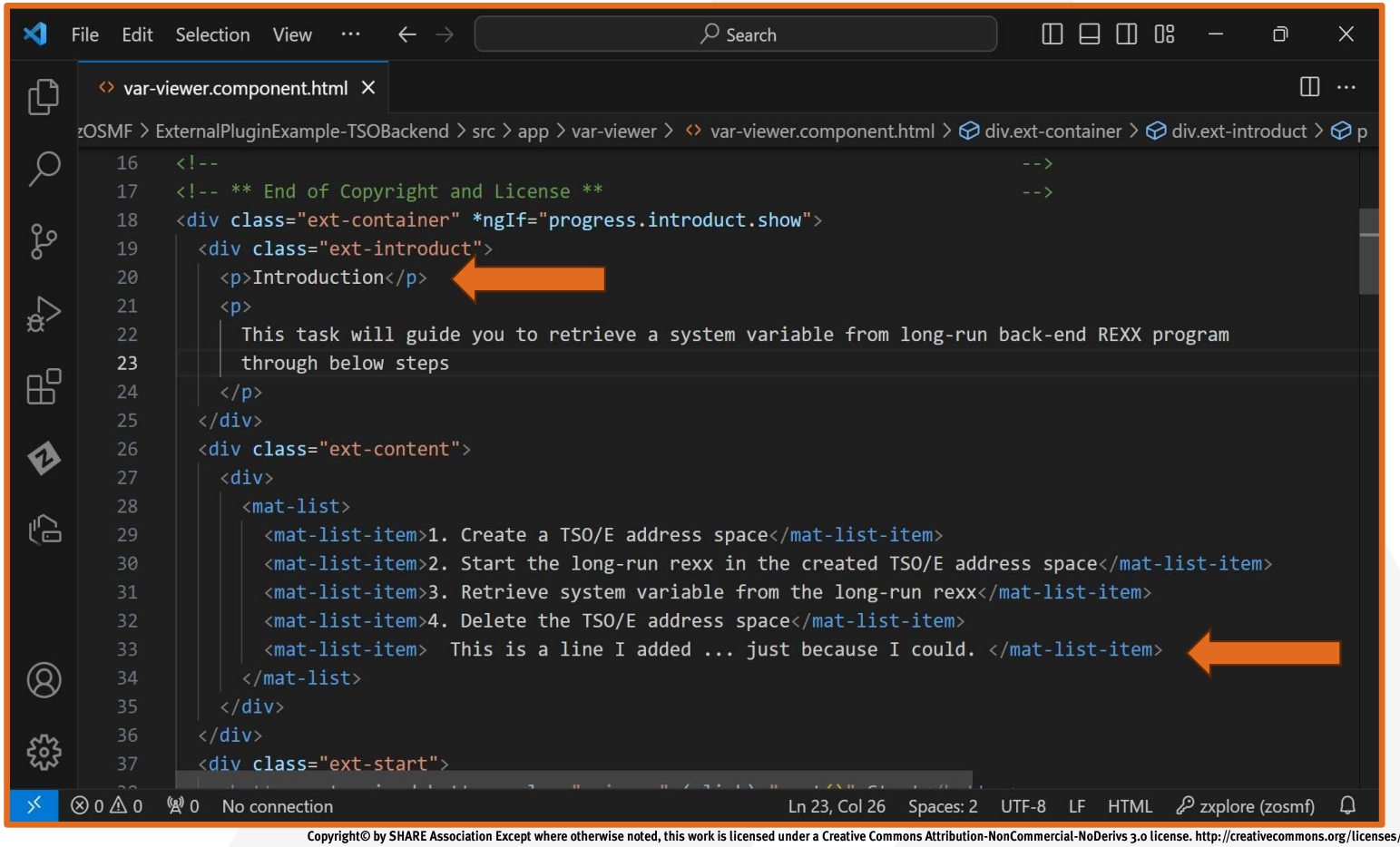
Looks great doesn't it?

