v 1.0

INSTITUT NATIONAL
DE L'INFORMATION
GÉOGRAPHIQUE
ET FORESTIÈRE

ESPACE REVENDEURS

03/12/2021

Manuel d'utilisation

Présentation des
fonctionnalités du widget.

1 PRESENTATION

1.1 Le widget Espace Revendeurs

Le site Espace Revendeurs propose un accès à un widget intégrable sur votre site web via un iframe. Le widget est paramétrable et il permet : de consulter les produits, rediriger vers la page produit désirée ou encore ajouter le produit dans le panier de votre boutique en ligne.

Le widget est accessible via cette URL : https://espace-revendeurs.ign.fr/widget/

2 INSTALLATION

2.1 Installer les pré-requis

Afin de pouvoir communiquer avec les actions du widget vous devez avoir accès au module « MapIFrameAPI » de l'extension Openlayers « ol-ext » (https://github.com/Viglino/ol-ext).

Vous devez Installer ol-ext sur votre projet via npm (voir les différentes possibilités sur le site)

2.2 Intégrer l'iframe sur votre page web

Intégrez dans votre html l'iframe utilisant comme source la route du widget Espace Revendeurs :

```
<iframe id="map" src="https://espace-revendeurs.ign.fr/widget/"
frameborder="0"></iframe>
```

2.3 Configurer le widget

Créez un script permettant de communiquer entre votre site web et le contenu de l'iframe.

Il est possible de configurer l'action à l'ouverture du widget, configurer les actions des boutons, récupérer les messages d'erreurs retournés par le widget ..etc

Afin de pouvoir débuter la configuration du widget vous devez :

- Importez le module MapIFrameAPI du module ol-ext (dépends de comment a été installé « ol-ext »)

- Préparez la structure de votre script

```
MapIFrameAPI.ready('map', function(api) {

    api.on('addToCart', function(Response) {

        // avec les informations du produit (Response) faite le nécessaire pour
        // ajouter le produit au panier de votre site de vente
    });

    api.on('moreDetail', function(Response) {

        // avec les informations du produit (Response) faite le nécessaire pour
        // rediriger l'utilisateur vers la page produit demandé
    });

    api.on('errorAlert', function(Response) {

        // avec les informations de l'erreur(Response) faite le nécessaire pour
        // gérer les différents cas
    });

    api.setParamsBeforeOpening(parameters);

}) ;
```

On remarque qu'il y 3 actions à configurer « addToCart », « moreDetail » et « errorAlert » :

- **« addToCart »** : vous permet d'intercepter l'action du bouton panier sur le widget. Vous allez ainsi recevoir une réponse au format json qui contient les informations du produit (code ean, code article et titre du produit).
- **« moreDetail » :** vous permet d'intercepter l'action du lien « Plus de détails » sur le widget. Vous allez ainsi recevoir une réponse au format json qui contient les informations du produit (code ean, code article et titre du produit).
- **« errorAlert » :** vous permet d'intercepter les messages d'erreur sur le widget. Vous allez ainsi recevoir une réponse au format json qui contient des informations sur l'erreur produite.

Pour finir il est possible de configurer les paramètres d'ouverture du widget via la méthode « setParamsBeforeOpening() ». Cette méthode prend en entrée différents paramètres :

- **ean :** Ouvre le widget sur le produit ciblé vai son code EAN
- **theme :** ouvre le widget sur le thème configuré
- **position :** ouvre le widget sur la position et le niveau de zoom souhaité (format lon lat)

- **removePrice :** si « true » le widget n'affiche pas le prix du produit

```
api.setParamsBeforeOpening({
    ean: "9782758543879", // ouverture sur un produit
    theme: "DÉCOUVERTE DES VILLES", // ouverture sur un thème
    position: {lon: 5.724524, lat: 45.188529, zoom:10}, // Ouverture sur une localisation
    removePrice: true // cacher le prix des produits
});
```

Il est possible de combiner ces paramètres d'ouverture :

```
api.setParamsBeforeOpening({
    theme: "DÉCOUVERTE DES VILLES", // ouverture sur un thème
    position: {lon: 5.724524, lat: 45.188529, zoom:10}, // Puis déplacement de la vue
});
```

3 GESTION DES ERREURS

Le widget peut retourner des messages d'erreurs au format json en cas de dysfonctionnement. Voici la liste des messages susceptibles d'être rencontrés.

Type	code erreur	Message *textStatus*	Message *errorThrown*	Contexte de l'erreur	Autres données reçues en retour
Échec de communication	1000	*textStatus* de réponse	*errorThrown* de réponse	En cas d'échec dans l'exécution d'une requête récursive d'obtention des features.	-
	1001	textStatus de réponse	errorThrown de réponse	En cas d'échec dans l'exécution d'une requête simple pour l'obtention de features (Recherche pour un n°EAN, code article ou par titre)	-
Échec de Résultat	1002	thème inconnu pour ouvrir le widget	-	En cas de tentative d'ouvrir le widget sur un thème non présent dans le sélectionneur de thèmes	• theme : le nom du thème sur lequel on a tenté d'ouvrir le widget
	1003	*textStatus* de réponse	*errorThrown* de réponse	En cas d'échec dans l'exécution d'une requête pour obtenir le grid d'une commune	• zipcode : le code postal de la commune ciblée • city : le nom de la commune ciblée
	1004	Aucune emprise pour la	-	La requête pour obtenir le *grid* d'une commune a bien fonctionné mais n'a pas	• zipcode : le code postal de la commune ciblée

		commune ciblée		donné de résultat.	• city : le nom de la commune ciblée
	1005	Recherche infructueuse	-	La requête pour un ou des produit(s) lors d'une recherche par n°EAN, code article ou par titre n'a donné aucun résultat.	• search_name : le nom du paramètre de recherche (code EAN, code article ou titre) • search_value : la valeur du paramètre de recherche
	1006	Recherche infructueuse	-	La recherche de produits sur une commune donnée et pour un thème donnée est infructueuse.	• zipcode : le code postal de la commune cible • city : le nom de la commune cible • theme : le nom du thème en cours • themeID : l'identifiant du thème en cours
	1007	Recherche infructueuse	-	La recherche de produits pour un thème donné est infructueuse.	• theme : le nom du thème en cours • themeID : l'identifiant du thème en cours