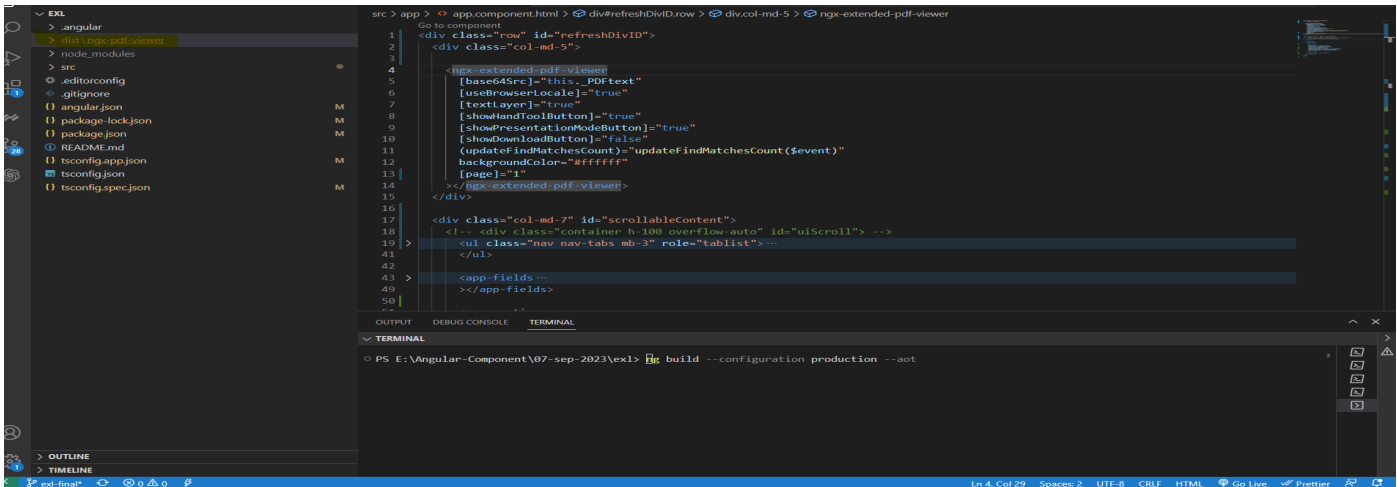# Developer Guide for Plugin

Step 1:

Production build Command: **ng build --configuration production --aot**

Step 2:

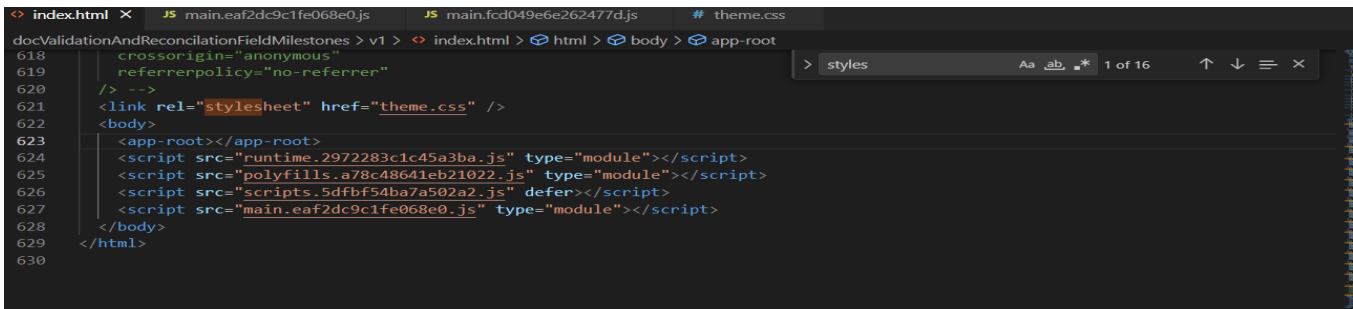After creating a production build, a 'dist' folder is created within the Angular source code package.

The specified file should be duplicated and relocated to the Appian plugin directory.

