

Introduction

Cette API permet de faciliter l'usage des web-services search et acp d'Antidot. L'objectif est alors de proposer une auto-complétion simple et optimisée s'appuyant uniquement sur du paramétrage.

Sommaire

- [Dépendances](#)
- [Instanciation](#)
 - [Paramètres globaux](#)
 - [Paramètres de l'acp](#)
 - [Paramètres du search](#)
- [Format de retour des données](#)
- [Affichage de l'auto-complétion](#)
 - [Choix des templates](#)
 - [Personnaliser les noms des catégories](#)
 - [Affichage des options](#)
 - [La hiérarchie des facettes](#)
 - [La séparation en colonnes](#)
- [Modifier le CSS](#)
- [Configurer son proxy avec MAMP](#)

Dépendances

Il faut importer dans l'ordre :

```
./js/jquery-1.11.1.js  
./js/jquery-ui.js  
./js/DecodeJson.js  
./js/antidot.js  
./js/Param.js  
./js/RenderUtils.js  
./js/templates.js  
./js/autocomplete.js
```

Et enfin pour les feuilles de css :

```
./css/jquery-ui.css  
./css/style.css
```

Instanciación

Pour initialiser un objet antidot, il faut 1 objet :

```
var antidot=Antidot.init({
  common : initParam,
  acp : initParamAcp,
  search : initParamSearch,
  searchacp : initParamSearchAcp,
});
```

Une fois l'objet antidot initialisé, on l'associe à un input :

```
Antidot_Initialize("#autocomplete", antidot, "#search");
```

La fonction Antidot_Initialize prend en argument :

1. L'id de l'input correspondant à la barre de recherche.
2. L'objet Antidot créé précédemment.
3. L'id du bouton de recherche

Paramètres globaux

initParam est un objet contenant les informations communes à l'acp et au search.

Paramètre	Utilité	Obligatoire	Type de valeur
acp	Permet d'utiliser le web-service acp lors du premier appel.	Oui	Booléen
search	Permet d'utiliser le web-service search lors du premier appel.	Oui	Booléen
searchacp	Permet d'utiliser le web-service search lors du second appel.	Oui	Booléen
serviceld	Indique votre numéro de service.	Oui	Integer
status	Indique le statut du service.	Oui	String
domain	Votre nom de domaine, ou si nécessaire, l'adresse du proxy.	Oui	String
language	Indique le langage voulu (Format abrégé deux lettres).	Non	String
timeOut	Indique le temps de timeOut pour une requête ajax.	Non	integer (1500 par défaut)
AffichageMaxPertinence	Indique le nombre d'items maximum pour créer l'affichage.	Non	booléen (true sans limite), Integer
maximum	Nombre d'items maximum affichés (Dans l'ordre).	Non	Integer
siteOrigin	Code du site d'origine.	Non	String

Paramètre	Utilité	Obligatoire	Type de valeur
userId	Le user ID du client courant.	Non	String
key	La clé applicative	Non	String.
sessionId	L'ID de la session du client courant.	Non	String
autofocus	Détermine si un autofocus sur le premier élément doit être effectué.	Non	Booléen

Exemple d'initialisation :

```
var initParam = {
  acp: false,
  search: true,
  serviceId: 7195,
  status: "rc",
  language : "fr",
  domain : "http://localhost/afs",
  timeOut: 100,
  AffichageMaxPertinence : true,
  maximum : 15,
}
```

Paramètres du web-service Acp

Ces paramètres ne sont pas utiles si vous n'utilisez pas ce web-service, vous pouvez alors le remplacer par {}, un objet vide. Les paramètres sont les mêmes que ceux de la documentation [AFS](#). Chaque paramètre admet plusieurs valeurs, ce sont donc tous des tableaux.

Exemple d'initialisation :

```
var initParamAcp={
  "afs:feedOrder": ["hitparade_fr"],
  "afs:feed":
["hitparade_fr","featured_products_2_fr","stores_fr","categories_fr", "brands_fr"],
  "afs:replies": [3],
}
```

Paramètres des web-services Search

Le web-service Search peut être utilisé dans deux cas : * En tant que première auto-complétion, le principal intérêt réside alors dans la récupération des facets de résultats. * En tant que seconde auto-complétion appliquée sur les résultats de la première lorsqu'ils sont survolés.

