**Documentation librarie AfbCore**

**I. Installation**

1. copier le fichier afbcore-0.0.2.tgz sur un emplacement de votre machine

2. Se positionne à la racine de votre projet

3. Exécuter la commande suivante:

npm install <PATH>/afbcore-0.0.2.tgz

L'installation de cette librairie se vérifie par l'ajout de la dépendance dans le fichier package.json



Une fois cette dépendance installée, vous pouvez utiliser les fonctionnalités qu’elle offre.

Importer la dépendance le **AppModule** (fichier : app.module.ts) de votre projet comme suit :

import { AfbcoreModule, AfbcoreService } from 'afbcore';

  imports: [

    BrowserModule,

    AfbcoreModule,

    AppRoutingModule

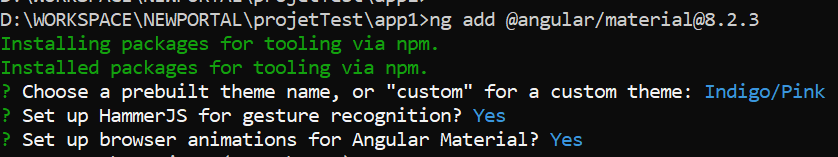
  ],

  providers: [

    AfbcoreService

  ],

4. Installer ng add @angular/material@8.2.3



5- npm i ngx-pagination

6- npm install @angular/cdk@8.2.3

7- npm i [primeicons@2.0.0](mailto:primeicons@2.0.0)

8- npm i [primeng@11.0.0](mailto:primeng@11.0.0)

9- npm i font-awesome

10- npm i ngx-permissions@8.1.1

**II. Fonctionnalités**

La librairie **AfbCore** propose les fonctionnalités suivantes :

* Création d’un menu
* Affichage du module de chargement lors de l’exécution d’un traitement
* Affichage des boites de dialogue (SUCCESS, ERROR, INFO, SESSION TIMEOUT …) ;
* La création d’une boite de dialogue personnalisé
* La création d’un formulaire (En cours)
* La création d’une table
* La centralisation de l’appel des webservice

1. **Affichage du dialogue de chargement**

La librairie **AfbCore** propose au développeur d’intégrer une boite de dialogue de chargement.

Procédure d’utilisation

Injecter dans le composant que vous voulez utiliser le service de la librairie **AfbCore**

import { AfbcoreService } from 'afbcore';

…

constructor(

    private afbcore: AfbcoreService

  ) {

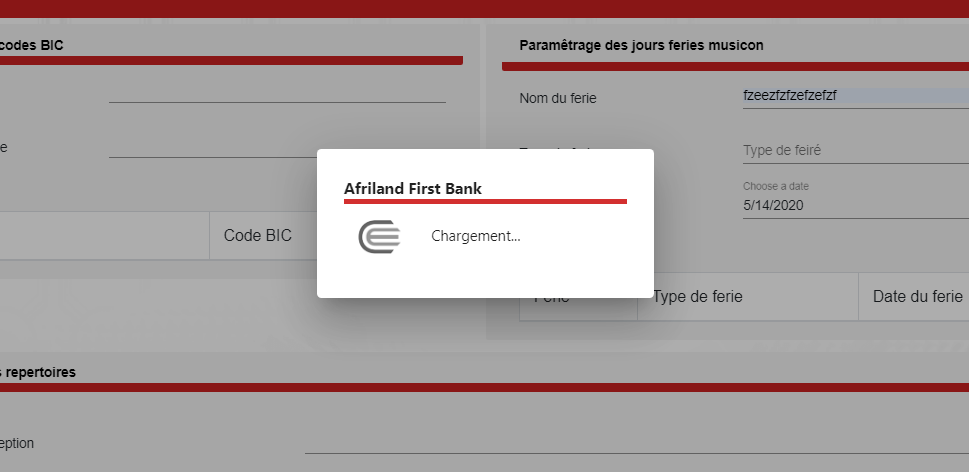
 }

Une fois le service importé, vous pouvez appeler la fonciton suivante :

this.afbcore.loading(true);

Cette fonction prend un boolean en paramètre. Ce booleen permet d’ouvrir ou fermer la boite de dialogue.

Exemple :



1. **Affichage de la boite de dialogue affichant le résultat d’une action**

La librairie **AfbCore** permet également d’afficher dans une boite de dialogue le résultat d’une action.

Procédure d’utilisation

Injecter dans le composant que vous voulez utiliser le service de la librairie **AfbCore**

import { AfbcoreService } from 'afbcore';

…

constructor(

    private afbcore: AfbcoreService

  ) {

 }

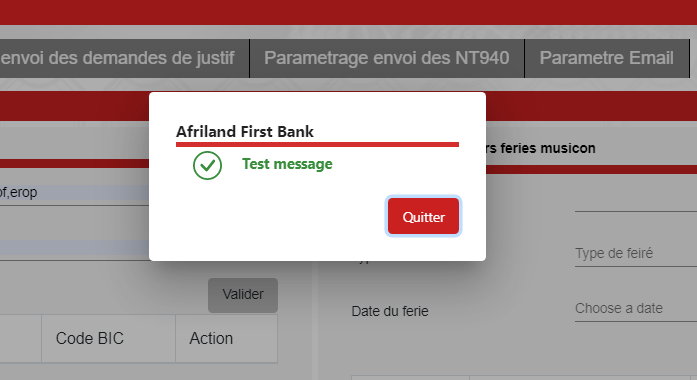
Une fois le service importé, vous pouvez appeler la fonciton suivante :

 this.afbcore.showMessage("SUCCESS", "Test message");

Cette fonction prend un deux variable de type **String** en paramètre.