A few fixes we need to make

"Introduct" spelled wrong

Needs a bit more pizzazz

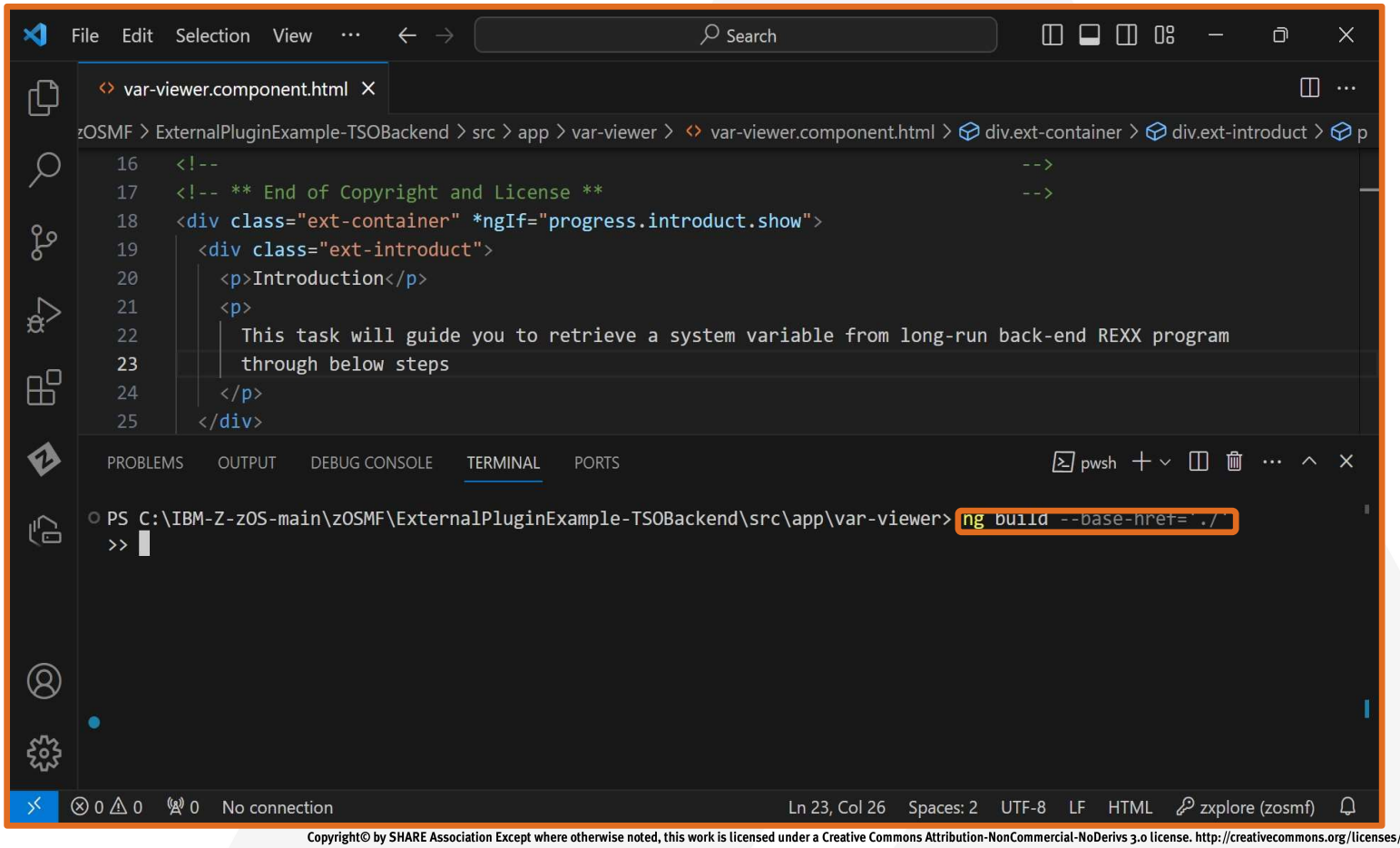
UNDERSTAND THE IBM SAMPLE PLUGIN



Let's change Introduct to Introduction

And let's add a line.

