

SUPPORT WEBDEV NUMERO 3



03/07/2019

Utilisation d'un Webservice Rest

L'objectif de ce support est d'utiliser un webservice Rest avec sécurisation OAuth2, d'intégrer un Javascript.

Support WebDev Numéro 3

VERSION 24

PRÉSENTATION DU TUTORIAL

Avec l'open data, de plus en plus de services publics publient leurs données. La méthode d'accès proposée est souvent via des webservices.

L'objectif de ce support est de vous proposer d'utiliser un webservice Rest de pole emploi avec identification **OAuth2**

OAuth2

De nos jours, les systèmes d'information doivent être flexibles afin de répondre aux enjeux de l'économie digitale. Cette ouverture n'est cependant pas sans conséquences et la sécurisation du droit d'accès aux ressources exposées devient, plus que jamais, critique pour l'ensemble des entreprises : c'est ici qu'intervient le « protocole de délégation d'autorisation » OAuth2.

Une demande d'autorisation avec OAuth est toujours initiée par un client. Pour tous les clients, il faudra donc les enregistrer auprès du serveur d'autorisation.

L'enregistrement nécessite au moins trois informations :

- L'identifiant du client ;
- Le mot de passe ou la paire de clés (publique/privée) pour les clients confidentiels ;
- Et une ou plusieurs URL de redirection.

Bien que les spécifications de OAuth 2.0 n'excluent pas l'utilisation de clients non-enregistrés, leur utilisation est au-delà des spécifications du protocole. Nous n'aborderons donc pas ce cas d'usage.

Le Token

La demande d'accès à une ressource protégée via OAuth se traduit par la délivrance d'un *token* au client. Le *token* représente juste une chaîne de caractère unique permettant d'identifier le client et les différentes informations utiles durant le processus d'autorisation.

Le serveur d'autorisation est en mesure d'en fournir deux types.

Token d'accès : Access token

Le *token* d'accès permet au client d'accéder à la ressource protégée. Ce *token* a une durée de validité limitée et peut avoir une portée limitée.

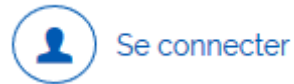
Cette notion de portée permet d'accorder un accès limité au client. Ainsi, un utilisateur peut autoriser un client à accéder à ses ressources qu'en lecture seule.

Mise en œuvre du tutoriel

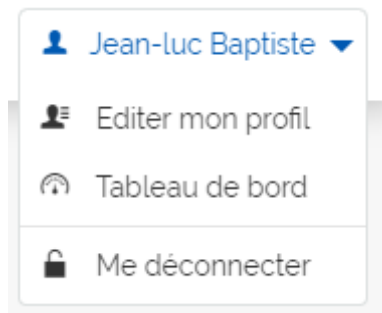
Tout d'abord il vous faut vous connecter sur le site développeur de pôle emploi :

<https://www.emploi-store-dev.fr>

Vous cliquerez sur le bouton se connecter :



Ensuite vous créez votre compte et vous renseignez votre profil. Une fois ces actions réalisées vous cliquerez sur le tableau de bord :



Dans le tableau de bord vous ajouterez une application :



**AJOUTER UNE
APPLICATION**

Donnez un nom à votre application puis cliquez sur **Ajouter une application**.

Application ajoutée



Votre application a été ajoutée avec succès

Voici les identifiants de votre application pour accéder à nos API :

Identifiant :	PAR_tutoapirest_432383dc2b12d6f95702071031487e0d54013cbf73b35e7e7e5d867
Clé secrète :	9786f8824cc989c9e8d6515eee21f893a1052edd3c46270dbe4

Vous pouvez gérer et suivre vos contrats via l'onglet « Vos contrats ».

Une fois votre application développée, vous pouvez initier une demande de référencement directement à partir de votre tableau de bord, via l'onglet « Référencement », en remplissant un formulaire de contact. Les équipes de l'Emploi Store étudieront votre demande et reviendront vers vous pour la suite.

[Commencez à développer.](#)

[Accéder au catalogue des API](#)

[Afficher mon tableau de bord](#)

Voilà, vous venez de récupérer votre identifiant et votre clé secrète.

Maintenant nous allons associer l'application à l'API désirée. Cliquez sur le bouton **Afficher mon tableau de bord**.