* La 1ere variable définit le type du résultat (SUCCESS/DANGER/INFO/WARN)
* La 2ieme variable affiche le message que vous souhaitez.

Exemple



1. **Création d’une boite de dialogue personnalisée**

La librairie **AfbCore** permet de créer une boite de dialogue personnalisée

Procédure d’utilisation

Injecter dans le composant que vous voulez utiliser le service de la librairie **AfbCore**

import { AfbcoreService } from 'afbcore';

…

constructor(

    private afbcore: AfbcoreService

  ) {

 }

Une fois le service importé, vous pouvez appeler la fonciton suivante

this.afbcore.openDialog(

      ParamNT940FormComponent, "850px",

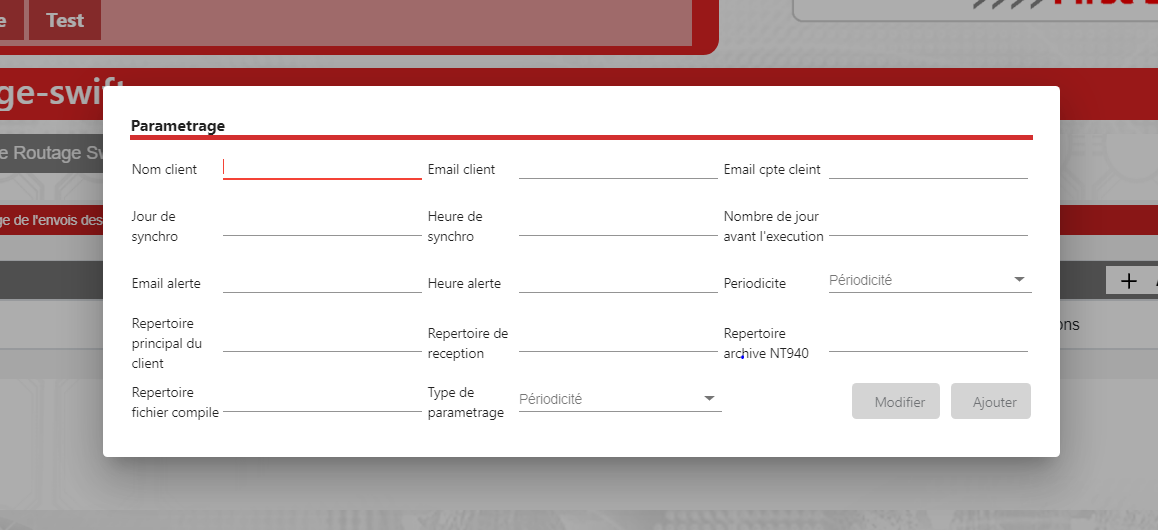
      false, null

    )

Cette fonction prend en paramètre :

* Le composant que vous voulez afficher dans la boite de dialogue
* La taille en longueur de la boite de dialogue
* Un booleen indiquant comment fermer la boite de dialogue
* Les données que l’on souhaite passer à la boite de dialogue

Exemple :



1. **Centralisation de l’appel de webservice**

La librairie **AfbCore** centralise les appels vers les webservices.

Procédure d’utilisation

Injecter dans le composant que vous voulez utiliser le service de la librairie **AfbCore**

import { AfbcoreService } from 'afbcore';

…

constructor(

    private afbcore: AfbcoreService

  ) {

 }

Une fois le service importé, vous pouvez appeler les foncitons suivantes

 callPostService(apiHost: String, path: String, object: any, action: String, role?: String): import("rxjs").Observable<any>;

    callGetService(apiHost: String, path: String, param: any, action: String, role?: String): import("rxjs").Observable<any>;

    callDeleteService(apiHost: String, path: string, param: string, action: String, role?: String): import("rxjs").Observable<any>;

    callUpdateService(apiHost: String, path: string, object: any, action: String, role?: String): import("rxjs").Observable<any>;

**apiHost** : est un parametre que nous allons charger à l’ouverture du module

1. **Creation d’une table de donnée**

La librairie afbcore propose un module permettant la création rapide d’un tableau de donnée.

Documentation :

<https://ngx-easy-table.eu/#/doc>

ou

<https://primefaces.org/primeng/showcase/#/table/dynamic>

1. **Genration d’un PDF**
2. openPdf(data: any): void;
3. /\*\*\*\*\*\*\*\*\*\*\*\*FORMAT CURRENCY \*\*\*\*\*\*\*\*\*\*\*\*\* \*/
4. formatToCurency(data: number): String;

La fonction **openPDF** permet d’ouvrir un document PDF. Cette fonction prend en paramètre un une chaine de caractère (base64).

La fonction **formatToCurency** permet de formatter un nombre au format de la devise XAF

Exemple : 1000000 => 1 000 000