

BPA-Benutzerhandbuch Benutzerdefiniertes MFE v4.1.2 erstellen

Inhalt

[Benutzerdefinierte MFE-Erstellung](#)

[MFE wird erstellt](#)

[Winkelbasierte Anwendungen erstellen](#)

[Erstellen von Winkelanwendungen in NX Monorepo](#)

[Allgemeine Änderungen für Winkel- und Nx-Winkelanwendungen](#)

[Änderungen bei Microservices für Onboarding von MFE-Anwendungen](#)

[Definieren von MFE-Details](#)

[Registrieren von MFE](#)

[Testen der benutzerdefinierten MFE](#)

[Lokale Einrichtung](#)

[Änderungen erstellen](#)

[Aktivieren der Seitennavigation](#)

[Hinzufügen einer neuen Route in MFE](#)

[Rendering eines geschachtelten Menüs für benutzerdefinierte Anwendungen](#)

Benutzerdefinierte MFE-Erstellung

MicroFrontEnd (MFE) ist ein Architekturmuster, bei dem verschiedene Fragmente von Benutzeroberflächen-Seiten zur Buildzeit oder Laufzeit zusammengesetzt werden. Jede MFE wird unabhängig entwickelt.

MFE wird erstellt

Anwendungen, die mit einem beliebigen Framework entwickelt wurden, können als MFE geladen werden. Der folgende Abschnitt ist ein Beispiel für das Hinzufügen einer Angular-Anwendung als MFE.

Winkelbasierte Anwendungen erstellen

1. Führen Sie den folgenden Befehl aus, um eine Angular-Anwendung mit dem Namen test-mfe zu erstellen.

```
ng new test-mfe --routing --style=scss
```

```

● [root@bpa-77 MFE]# ng new test-mfe --routing --style=scss
CREATE test-mfe/README.md (1061 bytes)
CREATE test-mfe/.editorconfig (274 bytes)
CREATE test-mfe/.gitignore (548 bytes)
CREATE test-mfe/angular.json (3106 bytes)
CREATE test-mfe/package.json (1039 bytes)
CREATE test-mfe/tsconfig.json (863 bytes)
CREATE test-mfe/.browserslistrc (600 bytes)
CREATE test-mfe/karma.conf.js (1425 bytes)
CREATE test-mfe/tsconfig.app.json (287 bytes)

```

Test-MFE

2. Führen Sie den Befehl `cd` aus, um zu `test-mfe` zu wechseln.
3. Führen Sie den folgenden Befehl aus, um ein einzelnes Spa in `test-mfe` hinzuzufügen:

```
ng add single-spa-angular@7 --project test-mfe
```

 Anmerkung: Es wird eine kompatible Version von Angular 14 und `single-spa-angular@7` verwendet.

```

● [root@bpa-77 test-mfe]# ng add single-spa-angular@7 --project test-mfe
Skipping installation: Package already installed
? Does your application use Angular routing? Yes
? What port should your project run on? 4200
  Added 'single-spa' as a dependency
  Added 'single-spa-angular' as a dependency
  Added 'style-loader' as a dependency
  Added '@angular-builders/custom-webpack' as a dependency
  Generated 'main.single-spa.ts'
  Generated 'single-spa-props.ts'
  Generated 'asset-url.ts'
  Generated 'extra-webpack.config.js'
  Using '@angular-builders/custom-webpack' builder.
  Updated angular.json configuration
  @angular-builders/custom-webpack:browser
  Warning: Since routing is enabled, an additional manual
  configuration will be required, see https://single-spa.js.org/docs/ecosystem-angular/#configure-routes
CREATE extra-webpack.config.js (303 bytes)
CREATE src/main.single-spa.ts (932 bytes)
CREATE src/app/empty-route/empty-route.component.ts (143 bytes)
CREATE src/single-spa/asset-url.ts (502 bytes)

```

Einzelner Spa im Testmodus

4. Geben Sie den Port für die lokale Entwicklung während des Aufrufs des vorherigen Befehls an. Dieselbe Einstellung wird hinzugefügt wie `deployUrl` in "angle.json".