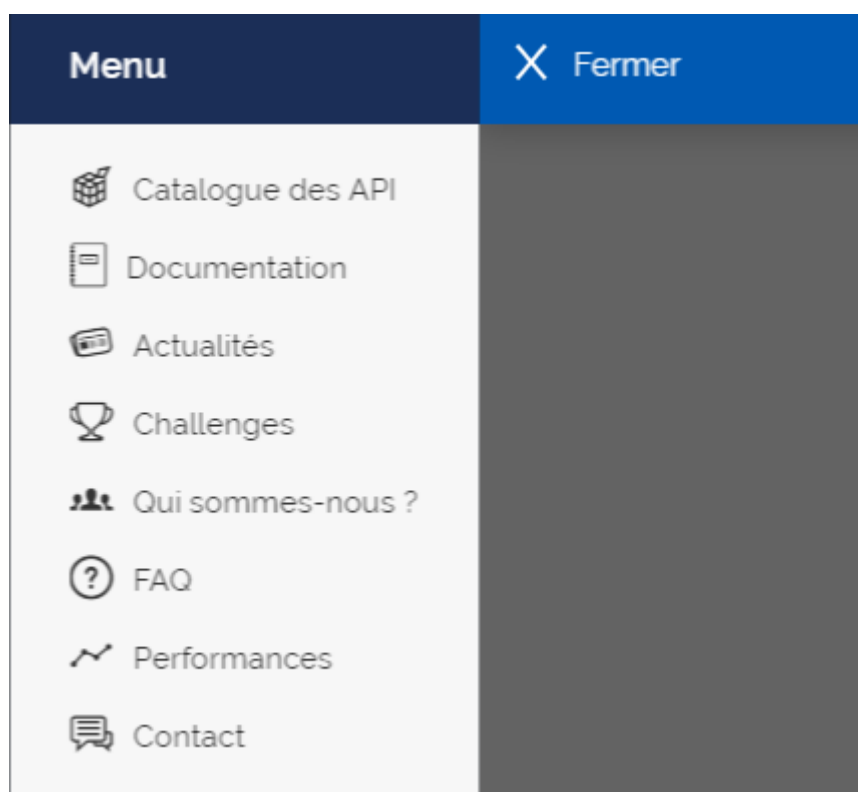
Maintenant dans le tableau de bord vous devriez voir votre application.



TutoApirest

En développement

Dans le menu, cliquez sur **Catalogue des API**.



Recherchez **Offres d'emploi Version 2**.



Cliquez dessus, vous devriez voir ceci :



Offres d'emploi

Version 2

[CONTRAT DE LICENCE](#)[DEMANDE D'ACCÈS](#)[</> TESTER L'API](#)[DOCUMENTATION](#)

L'API Offres d'emploi restitue en temps réel les offres d'emploi actives collectés par Pôle emploi. Elle vous permet de développer des solutions personnalisées de recherche d'emploi pour votre site ou votre application.

Trois ressources sont exposées pour :

- Réaliser une recherche d'offres à partir de critères de sélection et récupérer une liste de résultats paginée
- Consulter le détail d'une offre (intitulé, lieu, entreprise, contrat...)
- Restituer les référentiels utilisés par l'API Offres d'emploi (Lieu, Secteurs d'activités, contrats, formations, métiers...)


L'API Offres d'emploi offre la possibilité de filtrer sur plusieurs métiers, communes, départements, contrats

Widget(s) disponible(s) pour cette API

Un widget s'intègre au code d'un programme et accélère l'enrichissement d'un service

Requêter et restituer les offres d'emploi

Nous ce qui nous intéresse c'est d'utiliser cette API, cliquez donc sur le bouton Demande d'Accès.

 **DEMANDE D'ACCÈS**

Sélectionnez une application existante ou créez-en une





TutoApirest ▼


[Valider](#)


Sélectionnez votre application et cliquez sur **Valider** ensuite acceptez la licence.


Vous devriez maintenant voir que l'utilisation de l'api est autorisée pour votre application.


