

# 前端开发手册

本文档对平台自定义的空工程ui-demo进行说明，包括自定义的各个配置项的意义，以及引用和处理的说明。

- ui-demo工程位置
- 工程配置文件
  - package.json
  - tsconfig.json
  - angular.json
- 发布配置文件
  - artifact\_deploy\_po  
m.xml
  - assembly.xml
- 安装组件库
- 页面入口index.html
- 资源文件
  - 样式
  - 国际化
- 主题文件
- 环境配置文件environment
- 工程文件
  - 接口定义文件url-  
config.ts
  - 国际化文件加载列表  
i18n.ts
  - 根模块app.module.  
ts
    - @NgModule的  
import
    - @NgModule的  
providers
    - 为Angular国  
际化设置语  
言
  - 根组件app.  
component.ts
- 路由
  - 懒加载
  - 路由守卫

## ui-demo工程位置

git库: `ssh://git@git.acca.com.cn:7999/opra-git/ui-demo.git`

## 工程配置文件

### package.json

name为项目名，**根据项目进行修改**。

version为项目版本。

dependencies中添加最新版本araf-operetta依赖包。该包依赖以下包，在安装araf-operetta的时候会自动安装以下依赖包。

```
"@fortawesome/fontawesome-free",
"devextreme",
"devextreme-angular",
"devextreme-intl",
"jquery",
"ng-zorro-antd",
"jqwidgets-ng"
```

#### package.json

```
{
  "name": "ui-demo",
  "version": "0.0.0",
  // .....
  "private": true,
  "dependencies": {
    "@angular/animations": "~8.2.8",
    "@angular/common": "~8.2.8",
    "@angular/compiler": "~8.2.8",
    "@angular/core": "~8.2.8",
    "@angular/forms": "~8.2.8",
    "@angular/platform-browser": "~8.2.8",
    "@angular/platform-browser-dynamic": "~8.2.8",
    "@angular/router": "~8.2.8",
    "araf-operetta": "^0.1.0",
    "rxjs": "~6.4.0",
    "tslib": "^1.10.0",
    "zone.js": "~0.9.1"
  },
  // .....
}
```

### tsconfig.json

为DevExtreme添加jszip的path

#### tsconfig.json

```
{
  // .....
  "compilerOptions": {
    // .....
    "paths": {
      "jszip": [
        "node_modules/jszip/dist/jszip.min.js"
      ]
    }
  },
  // .....
}
```

## angular.json

定义需要引入的样式和js脚本文件路径，以及dev编译配置

具体为：

projects.【项目名】.architect.build.options.assets下添加zorro样式配置。

projects.【项目名】.architect.build.options.styles下添加zorro主题文件theme.less；dev样式路径；@fortawesome样式路径；jqwidgets-ng样式路径以及工程样式styles.css路径等。

projects.【项目名】.architect.build.options.scripts下添加jquery等js脚本文件。

projects.【项目名】.architect.build.configurations下添加dev等环境编译配置。其中budgets的maximumWarning和maximumError为包允许大小，可根据项目自定义；fileReplacements用于环境配置文件替换，如下例子中，即将src/environments/environment.dev.ts文件的内容替换src/environments/environment.ts文件的内容，所以在代码中引用，均直接import { environment } from 'src/environments/environment'；即可。

#### angular.json

```
{
  // .....
  "projects": {
    "ui-demo": {
      // .....
      "architect": {
        "build": {
          // .....
          "options": {
            // .....
            "assets": [
              // .....
              "src/assets",
              {
                "glob": "**/*",
                "input": "./node_modules/@ant-design/icons-angular/src/inline-svg/",
                "output": "/assets/"
              }
            ]
          }
        }
      }
    }
  }
}
```

```

    }
  ],
  "styles": [
    "src/theme.less",
    "node_modules/devextreme/dist/css
/dx.common.css",
    "node_modules/devextreme/dist/css
/dx.light.css",
    "node_modules/devextreme/dist/css
/dx.light.compact.css",
    "node_modules/@fortawesome
/fontawesome-free/css/all.min.css",
    "src/styles.scss",
    "node_modules/jqwidgets-ng
/jqwidgets/styles/jqx.base.css"
  ],
  "scripts": [
    "./node_modules/jquery/dist
/jquery.js"
  ]
},
"configurations": {
  "production": {
    // .....
  },
  "dev": {
    "fileReplacements": [
      {
        "replace": "src/environments
/environment.ts",
        "with": "src/environments
/environment.dev.ts"
      }
    ],
    "optimization": false,
    "outputHashing": "all",
    "sourceMap": false,
    "extractCss": true,
    "namedChunks": false,
    "aot": true,
    "extractLicenses": false,
    "vendorChunk": false,
    "buildOptimizer": true,
    "budgets": [
      {
        "type": "initial",
        "maximumWarning": "10mb",
        "maximumError": "20mb"
      }
    ]
  }
}
},
},

```

```

        // .....
    }
}
},
"defaultProject": "ui-demo"
}

```

## 发布配置文件

### artifact\_deploy\_pom.xml

需要修改

- <artifactId>

<artifactId>artifact-deploy-js-【项目名】</artifactId>

- <jar.name>

<jar.name>artifact-deploy-js-【项目名】-1.0.0-release.tar.gz</jar.name>

- <cloud-front.path>

<cloud-front.path>【项目名】</cloud-front.path>

### assembly.xml

该文件不需要修改。

## 安装组件库

```
npm install --registry=http://10.1.17.149:8081/repository/acca-group/
```

## 页面入口index.html

如果需要切换DevExtreme主题则需要在<head>中添加需要切换的主题路径ini（href），并定义名称（data-theme），data-active设置为false则reload后切换主题才会生效。

```

<link rel="dx-theme" data-theme="generic.light" href="assets/css/dx.light.css" data-active="false">
<link rel="dx-theme" data-theme="generic.light.compact" href="assets/css/dx.light.compact.css" data-active="false">

```

## 资源文件

资源文件路径为src/assets/.....