 Anmerkung: Die `deployUrl`-Einstellung dient nur der lokalen Entwicklung.

```
"deployUrl": "http://localhost:4201/"
```

Erstellen von Winkelanwendungen in NX Monorepo

1. Führen Sie den folgenden Befehl aus, um eine Angular-Anwendung innerhalb des Arbeitsbereichs NRWL-Erweiterungen (Nx) zu erstellen:

```
npx create-nx-workspace@15.0.0 --pre-set=angular --npmScope=cisco-bpa-plattform
```

2. Geben Sie die Benutzereingaben ein, wie im folgenden Beispiel veranschaulicht:

```
[root@bpa-77 MFE]# npx create-nx-workspace@15.0.0 --preset=angular --npmScope=cisco-bpa-plattform
> NX Let's create a new workspace [https://nx.dev/getting-started/intro]
✓ Repository name · bpa-test-mfe
✓ Application name · bpa-test-mfe
✓ Default stylesheet format · scss
✓ Enable distributed caching to make your CI faster · No
> NX Nx is creating your v15.0.0 workspace.
```

Benutzereingaben

Der Arbeitsbereich bpa-test-mfe Nx wird mit der Applikation bpa-test-mfe erstellt.

3. Wenden Sie "Single-SPA-Winkel"-Schemata an, um die neu erstellte Winkel-Anwendung zu konfigurieren. Die Version ändert sich je nach Winkelversion.
4. Führen Sie den Befehl `cd` aus, um zu test-mfe zu wechseln.
5. Führen Sie den folgenden Befehl aus, um ein einzelnes SPA in bpa-test-mfe hinzuzufügen:

```
npm install single-spa-angular@7 --legacy-peer-deps && npx nx g
single-spa-angular:ng-add --project bpa-test-mfe
```

Die folgenden Dateien werden erstellt und aktualisiert:

```
Configuration will be required, see https://single-spa.js.org/docs/ecosyste
CREATE apps/bpa-test-mfe/extra-webpack.config.js
CREATE apps/bpa-test-mfe/src/main.single-spa.ts
CREATE apps/bpa-test-mfe/src/app/empty-route/empty-route.component.ts
CREATE apps/bpa-test-mfe/src/single-spa/asset-url.ts
CREATE apps/bpa-test-mfe/src/single-spa/single-spa-props.ts
UPDATE package.json
UPDATE apps/bpa-test-mfe/tsconfig.app.json
UPDATE workspace.json
```

Einzelbad in bpa-test-mfe

6. Geben Sie den Port für die lokale Entwicklung während des Aufrufs des vorherigen Befehls an. Dieselbe Einstellung wird in "project.json" hinzugefügt wie `deployUrl`.



Anmerkung: Die `deployUrl`-Einstellung dient nur der lokalen Entwicklung.

```
"deployUrl": "http://localhost:4201",
```

Allgemeine Änderungen für Winkel- und Nx-Winkelanwendungen

Überprüfen der Webpack-Version

Aktualisieren Sie die "@angular-builders/custom-webpack"-Version, die der Angular-Version entspricht. Die Version im folgenden Beispiel wird für Angular 14-Anwendungen verwendet.

```
"@angular-builders/custom-webpack": "14.1.0",
```

Auswahl aktualisieren

Aktualisieren Sie die Vorlage mit dem eindeutigen Selektor "<app-test-mfe-root>" in ./src/main.single-spa.ts.

```
const lifecycles = singleSpaAngular({
  bootstrapFunction: singleSpaProps => {
    singleSpaPropsSubject.next(singleSpaProps);
    return platformBrowserDynamic(getSingleSpaExtraProviders()).bootstrapModule(AppModule);
  },
  template: '<app-test-mfe-root />',
  Router,
  NavigationStart,
  NgZone,
});
```

Auswahl wird aktualisiert

Aktualisieren der Auswahl in der Stammkomponente

Aktualisieren Sie denselben Selektornamen "<app-test-mfe-root>" in ./src/app/app.component.ts.

```
{ } angular.json M    { } package.json M    TS main.single-spa.ts U    TS app.component.ts X
MFE > test-mfe > src > app > TS app.component.ts > ...
  You, 1 second ago | 1 author (You)
  1  import { Component } from '@angular/core';
  2
  You, 1 second ago | 1 author (You)
  3  @Component({
  4  | selector: 'app-test-mfe-root',
  5  | templateUrl: './app.component.html',
  6  | styleUrls: ['./app.component.scss']
  7  | })
  8  export class AppComponent {
  9  |   title = 'test-mfe';
 10  | }
```