UNDERSTAND THE IBM SAMPLE PLUGIN

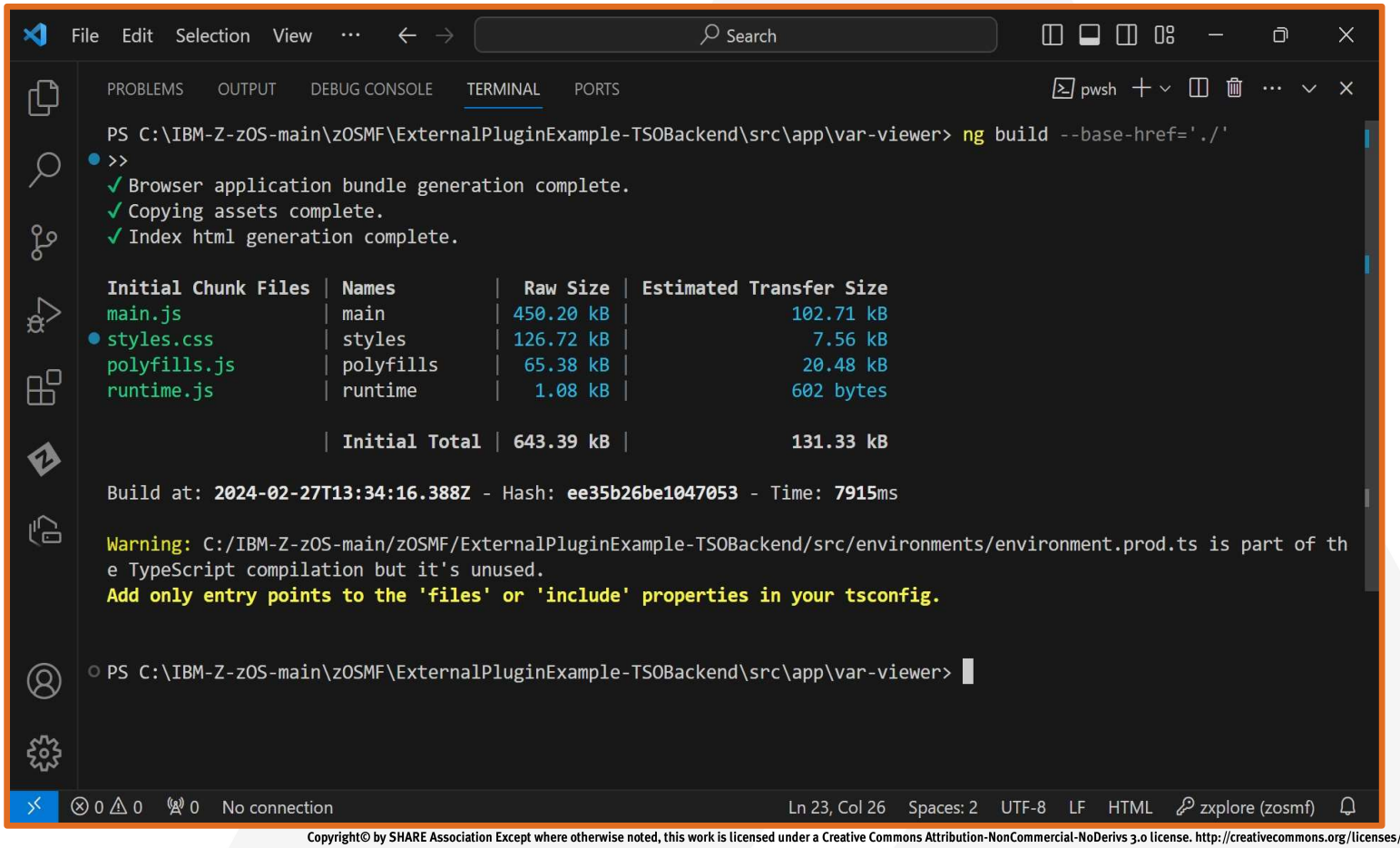


We save our changes, and now it's time to build it.

Build is analogous to a compile for a program.