De même que pour le web-service ACP, on peut utiliser les mêmes paramètres que dans la documentation [AFS](#).

Cependant, d'autres paramètres sont ajoutés :

Paramètre	Utilité	Type de valeur
OptionSearch	Permet de récupérer des attributs en plus de ce qui est récupéré naturellement.	"price" : Récupérer le prix minimum, "store_urls" : Récupérer l'url et l'image du premier magasin.
XPATH	Permet de récupérer un élément spécifique en proposant un XPATH (Partant de la racine du produit). Il faut rajouter store comme préfixe de chaque élément	Objet qui associe à un nom d'attribut la valeur du XPATH.
Flows	Détermine l'ordre d'affichage entre les feeds et facettes.	"facets" : La position de la facette vis à vis de son feed, "feeds" : position du feeds vis à vis de sa facette, "AllFacets" : Position de l'ensemble des facettes vis à vis de l'ensembles des feeds. Notez que si le tableau est initialisé sans "feeds", aucun item ne sera affiché. Par défaut, les facettes ne sont pas affichées.
facets	Un tableau de facettes devant être récupérées	Tableau de String

Exemples d'initialisations :

```
var initParamSearchAcp={
  "afs:feed": ["Catalog"],
  "afs:output": ["json"],
  "afs:replies": [5],
  "afs:filter": ['store="2"'],
  OptionSearch: [
    "price",
    "store_urls",
  ],
  XPATH : {
    gender : "/store:product/store:audience/store:genders/store:gender",
  },
}

var initParamSearch={
  "afs:replies": [3],
  Flows : ["feeds","facets"],
  facets : ["format", "classification","brand" ],
  "afs:feed": ["Catalog"],
  "afs:facetDefault" : ["replies=1","maxDepth=1"],
  "afs:facet" : ["classification,replies=2"],
  XPATH : {
    gender : "/store:product/store:audience/store:genders/store:gender",
  },
  OptionSearch:[
    "price",
  ]
}
```

Format de retour des données

Les données sont retournées avec un format simplifié : un tableau d'objets. Chaque objet correspond à un item, celui-ci contient l'ensemble des attributs utiles.

Il y a quelques attributs spécifiques de l'API :

- `_feedId` correspond à l'id du feed porteur d'une facette.
- `_category` correspond au feed de l'item/facette.
- `_deep` correspond à la profondeur d'une facette dans son arbre de valeur.
- `_deepmax` correspond à la profondeur maximum de l'arbre de la facette.

Exemple de retour avec les paramétrages précédents :

```
[
  {
    "name":"Around the World in 80 Days","label":"Around the World in 80
Days","category":"Home &
Decor","currency":"EUR","PRICE_FINAL":"5","_category":"Catalog","uri":"557_fr","doc
Id":168
  },
  {
    "name":"Silver Aviator Sunglasses","label":"Silver Aviator
Sunglasses","category":"Accessories","currency":"EUR","PRICE_FINAL":"280","PRICE_CU
T":"295","gender":"Male","_category":"Catalog","uri":"337_fr","docId":90
  },
  {
    "name":"Love is an Eternal Lie by The Sleeping Tree","label":"Love is an
Eternal Lie by The Sleeping Tree","category":"Home &
Decor","currency":"EUR","PRICE_FINAL":"2","_category":"Catalog","uri":"561_fr","doc
Id":172
  },
  {
    "items":2,"key":"Downloadable","_feedId":"format","label":"Downloadable","name":"Do
wnloadable","_category":"format","_deep":0,"_deepmax":0
  },
  {
    "items":3,"key":"6","_feedId":"classification","label":"Accessories","name":"Access
ories","_category":"Catégorie","_deep":0,"_deepmax":1
  },
  {
    "items":1,"key":"6|21","_feedId":"classification","label":"Bags &
Luggage","name":"Bags & Luggage","_category":"Catégorie","_deep":1,"_deepmax":1
  },
  {
    "items":1,"key":"6|18","_feedId":"classification","label":"Eyewear","name":"Eyewear"
,"_category":"Catégorie","_deep":1,"_deepmax":1
  },
  {
    "items":8,"key":"7","_feedId":"classification","label":"Home &
```

```
Decor", "name": "Home & Decor", "_category": "Catégorie", "_deep": 0, "_deepmax": 1
    },
    {
        "items": 3, "key": "7|22", "_feedId": "classification", "label": "Books &
Music", "name": "Books & Music", "_category": "Catégorie", "_deep": 1, "_deepmax": 1
    },
    {
        "items": 3, "key": "7|25", "_feedId": "classification", "label": "Decorative
Accents", "name": "Decorative Accents", "_category": "Catégorie", "_deep": 1, "_deepmax": 1
    },
    {
        "items": 1, "key": "233", "_feedId": "brand", "label": "Madison", "name": "Madison", "_category": "Marque", "_deep": 0, "_deepmax": 0
    }
]
```

Affichage de l'auto-complétion

Choix des templates

On propose 3 templates pré-faits, chacun ayant sa spécificité.