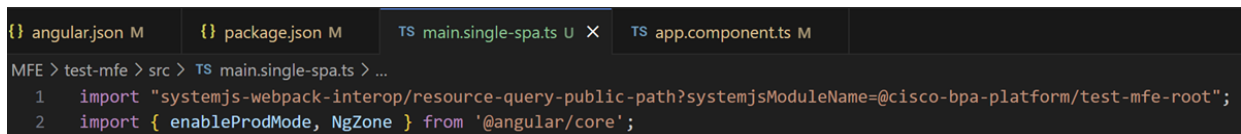
Auswahl in Stammkomponente wird aktualisiert

Installieren von Systemsjs-web-interop

Das NPM-Paket "systemsjs-web-interop" dient zur dynamischen Festlegung des öffentlichen Pfads für Ressourcen.

```
systemjs-webpack-interop": "^2.3.7",
```

Installieren Sie "systemsjs-web-interop" und fügen Sie die folgende Zeile als erste Importanweisung in die Datei "main.single-spa.ts" ein.



```
angular.json M package.json M TS main.single-spa.ts U X TS app.component.ts M
MFE > test-mfe > src > TS main.single-spa.ts > ...
1 import "systemjs-webpack-interop/resource-query-public-path?systemjsModuleName=@cisco-bpa-platform/test-mfe-root";
2 import { enableProdMode, NgZone } from '@angular/core';
```

Linie in main.single-spa.ts

systemjsModuleName muss eindeutig sein und als Name verwendet werden, der die MFE wie oben dargestellt identifiziert. Die obige Änderung hilft beim Auflösen der langsam geladenen Ressourcen vom entsprechenden Host der MFE.

Route wird aktualisiert

Schließen Sie die anwendungsspezifischen Routen in die Datei "app.routes.ts" der neuen eckigen Anwendung ein.



Anmerkung: Der folgende Pfad zu "app.routes.ts" muss enthalten sein.

```
{ path: '**', component: EmptyRouteComponent }
```

Änderungen bei Microservices für Onboarding von MFE-Anwendungen

Die meisten UI-Anwendungen sind einem Microservice per Design zugeordnet. Die folgenden Schritte in den folgenden Abschnitten sollten ausgeführt werden, damit der verknüpfte Mikrodienst die verknüpfte Benutzeroberflächenanwendung als MFE lädt.

Definieren von MFE-Details

1. Erstellen Sie eine Datei "mfe.json" in microservice/src/config/mfe.json.

```
[
  {
    "layoutenabled" : true,
```

```

    "name" : "@cisco-bpa-platform/test-mfe-root",
    "type": "application",
    "path": "ex/test-mfe",
    "display_name": "Test MFE",
    "menu_path": "Application",
    "custom": true,
    "default": false,
    "distribution_path": "/assets/cisco-bpa-platform/mw-device-manager/test-mfe/main.js",
    "version": "4.1.0-501"
  }
]

```

2. Fügen Sie die Details der MFE wie folgt hinzu:

- Benutzerdefiniert: Gibt an, ob diese Anwendung benutzerdefiniert ist.
- Layout aktiviert: Dieses Attribut aktivieren, wenn ein Header erforderlich ist
- Name: Eindeutiger Name für die MFE-Anwendung
- Anzeigename: Name, der in der Seitennavigation als Menüelement angezeigt wird
- Typ: Zeigt die MFE-Anwendung an
- Menüpfad: Gibt das Navigationsmenü an, unter dem die MFE-Anwendung als Menüelement hinzugefügt wird. funktioniert in Kombination mit benutzerdefiniertem Attribut
- Verteilungspfad: Zeigt die statische Datei an, die die MFE-Anwendung lädt.
- Pfad: Gibt die URL-Route an, der die MFE-Anwendung zugeordnet ist.
- Standard: Bei "true" sollte der Pfad nicht vorhanden sein. Wenn sie auf false gesetzt ist, ist der Pfad vorhanden.

Registrieren von MFE

Aufrufen eines Core-API-Endpunkts (z. B. POST-mfe/apps), um die entsprechenden MFE(s) zu integrieren Zur Vereinfachung des Prozesses wird eine addMfeApps-Dienstprogramm-methode erstellt, um die MFE-Informationen zu akzeptieren. So registrieren Sie die in der BPA-Datei "mfe.json" definierten MFE-Details

1. Importieren Sie mfeHelper aus der gemeinsamen Anwendung im Microservice "app.js".

```
const { mfeHelper } = require('@cisco-bpa-platform/mw-util-common-app');
```

2. Registrieren Sie MFE-Details mit der addMfeApps()-Methode. Damit wird ein Eintrag in der Sammlung "mongo db core_db -> mfe_apps" hinzugefügt.

```

async function mfeAppProcessor() {
  try {
    var data = require('./config/mfe.json');
    await mfeHelper.addMfeApps(data);
    return;
  } catch (error) {
    return error;
  }
}

```

MFE-Details registrieren

Die MFE ist bereit zum Testen.