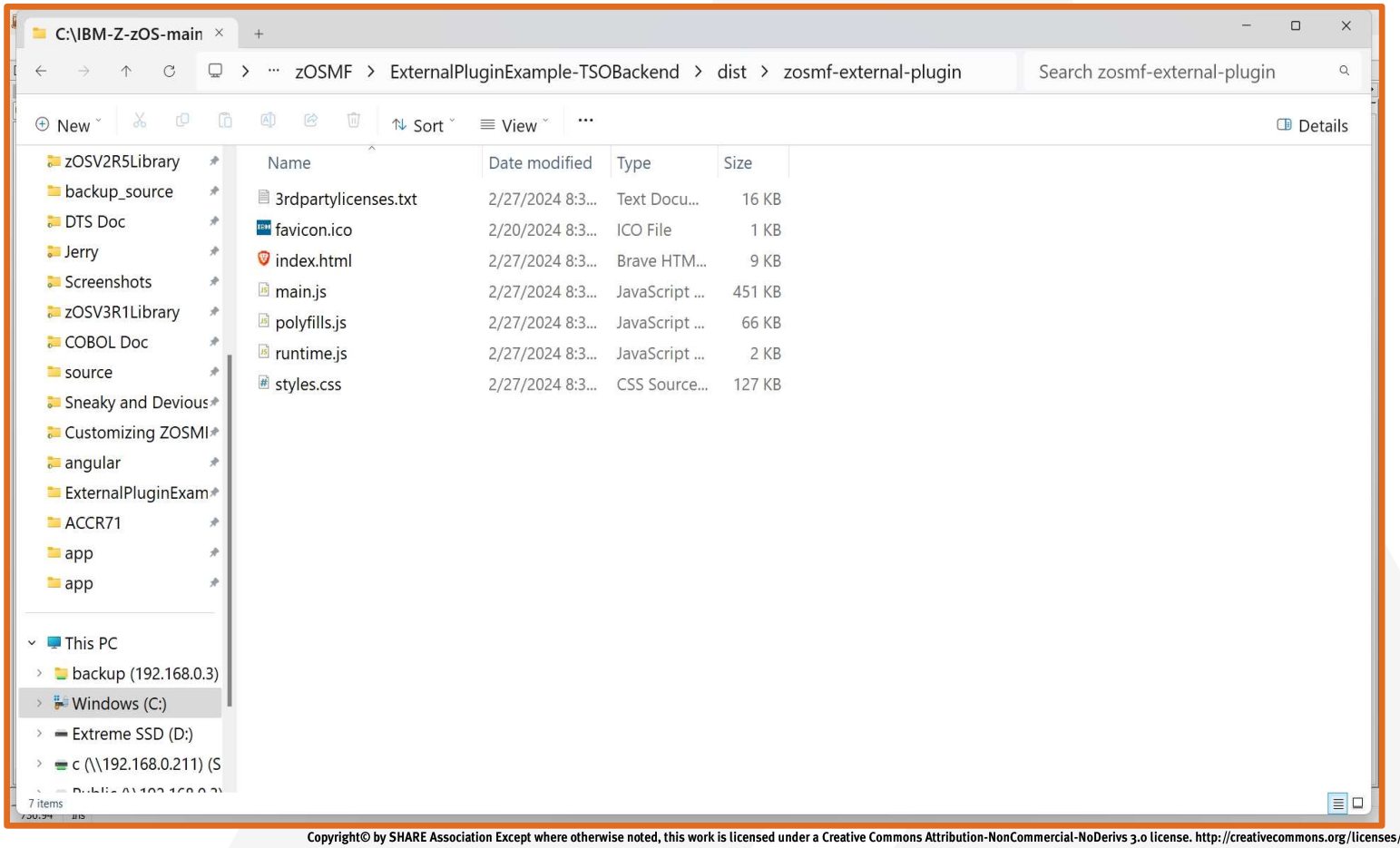
I don't know what all it does, yet. But I do know that it creates the binary files for the app.