TutoApirest

 Configuration
 Vos contrats
 Statistiques
 Référencement


Offres d'emploi - V2


Accepté

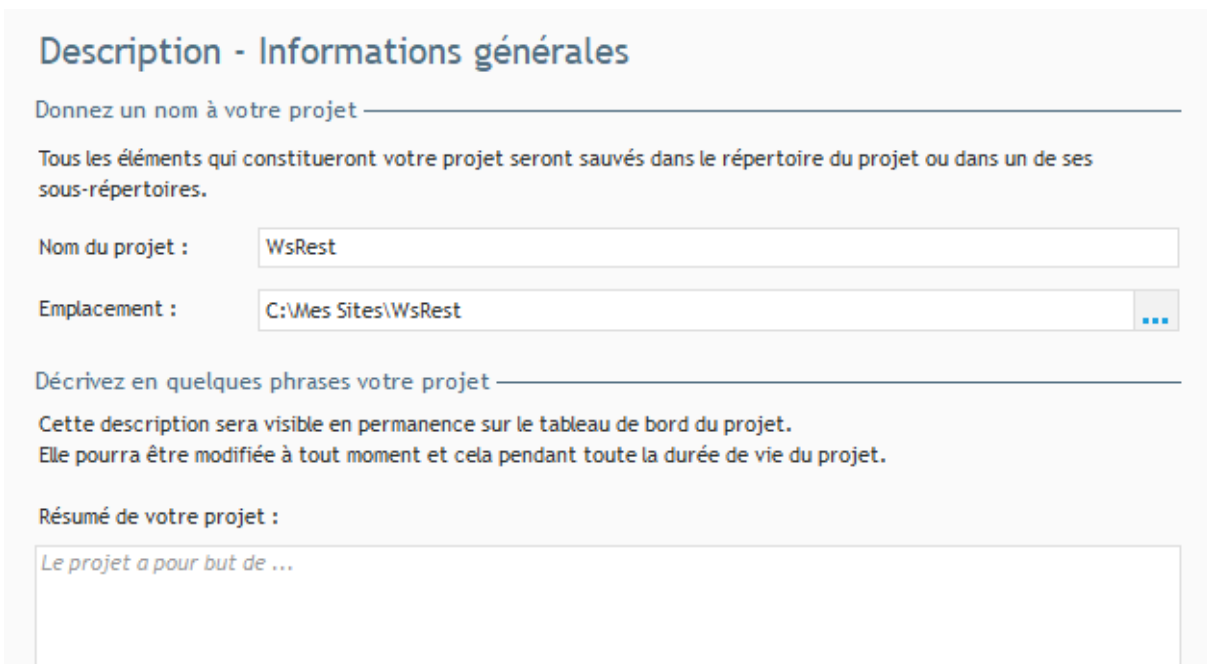

3 appels /s



Voilà, la configuration coté Pole emploi est finie, nous n'aurons besoin que de l'identifiant et de la clé secrète pour la suite.

MISE EN ŒUVRE AVEC WEBDEV

Créez un nouveau projet nommé WsRest.



Durant les étapes de l'assistant, choisissez :

- De créer un projet vierge,
- De créer un type de site WEBDEV,
- En mode Session,
- N'utilisant pas le GDS,
- Utilisant la charte de programmation standard
- Utilisez la charte graphique ou ambiance du projet que vous souhaitez,
- Au niveau de l'internationalisation du projet gardez le Français,
- Ne pas utiliser de base de données,

Ensuite créez une nouvelle page vierge que vous nommerez départ.

Sur cette page vous placerez 2 champs de saisie :

Le premier, que vous nommerez SAI_CodetInsee, permettra de saisir le code insee de votre commune pour en récupérer les offres d'emploi. Le type saisi est une chaîne de 5 caractères

Le second, que vous nommerez SAI_Distance, permettra d'indiquer le rayon de recherche en kilomètres autour de la commune. Le type saisi est un entier.


Ensuite vous placerez un bouton avec Afficher comme libellé.

Et enfin un champ Html.

Votre page devrait ressembler à ceci :

Veillez saisir le code insee de votre commune :

Distance de recherche :



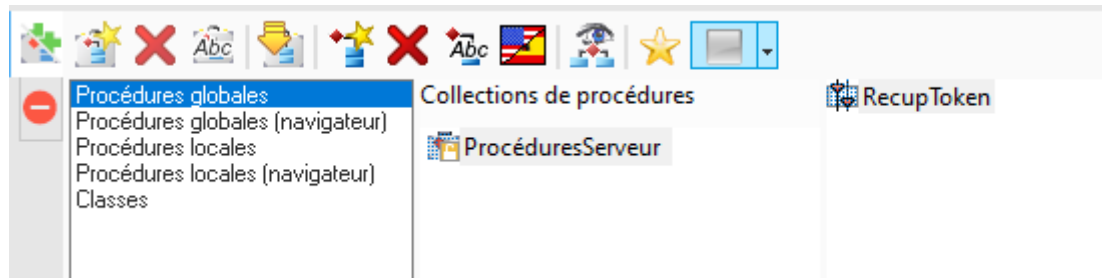
Principe de fonctionnement de la page.