Testen der benutzerdefinierten MFE

Lokale Einrichtung

1. Erstellen der MFE-Anwendung
2. Stellen Sie die winkelmäßige Anwendungsverteilung im Assetordner des Containers bereit, der den entsprechenden Microservice bereitstellt (z. B. /root/MFE-app/dist:/home/node/app/assets).
3. Starten Sie den Microservice-Container neu.
4. Browser aktualisieren

Änderungen erstellen

1. Aktualisieren Sie während der Image-Erstellung die Dockerdatei, um die Angular-Anwendungsverteilung in den Assets-Ordner im Container zu packen. In der Seitennavigation wird das Menü Test MFE angezeigt.
2. Klicken Sie auf das Menü Test MFE, um die MFE zu laden (URL: <https://<host>/ex/test-mfe>).

 Anmerkung: Beim Laden der neuen MFE wird die Seitennavigation nicht angezeigt, da sie nicht in die neue MFE integriert ist.

Aktivieren der Seitennavigation

1. Erstellen Sie eine neue Datei im Pfad test-mfe/src/types/decl.d.ts.
2. Fügen Sie die folgende Deklarationszeile hinzu:

```
declare module '@cisco-bpa-platform/utility-state';
```

3. Öffnen Sie "test-mfe/extra-webpack.config.js", und fügen Sie `config.externals = ['@cisco-bpa-platform/utility-state']` hinzu.

Die obigen Schritte definieren `@cisco-bpa-platform/Betriebszustand` und überspringen den Importfehler.

4. Überprüfen und installieren Sie die folgenden Abhängigkeiten:

- `@angular/cdk: "~14,2,7"`
- `"@angular/material": "~14,2,5"`
- `"@angular/localize": "^14,2,6"`
- `"@cisco-bpa-platform/cxui-components": "4.0.1-1003"`
- `"@cisco-bpa-platform/data-access-services": "4.1.1-1003"`
- `"@cisco-bpa-platform/Anmeldung": "4.0.3-1002"`
- `"@cisco-bpa-platform/shared-library": "4.0.1-1001"`
- `"@cisco-bpa-platform/spogui-components": "4.1.1-1005"`
- `"@cisco-bpa-platform/ui-shared-components": "4.1.1-1005"`
- `@cisco-bpa-platform/util: "4.0.1-1002"`
- `"@ngrx/Component Store": "~14,0,0"`
- `"@ngrx/Effekte": "~14,0,0"`
- `"@ngrx/entity": "~14,0,0"`
- `"@ngrx/router-store": "~14,0,0"`
- `"@ngrx/store": "~14,0,0"`
- `"Dateischroner": "^2,0,5"`
- Entwicklungsabhängigkeiten
- `"@types/Dateischroner": "^2.0.7"`

 Anmerkung: Es müssen die korrekten Versionen für alle cisco-bpa-Plattform-Bibliotheken verwendet werden.

5. Aktualisieren Sie die folgenden Stammdateien wie unten beschrieben:

- "app.component.html"

- "app.component.ts"

```
import { Component, HostListener } from '@angular/core';
import { Router } from '@angular/router';
import { Observable } from 'rxjs';
import { Loaderservice } from '@cisco-bpa-platform/data-access-services/auth';
@Component({
  selector: 'app-test-mfe-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
  title = 'test-mfe';
  showLoader = false;
```

LoadingSpinner\$: Observable

```
= this.loaderService.spinnerObsv$; leftNavWidth = ['50px', '250px']; isExpanded = false; t
```