UNDERSTAND THE IBM SAMPLE PLUGIN



The build successfully completed …

UNDERSTAND THE IBM SAMPLE PLUGIN



And these are the binary files that were created.

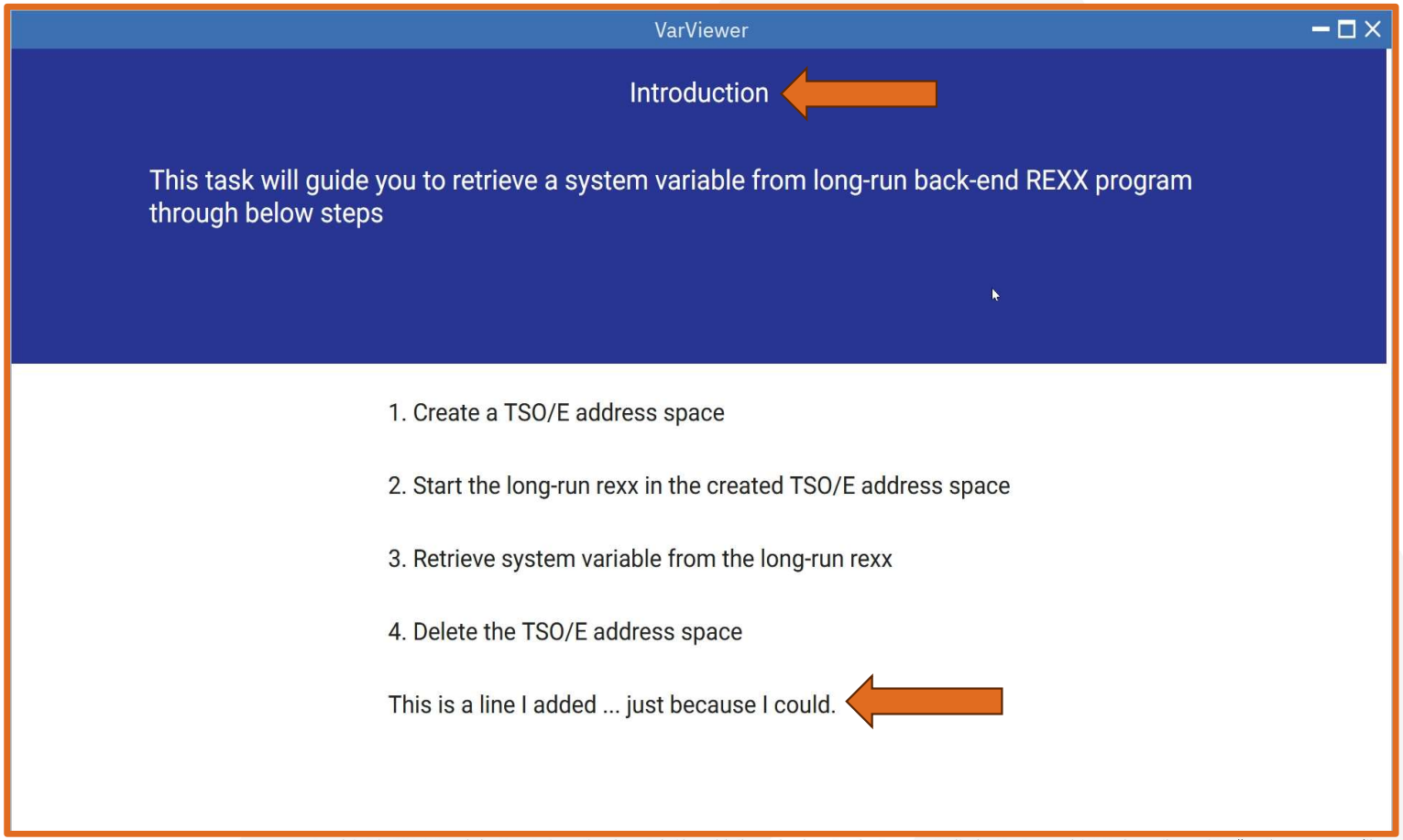
UNDERSTAND THE IBM SAMPLE PLUGIN

```
ftp> binary
200 Representation type is Image
ftp> lcd C:\IBM-Z-zOS-main\zOSMF\ExternalPluginExample-TSOBackend\dist\zosmf-external-plugin
Local directory now C:\IBM-Z-zOS-main\zOSMF\ExternalPluginExample-TSOBackend\dist\zosmf-external-plugin.
ftp> cd /usr/lpp/myextapp/dist
250 HFS directory /usr/lpp/myextapp/dist is the current working directory
ftp> mput *.*
```