## 样式

样式文件路径为src/assets/css/

用于切换样式的DevExtreme的样式文件可以放在这个目录下。

## 国际化

国际化文件路径为src/assets/i18n/

业务页面的国际化文件分为两个，【名称】-en.json和【名称】-zh.json。文件格式为：

(1) 【名称】-en.json

```
{
  "en": {
    "demoTitle": "Demo"
  }
}
```

(2) 【名称】-zh.json

```
{
  "zh": {
    "demoTitle": ""
  }
}
```

## 主题文件

zorro的主题文件路径为：src/theme.less

## 环境配置文件environment

src/environments/.....

一个环境对应一个文件，该文件用于angular.json中为对应环境配置fileReplacements属性中的“with”。

文件内容如下，其中production标识是否为prod，backend用于定义后端接口ip。

#### environment

```
export const environment = {
  production: false,
  backend: {
    ip: "https://10.1.17.141/api/demo"
  }
};
```

## 工程文件

### 接口定义文件url-config.ts

路径为src\app\url-config.ts。

文件内容如下。使用时需要import { urlConfig } from 'src/app/url-config';，如下例子中，获取Demo\_Query对应接口直接用urlConfig.Demo\_Query即可。

```
export const urlConfig = {
  Demo_Query: "/users",
};
```

### 国际化文件加载列表il8n.ts

路径为src\app\il8n.ts。

该文件定义了en与zh对应加载的文件，定义加载的形式如下：

```
declare var require: any;

export const il8n_files = {
  en: [
    require('../assets/il8n/demo-en.json'),
  ],
  zh: [
    require('../assets/il8n/demo-zh.json'),
  ]
}
```

## 根模块app.module.ts

### @NgModule的import

- 如果需要使用zorro的组件动画效果，则需要import BrowserAnimationsModule;
- 如果需要使用arraf-operetta的权限初始化方法，则需要import HttpClientModule;

## @NgModule的providers

- 如果使用哈希方式进行路由，即发布在服务器后可以进行浏览器刷新或手动输入路由地址进行路由，需要添加{ provide: LocationStrategy, useClass: HashLocationStrategy };  
需要  

```
import { LocationStrategy, HashLocationStrategy } from '@angular/common';
```
- 如果使用Http拦截进行Token验证，需要添加{ provide: HTTP\_INTERCEPTORS, useClass: ArafAuthInterceptor, multi: true };  
需要  

```
import { HTTP_INTERCEPTORS } from '@angular/common/http';  
import { ArafAuthInterceptor } from 'araf-operetta';
```
- 如果使用zorro组件，需要添加{ provide: NZ\_I18N, useValue: zh\_CN }  
需要  

```
import { NZ_I18N, zh_CN } from 'ng-zorro-antd';
```

## 为Angular国际化设置语言

因为zorro组件部分功能使用的是Angular自带的语言包，如果使用zorro需要进行registerLocaleData(zh);处理

需要

```
import { registerLocaleData } from '@angular/common';  
  
import zh from '@angular/common/locales/zh';
```

## 根组件app.component.ts

用户主题设置通过sessionStorage.getItem('theme')获取；

用户国际化设置通过sessionStorage.getItem('lang')获取。

国际化需要处理DevExtreme组件，zorro组件以及araf-operetta组件；并通过loadMessages加载本项目国际化文件。最后通过locale(【“en”/“zh”等】);进行语言转换。

通过araf-operetta包的类ArafAuthService的initAccessElements()获取有权限的菜单列表和按钮列表。

如果需要对路由地址进行转换，则可通过araf-operetta包的类ArafAuthService的getUrlParameter方法进行路由解析。

## 路由

### 懒加载

可以对模块进行懒加载，只有路由到该模块，才会加载这个模块，这样可以提升项目打开速度。写法如下（Angular8）：

```
const routes: Routes = [  
  {  
    path: 'example',  
    loadChildren: () => import('./example/  
example.module').then(m => m.ExampleModule)  
  }  
];
```

### 路由守卫

路由守卫通过定义canActivate实现。可以使用araf-operetta包中ArafAuthGuardService。根据菜单权限和用户登录状态进行路由守卫。

**注意：**路由守卫要放到**component级**，不要放到懒加载的module级。

```
import { ArafAuthService } from 'araf-  
operetta';  
// .....  
const routes: Routes = [  
  {  
    path: 'demo-demo',  
    component: DemoDemoComponent,  
    canActivate: [ArafAuthGuardService]  
  }  
];  
// .....
```