Lorsque le projet se chargera, il ira initialiser le Token access qui servira pour chercher ensuite les annonces selon les 2 critères : Commune et Distance.

Vous allez commencer par déclarer une variable globale au projet nommée LeToken de type chaîne

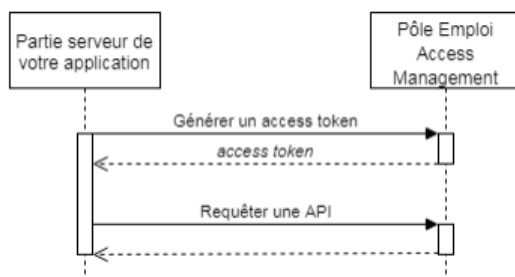
```
P Initialisation de WsRest (serveur) *
gsLeToken est une chaîne
```

Vous allez ensuite créer une procédure globale nommé RecupToken



Voici ce que pole emploi nous dit de faire pour récupérer le Token

La cinématique sans authentification utilisateur s'appuie sur le type d'autorisation [Client credentials grant](#) d'OAuth.



Elle comprend les étapes suivantes :

- **Etape 1 / Générer un access token**
L'application s'authentifie auprès de Pôle Emploi Access Management et demande un *access token* en spécifiant les API souhaitées.
Si l'application est autorisée, Pôle Emploi Access Management génère et transmet un *access token*
- **Etape 2 / Requêter une API**
L'application fournit l'*access token* à chaque interrogation d'une API

Voici les exemples fournis pour générer le token :

Point d'accès :

```
POST https://entreprise.pole-emploi.fr/connexion/oauth2/access_token
```

Détail des paramètres à valoriser :

Paramètre(s)	Valeur
realm	/partenaire
En-tête(s)	Valeur
Content-Type	application/x-www-form-urlencoded
Corps de la requête	Valeur
grant_type	client_credentials
client_id	Votre identifiant client
client_secret	Votre clé secrète
scope	Liste des scopes techniques et applicatifs correspondant aux API que vous souhaitez manipuler (séparés par des espaces)

Exemple d'appel :

```
POST /connexion/oauth2/access_token?realm=%2Fpartenaire
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
&client_id=[identifiant client]
&client_secret=[clé secrète]
&scope=application_[identifiant client]%20api_labonneboitev1
```

Nous allons donc traduire cet exemple pour le faire fonctionner avec Webdev et ainsi récupérer le Token Access. C'est le rôle de la procédure globale **RecupToken**.

Voici le code :

```
MaRequete est une restRequête
```

```
vtoken est un Variant
```

```
Lareponse est un restRéponse
```

```
// Création du corps de la requête
```

```
sCorpsRequete est une chaîne ANSI = "grant_type=client_credentials"
```

```
sCorpsRequete += "&client_id=PAR_TutoApiRest_od39a550"
sCorpsRequete += "&client_secret=30936248da8ade17435cf6af3ef1d70c8"
sCorpsRequete += "&scope=application__TutoApiRest__od39a550_api_offresemploi2
o2dsoffre"

// Encodage et préparation de la requête

sCorpsRequete = URLEncode ( sCorpsRequete )
MaRequete ..URL = URLEncode ( "https://entreprise.pole-
emploi.fr/connexion/oauth2/access_token?realm=/partenaire" )
MaRequete ..Méthode = httpPost
MaRequete .ContentType = "application/x-www-form-urlencoded"
MaRequete .Contenu = sCorpsRequete

// Envoie de la requête
Lareponse = RESTEnvoie ( MaRequete )

//Extraction des éléments de la requête
vtoken=JSONVersVariant(Lareponse.Contenu)
gsLeToken=vtoken."access_token"
```

```

PROCÉDURE RecupToken()
MaRequete est une restRequete
vtoken est un Variant
Lareponse est un restréponse

// Création du corps de la requête
sCorpsRequete est une chaîne ANSI = "grant_type=client_credentials"
sCorpsRequete += "&client_id=PAR_TutoApiRest_0d39ac96ad4f7102b8f11b23e4def7c50e2f8fe9ce53d7ec665326c62abd7550"
sCorpsRequete += "&client_secret=30936248da8ade17435cf6af3ef1d308f6ca973a4cf1d47862c197a1e8c070c8"
sCorpsRequete += "&scope=application_PAR_TutoApiRest_0d39ac96ad4f7102b8f11b23e4def7c50e2f8fe9ce53d7ec665326c62abd7550 api_offresemploi v2 o2dsoffre"

// Encodage et préparation de la requête
sCorpsRequete = URLEncode ( sCorpsRequete )
MaRequete ..URL = URLEncode ( "https://entreprise.pole-emploi.fr/connexion/oauth2/access_token?realm=/partenaire" )
MaRequete ..Méthode = httpPost
MaRequete .ContentType = "application/x-www-form-urlencoded"
MaRequete .Contenu = sCorpsRequete

// Envoie de la requête
Lareponse = RESTEnvoie ( MaRequete )

//Extraction des éléments de la requête
vtoken=JSONVersVariant(Lareponse.Contenu)
gsLeToken=vtoken."access_token"