Let's send the files to the mainframe

BINARY is very, very important

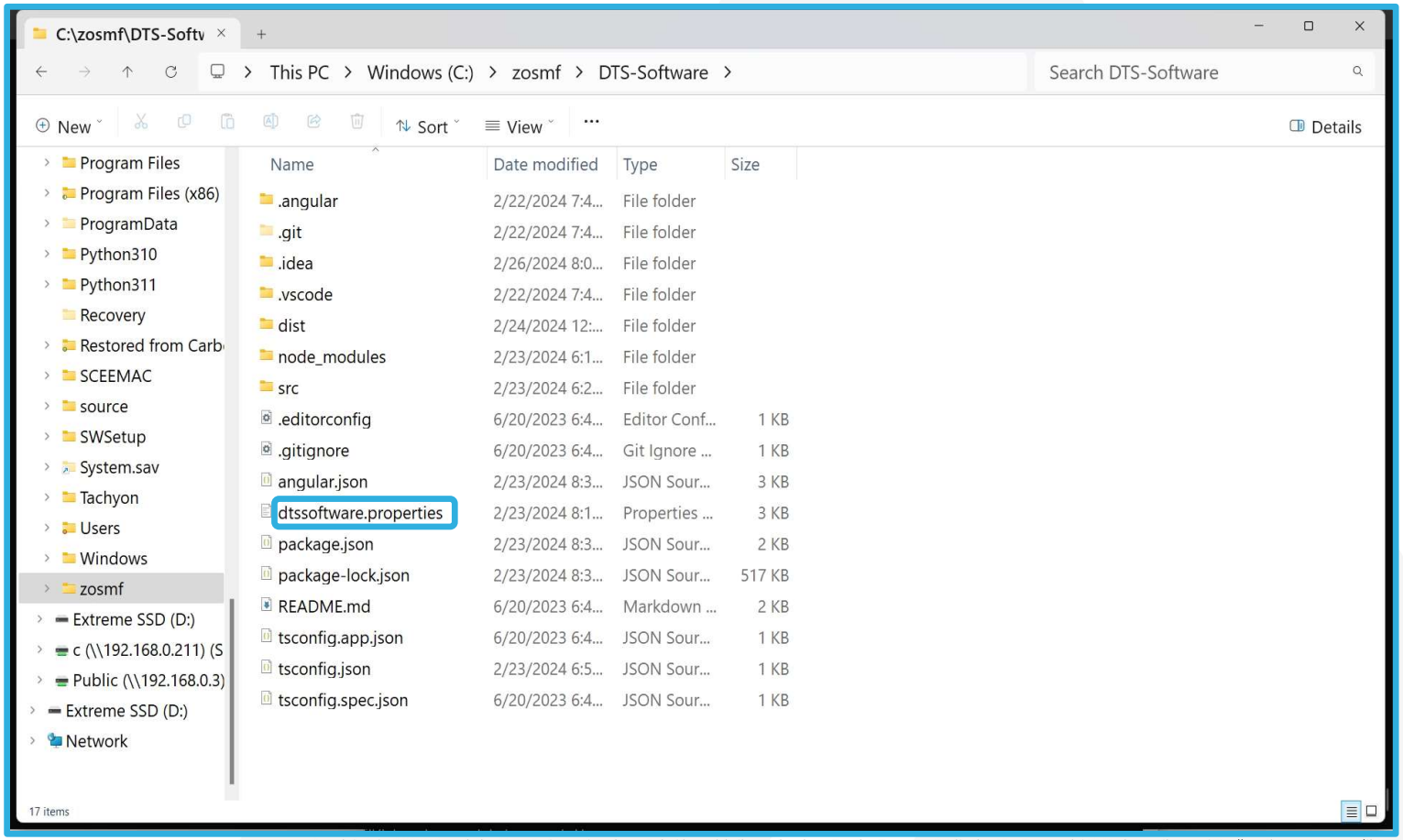
UNDERSTAND THE IBM SAMPLE PLUGIN



And here's the modified screen

MAKE YOUR OWN PLUGIN

MAKE YOUR OWN PLUGIN



I used IBM's example as a template

I renamed the properties file from myextapp.properties to dtssoftware.properties and changed it to match my requirements

MAKE YOUR OWN PLUGIN

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

```
importType=plugin

# below properties are for plugin
izu.externalapp.file.version=1.0.0
izu.externalapp.local.context.root=myextapp
izu.externalapp.code.root=dist
# izu.external.app.help.root=
# izu.external.app.helpdoc=

pluginId=MYEXTAPP
pluginDefaultName=MyExternalApp
pluginDescription=Demo_of_external_plugin
aboutPanelPath=myextapp.txt

# below properties are for task
taskId1=APPTSO
taskVersion1=1.0
taskCategoryId1=4
taskDispName1=VarViewer
taskDispDesc1=Help users to view system variables
taskSAFResourceName1=ZOSMF.IBM_MYEXTAPP.APPTSO.VARVIEWER
# taskMultiSysplexScope=true
# taskHandlerEligible=false
taskAuthenticatedGuestEligible1=true
taskNavigationURL1=index.html
taskBundleUrl1=/usr/lpp/myextapp/dist/nls
taskBundleFileName1=bundle.js
taskMinZOS1=04.25.00
taskMinZOSMF1=04.25.00
```

```
importType=plugin

# below properties are for plugin
izu.externalapp.file.version=1.0.0
izu.externalapp.local.context.root=dtssoftware
izu.externalapp.code.root=dist
# izu.external.app.help.root=
# izu.external.app.helpdoc=

pluginId=DTS_SOFTWARE
pluginDefaultName=DTS_Software
pluginDescription=DTS Software external plugin
aboutPanelPath=dtssoftware.txt

# below properties are for task
taskId1=APPTSO
taskVersion1=1.0
taskCategoryId1=4
taskDispName1=DTS_Software
taskDispDesc1=Use DTS Software
taskSAFResourceName1=ZOSMF.IBM_DTSSOFTWARE.APPTSO.DTSSOFTWARE
# taskMultiSysplexScope=true
# taskHandlerEligible=false
taskAuthenticatedGuestEligible1=true
taskNavigationURL1=index.html
taskBundleUrl1=/usr/lpp/dtssoftware/dist/nls
taskBundleFileName1=bundle.js
taskMinZOS1=04.25.00
taskMinZOSMF1=04.25.00
```