- Main.js (Start with main.)
- Polyfills.js (start with polypills.)
- Runtime.js (start with runtime.)
- Scripts.js (start with scripts.)
- Assets folder (pdf related all things)
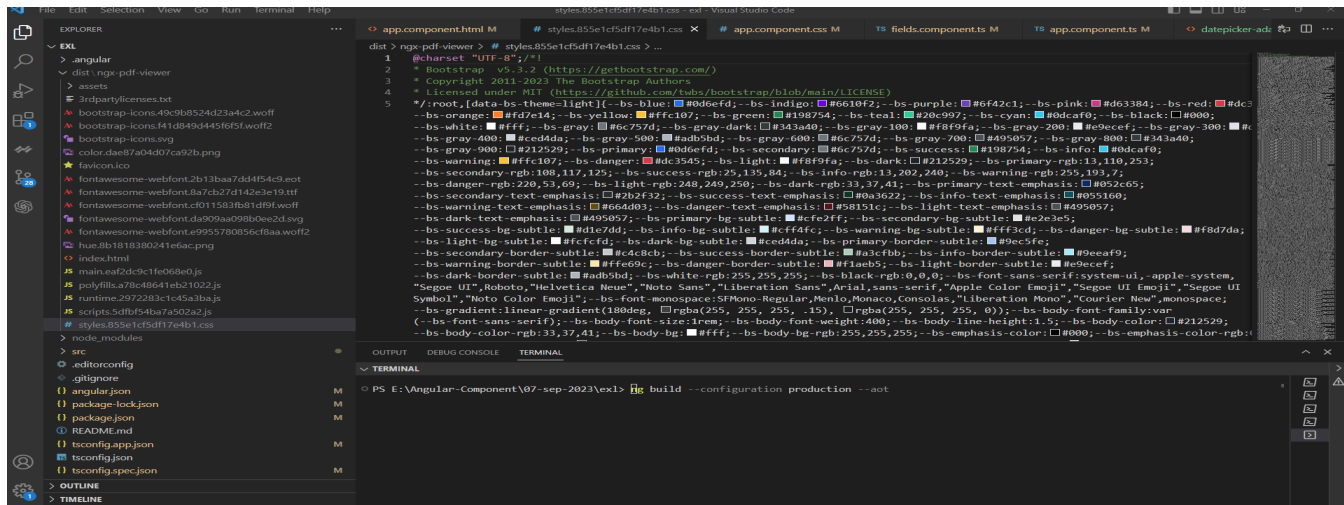
Step 3:

All minified JS files need to be referenced to the Appian plugin index.html file. Reference below attached.

Step 4:

The third one you need to copy all CSS styles in the built angular source inside the dist. folder starting with styles.css

After copying you need to paste the Appian plugin index.html file style section. For your reference sample is attached below.

Step 5:

If you want to modify any accordion related to styles you can check the theme.css file in the Appian plugin folder.

The theme.css file refers to the index.html. The reference is attached below.

Step 6:

Finally, we need to modify the start with the main.js file below mentioned things needed to be changed.

Keywords:

1.assetsFolder:

Old:

```
3943        assetsFolder: "assets",
```

New:

```
assetsFolder: "suite/rest/a/content/latest/webcontent/docValidationAndReconciliationFieldMilestones/assets",
```

Changed Text:

Suite/rest/a/content/latest/webcontent/docValidationAndReconcilationFieldMilestones/assets

2.consumeScrollEvents(Method):

Old:

```
}
```

New:

```
//e instanceof GD && (e.position ? "top" === this.options.scrollPositionRestoration ? this.viewportScroller.scrollT
```

Changed Text:

```
consumeScrollEvents() {
    return this.transitions.events.subscribe(e => {
    const lblcheck=document.getElementById('lblhidden');
    if(lblcheck!=null){
        const lblhidden = document.getElementById('lblhidden').innerHTML;
        let accordianId = 'lbl' + lblhidden.split('_')[0].toLowerCase();
            if (
                lblhidden.toLowerCase().includes('settlement') ||
                lblhidden.toLowerCase().includes('initial') ||
                lblhidden.toLowerCase().includes('instalment') ||
                lblhidden.toLowerCase().includes('noOfInstalments')||
                lblhidden.toLowerCase().includes('period') ||
                lblhidden.toLowerCase().includes('inception') ||
                lblhidden.toLowerCase().includes('expiry') ||
                lblhidden.toLowerCase().includes('written')
            ) {
                accordianId = "lblperiod";
            }
            else {}
                const elbl = document.getElementById(accordianId);
                if (null != elbl) {
                    if (e.type != undefined && e.type == 16) {
                        elbl.scrollIntoView({ behavior: "smooth" }),
                        elbl.focus()
                    }
                }
            }
        })
    }
```

3. addTranslationsUnlessProvidedByTheUser (Method):

Old:

New:

Changed Text:

```
addTranslationsUnlessProvidedByTheUser() {
   const n = document.querySelectorAll('link[type="application/l10n"]'),
   e = n.length,
   i = document.querySelector('script[type="application/l10n"]'),
   r = e > 0 || !!i;
   if (void 0 === this._useBrowserLocale && (this.useBrowserLocale = !r), r) {
      if (this.useBrowserLocale && e > 0) {
          const o = n[0].attributes.origin;
          o && "ngx-extended-pdf-viewer" !== o.value && console.error("Please set the
          attribute 'useBrowserLocale' to false if you provide the translations yourself
          in a script or link tag.")
       }
   } else {
      this.useBrowserLocale || (console.error("If you set the attribute
      'useBrowserLocale' to false, you must provide the translations yourself in a script
      or link tag."), console.error("The easiest way to do this is to add them to the
      index.html."), console.error("The PDF viewer ignores your setting and loads the
      default translations."));
      const o = this.renderer.createElement("link");
      o.rel = "resource",
      o.type = "application/l10n",
      o.setAttribute("origin", "ngx-extended-pdf-viewer"),
      this.renderer.appendChild(this.elementRef.nativeElement, o)
   }
 }
```

4. supportsOptionalChaining(Method):

Old:

New:

Changed Text:

```
supportsOptionalChaining() {
    return new Promise(n => {
        const e = window.supportsOptionalChaining;
        n(void 0 !== e ? e : e)
    })
}
```

Step 7:

Upon finishing the initial six steps, the deployable zip file for the Appian plugin will be produced.