```

Comme vous le voyez, la requête est constituée dans la variable CorpsRequete, ensuite elle est envoyée et le résultat retourné au format Json.

Champ Variant

Le champ variant est intéressant car il permet de stocker quasiment de tout :

Le type Variant permet de :

- Stocker n'importe quelle valeur **de type simple** : booléen, numérique (Monétaire, Réel, Entier, ...), caractères et chaîne de caractères, date (Date, Heure, DateHeure et Durée), ...
- Stocker des sous-éléments nommés ou indicés.
- Stocker n'importe quelle valeur **de type complexe** : structures, classes, types avancés, tableaux, tableaux associatifs, files, piles et listes.
- Manipuler la valeur NULL dans le WLangage.
- Stocker des interactions avec les objets ActiveX et la programmation Automation.

Ici l'astuce consiste à transformer la valeur retournée au format Json en variant. L'intérêt est ensuite de pouvoir rechercher directement et facilement la valeur de la rubrique access_token. C'est le but de la dernière ligne du code qui récupère juste le token et l'affecte ensuite dans notre variable globale.

Passez cette procédure en mode Ajax et faites-en un thread :

Automatisme de la procédure

Type

Aucun

Thread

Thread Principal

Tâches WEBDEV

Comment ?

Nombre de fois : fois Infini

Espacement entre les appels : (HH:MM:SS:CC)

Sans utilisation de HFSQL

Avec utilisation de HFSQL : copie de contexte complète (connexions, positions, requêtes, etc...)

Avec utilisation de HFSQL : copie de contexte légère (connexions uniquement)

Quand ?

Immédiatement au moment de l'appel

Démarrage automatique (après le code d'initialisation du projet), après un délai de (HH:MM:SS:CC)

Différé de (HH:MM:SS:CC), après l'appel de la procédure

Ala fin du traitement contenant l'appel

Ala fin de l'événement contenant l'appel

Maintenant dans le code d'initialisation du projet appelez la procédure :

```

gsLeToken est une chaîne
RecupToken()

```

Maintenant dès que le projet sera lancé, le token sera récupéré et placé dans la variable globale LeToken.

Passons maintenant au code du bouton de la page principale.

Veuillez saisir le code insee de votre commune :

Distance de recherche :

Comme vous l'imaginez, il utilisera le token, le code insee de la commune et la distance de recherche.

Voici ce que nous dit le site pôle emploi pour l'utilisation de leur Widget :

Présentation

Ce widget permet de bénéficier d'une interface préconstruite pour requêter et afficher sur une carte les offres d'emploi issues de l'API Offres d'emploi.

Le widget s'appuie sur [StencilJS](#) qui utilise la technologie Web Component.