MAKE YOUR OWN PLUGIN

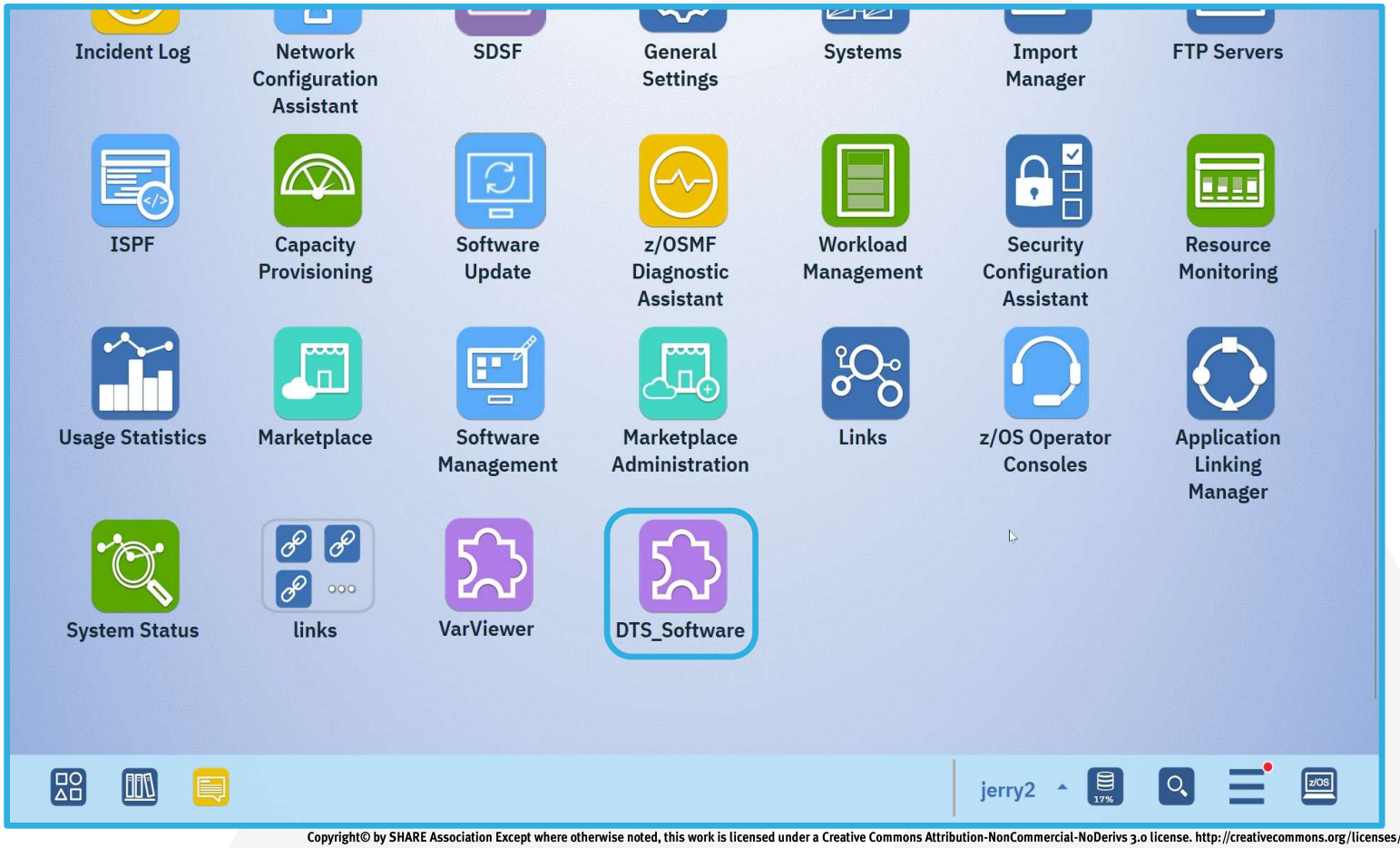
```
<div class="ext-container" *ngIf="progress.introduct.show">
  <div class="ext-introduct">
    <p>----------------- DTS Software Primary Options -----------------</p>
  </div>
  <div class="ext-content">
    <div>
      <mat-list>
        <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">DIF</button> DIF Interface</mat-list-item>
        <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">MON</button> Monitor Interface</mat-list-item>
        <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">DSL</button> DLIMIT Interface</mat-list-item>
        <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">DLM</button> DLm Interface</mat-list-item>
        <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">ZVT</button> zVT Interface</mat-list-item>
        <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">SMS</button> SMS Display, Debug, and Audit</mat-list-item>
        <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">ACS</button> ACS Routine Testing and Debugging</mat-list-item>
        <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">UTA</button> Useful Tools and Accessories</mat-list-item>
        <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">EXT</button> Easy-Exit FTP SMF Reports</mat-list-item>
      </mat-list>
    </div>
  </div>
</div>
<div class="ext-container" *ngIf="progress.createTSO.show">
  <div class="ext-introduct">
    <p>------------------------- DTS Software DIF Commands -------------------------</p>
    <div class="ext-content">
      <div>
        <mat-list>
          <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">Status</button> DIF and Product Status and Commands </mat-list-
item>
          <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">Command</button> Execute a DIF Command</mat-list-item>
          <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">Pool</button> Display Product Pool Information</mat-list-item>
          <mat-list-item>  <button mat-raised-button color="primary" (click)="next()">Variables</button> DIF Variable Functions</mat-list-item>
          <mat-list-item>  <button mat-raised-button color="primary" (click)="onExit()">Exit</button> Exit</mat-list-item>
        </mat-list>
      </div>
    </div>
  </div>
```

I then tweaked the panels from the IBM example.

Instead of IBM's mundane buttons, I replaced them with DTS's amazing products.