- "app.module.ts"

```
import { HTTP_INTERCEPTORS, HttpClientModule } from '@angular/common/http';
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { LoaderModule } from '@cisco-bpa-platform/cxui-components/loader';
import { AuthEffects, AuthInterceptor, authReducer } from '@cisco-bpa-platform/login/feature-login';
import { SideNavModule } from '@cisco-bpa-platform/spogui-components/side-nav';
import { EffectsModule } from '@ngrx/effects';
import { StoreModule } from '@ngrx/store';
import { environment } from '../environments/environment';
import { AppComponent } from './app.component';
import { AppRoutingModuleModule } from './app-routing.module';
```

```
@NgModule({
  declarations: [AppComponent],
  imports: [BrowserModule, AppRoutingModuleModule, HttpClientModule,
    BrowserAnimationsModule,
    HttpClientModule,
    StoreModule.forRoot([{'auth': authReducer}], {
      runtimechecks: {
        strictStateImmutability: false,
        strictActionImmutability: false
      }
    }),
    EffectsModule.forRoot([AuthEffects]), LoaderModule, SideNavModule],
  providers: [
    { provide: 'config', useValue: { apiUrl: environment.apiUrl, mfa: environment.mfa } },
    {
      provide: HTTP_INTERCEPTORS,
      useClass: AuthInterceptor,
      multi: true,
    },
  ],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

- "environment.prod.ts"

```
const URL = `${window.location.href.substring(0, window.location.href.lastIndexOf('ex'))}bpa`;
export const environment = {
  production: true,
  appTitle: "BPA | ",
```

```

    appName: "Business Process Automation",
    apiUrl: `${URL}`,
    assetsUrl: "./assets",
    packageUrl: "./assets/packages",
    mfa: {
      enabled: false,
      otp_url: 'https://ssologon-prd.sm.bankofamerica.com/ssoconnect/sendOTP.html'
    },
    sync: {
      groups: false,
      users: false
    },
    autoRefreshInterval: 60000,
    showFormBuilderListItems: false,
    deploymentType: "not_on_prem"
  };

```

- "environment.ts"

```

const HOSTNAME = window.location.hostname;
const URL = `http://${HOSTNAME}: 8000`;
export const environment = {
  production: false,
  appTitle: "BPA | ",
  appName: "Business Process Automation",
  apiUrl: `${URL}`,
  packageUrl: `http://${HOSTNAME}:9090/assets/packages`,
  mfa: {
    enabled: false,
    otp_url: 'https://ssologon-prd.sm.bankofamerica.com/ssoconnect/sendOTP.html'
  },
  sync: {
    groups: false,
    users: false
  },
  autoRefreshInterval: 60000,
  showFormBuilderListItems: false,
  deploymentType: "not_on_prem"
};

```

Die Seitennavigation ist bereit zum Testen.

Hinzufügen einer neuen Route in MFE

1. Generieren Sie Komponenten der Kategorie 1 und 2.
2. Importieren Sie diese Komponenten in die Routing-Datei.

```

import { CompOneComponent } from './comp-one/comp-one.component';
import { CompTwoComponent } from './comp-two/comp-two.component';

```

3. Definieren Sie den Präfixpfad.

```

const includePath = window.location.pathname.substring(1, window.location.pathname.lastIndexOf('/ex'));
let prefixPath = 'ex';
if (includePath.length > 1) {
  prefixPath = `${includePath}/${prefixPath}`
}

```

4. Definieren Sie die Routen. Platzieren Sie die Standardroute als letzte.

```

const routes: Routes = [
  { path: `${prefixPath}/test-mfe/compone`, component: CompOneComponent },
  { path: `${prefixPath}/test-mfe/comptwo`, component: CompTwoComponent },
  { path: '**', component: EmptyRouteComponent }
];

```

5. Überprüfen Sie den vollständigen Code.

```

import { APP_BASE_HREF } from '@angular/common';
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { EmptyRouteComponent } from './empty-route/empty-route.component';
import { CompOneComponent } from './comp-one/comp-one.component';
import { CompTwoComponent } from './comp-two/comp-two.component';

const includePath = window.location.pathname.substring(1, window.location.pathname.lastIndexOf('/ex'));
let prefixPath = 'ex';
if (includePath.length > 1) {
  prefixPath = `${includePath}/${prefixPath}`
}

const routes: Routes = [
  { path: `${prefixPath}/test-mfe/compone`, component: CompOneComponent },
  { path: `${prefixPath}/test-mfe/comptwo`, component: CompTwoComponent },
  { path: '**', component: EmptyRouteComponent }
];