Il s'agit d'une première version du widget, des évolutions sont possibles. [N'hésitez pas à nous faire part de vos retours.](#)

Pré-requis

Le widget est couplé à l'API Offres d'emploi. Il est donc nécessaire de remplir les conditions suivantes :

1. Se créer un compte sur l'Emploi Store Développeurs
2. [Demander l'accès à l'API Offres d'emploi](#)
3. Être en mesure de faire des appels sur l'API Offres d'emploi :
 - [Utiliser les API](#)
 - [Mise en place de la mécanique de sécurité](#)

Récupération du widget

Le widget est disponible à l'url suivante : <https://www.emploi-store-dev.fr/widget/pe-offres-emploi/pe-offres-emplois>

Afin d'intégrer le widget dans votre application, vous devez mettre un lien vers cette librairie.

Et voici un exemple complet d'utilisation :

Exemple complet d'intégration en HTML

```

<!DOCTYPE html>
<html lang="fr">
<head>
  <script src="https://www.emploi-store-dev.fr/widget/pe-offres-emploi/pe-offres-emploi.js"></script>
</head>
<body>
  <pe-offres-emploi></pe-offres-emploi>
  <script >
    var macarte = document.querySelector('pe-offres-emploi');
    macarte.options = {
      rechercheAuto: false,
      technicalParameters: {
        range: { value: '0-49', show: false, order: 1},
        sort: { value: 1, show: true, order: 3}
      },
      criterias: {
        domaine: { value: null, show: false, order: 1},
        codeROME: { value: null, show: false, order: 1},
        appellation: { value: null, show: false, order: 1},
        theme: { value: null, show: false, order: 1},
        secteurActivite: { value: null, show: false, order: 1},
        experience: { value: null, show: false, order: 1},
        typeContrat: { value: null, show: true, order: 2},
        natureContrat: { value: null, show: false, order: 1},
        qualification: { value: null, show: false, order: 1},
        tempsPlein: { value: null, show: true, order: 4},
        commune: { value: null, show: true, order: 1},
        distance: { value: null, show: false, order: 1, min: 0, max: 100},
        departement: { value: null, show: false, order: 1},
        inclureLimitrophes: { value: null, show: false, order: 1},
        region: { value: null, show: false, order: 1},
        paysContinent: { value: null, show: false, order: 1},
        niveauFormation: { value: null, show: false, order: 1},
        permis: { value: null, show: false, order: 1},
        motsCles: { value: null, show: true, order: 0},
        salaireMin: { value: null, show: false, order: 1, min: 0, max: 100},
        periodeSalaire: { value: null, show: false, order: 1},
        accesTravailleurHandicape: { value: null, show: false, order: 1},
        publieeDepuis: { value: null, show: false, order: 1}
      }
    };
    macarte.token = "mon token widget offre";
  </script>
</body>
</html>

```

C'est ce code que nous allons adapter et envoyer au composant html. Si vous regardez bien vous verrez la ligne correspondante à la commune, celle de la distance et le token. Les modifications porteront là car c'est ici que nous passerons les variables.

Voyons notre code sur le clic du bouton :

```
sMapage est une chaîne=[
<!DOCTYPE html><html lang="fr">
  <head>
<script src="https://www.emploi-store-dev.fr/widget/pe-offres-emploi/pe-offres-emploi.js"></script>
  </head>
  <body>
    <pe-offres-emploi></pe-offres-emploi>
<script >
  var macarte = document.querySelector('pe-offres-emploi');
  macarte.options = {
    rechercheAuto: true,
    technicalParameters: {
      range: { value: '0-49', show: false, order: 1},
      sort: { value: 1, show: true, order: 3}
    },
    criterias: {
      domaine: { value: null, show: false, order: 1},
      codeROME: { value: null, show: false, order: 1},
      appellation: { value: null, show: false, order: 1},
      theme: { value: null, show: false, order: 1},
      secteurActivite: { value: null, show: false, order: 1},
      experience: { value: null, show: false, order: 1},
      typeContrat: { value: null, show: true, order: 2},
      natureContrat: { value: null, show: false, order: 1},
      qualification: { value: null, show: false, order: 1},
      tempsPlein: { value: null, show: true, order: 4},
```



```

        commune: { value: "%2", show: true, order: 1},
        distance: { value: %3, show: false, order: 1, min: 0, max: 100},
        departement: { value: null, show: false, order: 1},
        inclureLimitrophes: { value: null, show: false, order: 1},
        region: { value: null, show: false, order: 1},
        paysContinent: { value: null, show: false, order: 1},
        niveauFormation: { value: null, show: false, order: 1},
        permis: { value: null, show: false, order: 1},
        motsCles: { value: null, show: true, order: 0},
        salaireMin: { value: null, show: false, order: 1, min: 0, max: 100},
        periodeSalaire: { value: null, show: false, order: 1},
        accesTravailleurHandicape: { value: null, show: false, order: 1},
        publieeDepuis: { value: null, show: false, order: 1}
    }
};
macarte.token = "%1";
</script>
</body>
</html>
]
HTM_Carte=ChaîneConstruit(sMapage,gsLeToken,SAI_CodeInsee,SAI_distance)
HTM_Carte.Visible=Vrai