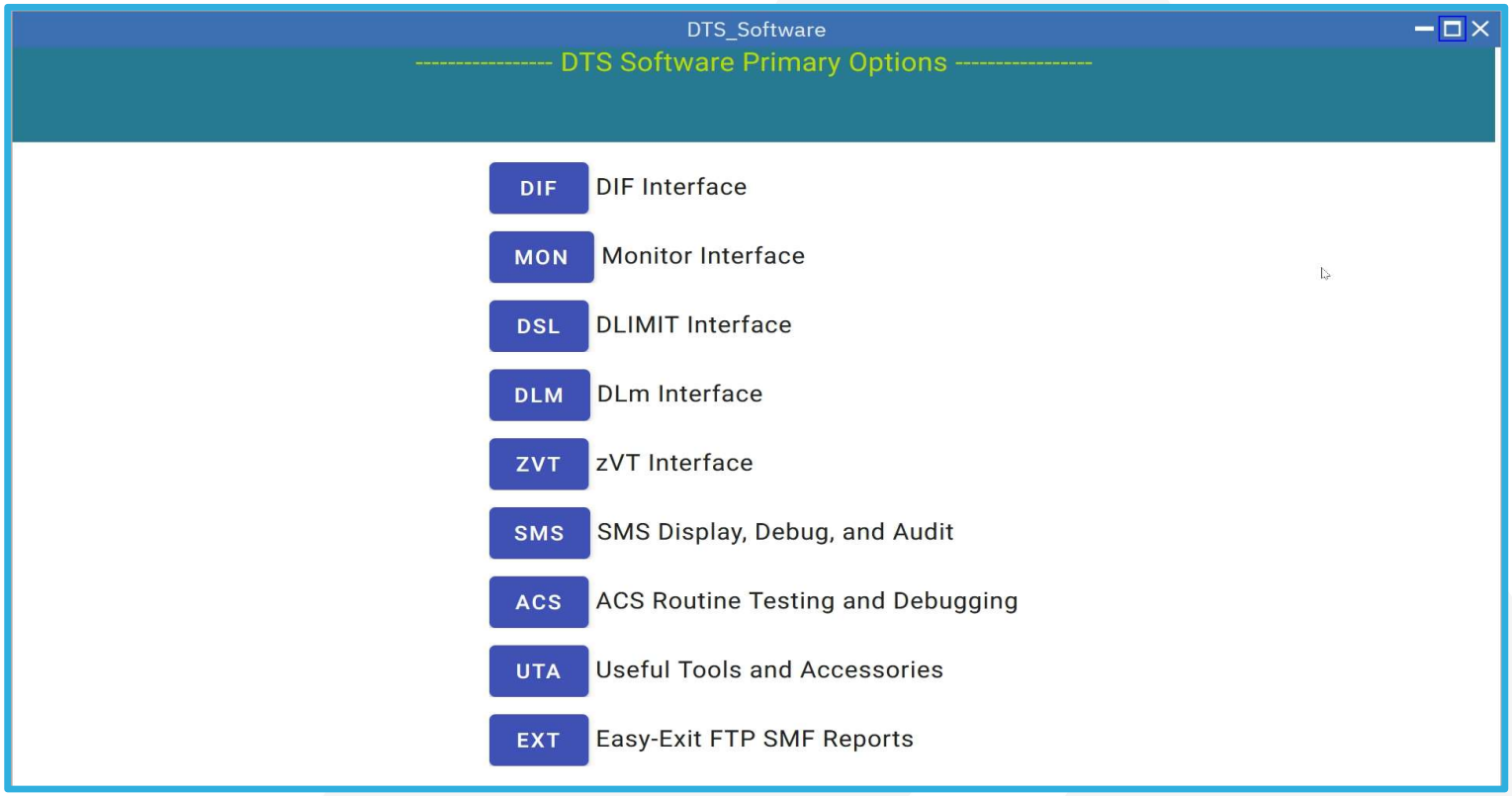
Come visit us at booth 208 ☺

MAKE YOUR OWN PLUGIN



And now I have my first rudimentary plugin

MAKE YOUR OWN PLUGIN



Here is the modernized version of our ISPF primary option page

MAKE YOUR OWN PLUGIN

This is a work in progress

Classic good news / bad news scenario

Good news is:
This is the panel for the DIF Interface button

Bad news is:
I don't know why STATUS is infringing on the blue

Only thing that works is the Exit button

I don't like the uneven descriptions

WRAP UP

WRAP UP

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

We have:

- Retrieved the IBM Example Plugin
- Installed and understood it
- Cloned it to make our own plugin
- Introduced Angular which is something else to learn

I will:

- Enhance my Angular knowledge
- Make progress on my plug-in
- Share my progress when it's all nice and pretty … and it works!

QUESTIONS

Your feedback is important!

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

Submit a session evaluation for each session you attend:

www.share.org/evaluation

SHARE mobile app