You, 5 minutes ago | 1 author (You)
@NgModule({
  imports: [RouterModule.forRoot(routes, { enableTracing: true })],
  exports: [RouterModule],
  providers: [
    { provide: APP_BASE_HREF, useValue: '/' },
  ],
})
export class AppRoutingModule { }

```

Überprüfungscode

Rendering eines geschachtelten Menüs für benutzerdefinierte Anwendungen

Ein geschachteltes Menü kann aus einer benutzerdefinierten Anwendung gerendert werden. In diesem Abschnitt werden die erforderlichen Schritte zur Vorbereitung des MFE-Registrierungsdokuments beschrieben. Zu diesem Zweck wurden die folgenden Eigenschaften eingeführt:

- Ebene: Akzeptiert Zahlen, beginnend bei eins; Die Zahl erhöht sich je nach Anzahl der Menüs und mit zunehmender Stufe. niedrigere Ebenen sollten zuerst in BPA registriert werden
- Eltern: Speichert den Wert der übergeordneten display_name-Eigenschaft, unter der das aktuelle Menü angezeigt werden soll

Beispiel für ein geschachteltes Menü, das für die benutzerdefinierte Test MFE-MFE-Anwendung erstellt wurde:



Verschachteltes Menü

So rendern Sie ein verschachteltes Menü:

1. Fügen Sie die richtigen Werte für die Ebene, den Pfad und die übergeordneten Eigenschaften hinzu.

```
"layout_enabled": true,
"name": "@cisco-bpa-platform/test-mfe-root",
"type": "application",
"display_name": "Test MFE",
"menu_path": "Application",
"custom": true,
"default": false,
"level": 1,
"distribution_path": "/assets/cisco-bpa-platform/mw-device-manager/test-mfe/main.js",
"version": "4.1.0-501"
```

Die Pfadeigenschaft sollte nicht vorhanden sein, wenn ein Menü über untergeordnete Menüs verfügt. Stufe 1 gibt an, dass dieses Menü zuerst registriert werden soll.

```
"layout_enabled": true,
"name": "@cisco-bpa-platform/test-mfe-root",
"type": "application",
"display_name": "Test MFE",
"menu_path": "Application",
"custom": true,
"default": false,
"level": 1,
"distribution_path": "/assets/cisco-bpa-platform/mw-device-manager/test-mfe/main.js",
"version": "4.1.0-501"
```

In der obigen Abbildung ist der übergeordnete Wert Test MFE. Daher ist das Menü Test MFE Level One unter dem Menü Test MFE verschachtelt.

Stufe 2

```
"layout_enabled": true,  
"name": "@cisco-bpa-platform/test-mfe-root",  
"type": "application",  
"path": "ex/test-mfe/compone",  
"display_name": "Test MFE One",  
"menu_path": "Application",  
"parents": "Test MFE",  
"custom": true,  
"default": false,  
"level": 2,  
"distribution_path": "/assets/cisco-bpa-platform/mw-device-manager/test-mfe/main.js",  
"version": "4.1.0-501"
```

Stufe 4

```
"layout_enabled": true,  
"name": "@cisco-bpa-platform/test-mfe-root",  
"type": "application",  
"path": "ex/test-mfe/comptwo",  
"display_name": "Test MFE Two",  
"menu_path": "Application",  
"parents": "Test MFE",  
"custom": true,  
"default": false,  
"level": 4,  
"distribution_path": "/assets/cisco-bpa-platform/mw-device-manager/test-mfe/main.js",  
"version": "4.1.0-501"
```

Stufe 5

```
"layout_enabled": true,  
"name": "@cisco-bpa-platform/test-mfe-root",  
"type": "application",  
"path": "ex/test-mfe/compone",  
"display_name": "Test MFE First one",  
"menu_path": "Application",  
"parents": "Test MFE Level One",  
"custom": true,  
"default": false,  
"level": 5,  
"distribution_path": "/assets/cisco-bpa-platform/mw-device-manager/test-mfe/main.js",  
"version": "4.1.0-501"
```

Das Menü Test MFE First One ist unter dem Menü Test MFE Level One verschachtelt.

Stufe 6

```
"layout_enabled": true,  
"name": "@cisco-bpa-platform/test-mfe-root",  
"type": "application",  
"path": "ex/test-mfe/comptwo",  
"display_name": "Test MFE First Two",  
"menu_path": "Application",  
"parents": "Test MFE Level One",  
"custom": true,  
"default": false,  
"level": 6,  
"distribution_path" : "/assets/cisco-bpa-platform/mw-device-manager/test-mfe/main.js",  
"version": "4.1.0-501"
```

Das Menü Test MFE First Two (Erste MFE testen) ist unter dem Menü Test MFE Level One verschachtelt.

Starten Sie nach dem Vorbereiten der Datei "mfe.json" in Schritt 1 den Microservice-Container neu, um die Einträge zu registrieren.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.