```

Explications :

On construit le code html que l'on affecte à la variable Mapage. Ensuite dans le code on installe la récupération des paramètres qui va se faire grâce à l'utilisation de %X.

Regardez la ligne macarte.token= « %1 ». Le %1 représente la première variable de la ligne ChaîneConstruit(sMapage,gsLeToken,SAI_CodeInsee,SAI_distance). Vous comprenez maintenant où vous trouverez le %2 et le %3 !

Rusé non ?

Modification du champs Html

Vous allez le nommer HTM_Carte, le rendre invisible.

Voilà le tour est joué. Lancez le projet

Veillez saisir le code insee de votre commune :


Distance de recherche :

Voici un exemple d'informations saisie.

Et voici le résultat à l'exécution du bouton remplir :

Veillez saisir le code insee de votre commune :

Distance de recherche :

 Offres d'emploi ✕

Aide caviste
Publiée le 1 juillet 2019

STE COOPERATIVE VINICOLE DE TERRATS
66 - TERRATS - 66300

Cave coopérative des Aspres recherche des aides cavistes pour les vendanges 2019. Vous serez affecté aux différentes tâches liées à la réception des vendanges, au pressurage et la filtration des vins ainsi que le nettoyage des cuivés et du matériel. Nous attendons des

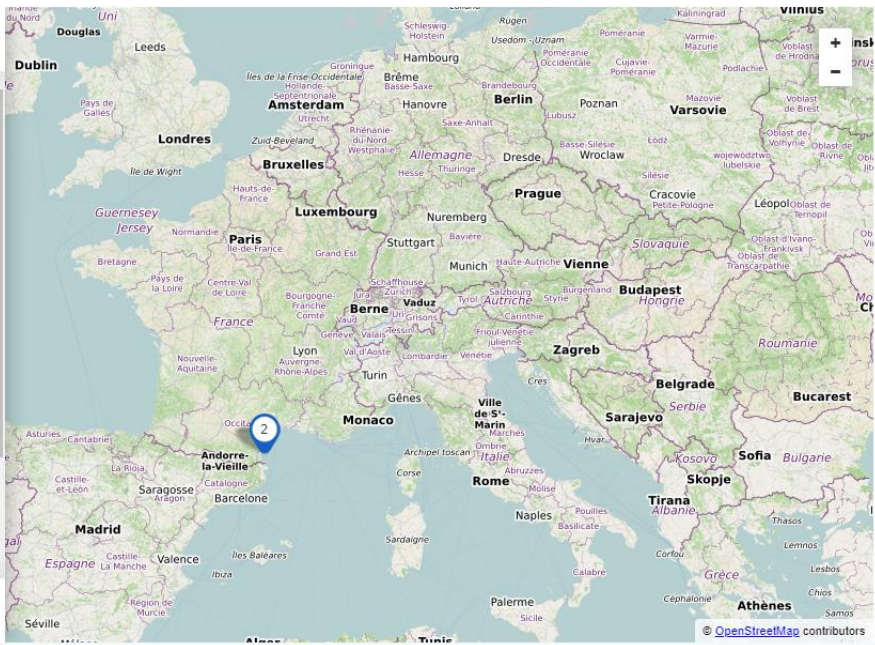
[Lire la suite](#)

Chauffagiste (H/F)
Publiée le 19 juin 2019

AURELIEN ESCRIBANO
66 - PONTEILLA - 66300

recherche chauffagiste sachant faire les entretiens sur chaudières Gaz/Fioul ainsi qu'un

2 offres
1 - 2



Voilà vous venez d'utiliser un webservice Rest avec authentification Oauth2, vous avez intégré et paramétré un javascript ! Bravo !

Un grand merci à Fabrice Chambon et Marc pour l'aide apportée pour la construction de la requête d'authentification.