- Le template 1 permet de regrouper les objets selon leur feed.
- Le template 2 permet en plus de montrer plusieurs attributs de chaque objets.
- Le template 3 implémente en plus l'image des objets.

Vous pouvez cependant coder votre propre template en utilisant la documentation de [jQuery-ui](#). Le format est par ailleurs standard et peut donc être utilisé facilement par le template de base de jquery-ui.

Pour que l'affichage de nos templates soit vraiment abouti, il est intéressant de rajouter des paramètres.

Personnaliser les noms de catégories

Il est intéressant pour chaque feed de personnaliser la recherche et l'affichage. Ainsi, à une expression régulière matchant des noms de feeds, on associe un paramètre :

Paramètre	Explication
name	String à afficher, par défaut c'est le nom du feed.
SearchFocus	Booléen déterminant si une recherche doit être effectuée en cas de survol d'un item du feed (searchacp doit aussi être activé).
filter	Booléen déterminant si la recherche doit être effectuée en filtrant sur le feed et sa valeur. Faux par défaut.
query	Booléen déterminant si le nom de l'item doit être passé en query. Vrai par défaut si le mode filter est désactivé.
label	Objet proposant la même chose que name, mais pour des langages spécifiques.

Exemple d'initialisation de la variable globale :

```
_name_element={
  "[Cc]atalog*" : {
    name : "Produits",
    SearchFocus : true,
    filter : false,
    query : true,
    label : {
      fr: "Produits français",
      de : "Produits_de",
    }
  },
  "[Hh]itparade*": {
    name : "Meilleurs résultats",
    SearchFocus : true
  },
  "[Cc]at[ée]gorie*" : {
    name : "Catégories",
    SearchFocus : true,
    filter : true,
    query : false
  },
  "featured*": {
    name : "Produits",
    SearchFocus : true
  },
  "[Pp]roper*": {
    name : "Propriétés",
    filter : true,
    query : false,
    SearchFocus : true
  },
  "brand*": {
    name : "Marques",
    SearchFocus : true,
    filter:true ,
    query:false
  },
  "[Pp]romote": {
    name : "",
    SearchFocus : false
  },
  "stores*":{
    name : "Magasins",
  },
  "[Aa]rticle*":{
    name : "Articles",
  },
  "Marque":{
    SearchFocus :true ,
    filter: true ,
  },
  "Couleur":{
    SearchFocus:true ,
  },
}
```

```

        filter: true
    },
    "format":{
        SearchFocus:true ,
        filter: true,
    },
}

```

Affichage des options

Par défaut, aucun attribut n'est affiché, sauf pour le template 1 qui permet l'affichage uniquement du label d'un item. Pour les autres templates, il faut régler l'affichage. De même que pour `name_element`, il y a plusieurs paramètres pour chaque nom d'attribut.

name	Explication
name	String à afficher, vous pouvez accéder à n'importe quelle valeur de l'item en utilisant <code>&NOMATTRIBUT</code> , le code HTML est supporté. Vous pouvez également pousser la personnalisation avec une fonction, celle-ci prend en paramètre l'item et retourne une String.
defaultvalue	Permet d'attribuer une valeur par défaut. Elle est utilisée uniquement lors de sa désignation par <code>&NOMATTRIBUT</code> .
label	Un objet qui associe un nom à un langage. Ce nom fonctionne comme l'attribut <code>name</code> .

Exemple de paramétrage :

```

_AfficheOptions={
  "name" : {
    name: "&name",
  },
  "stock" : {
    name : "Stocks : &stock",
    label: {
      fr : function(item){
        if(item.stock < 5)
          return "Plus que "+item.stock+" en stock !";
      }
      de : "Deutsch stock",
    }
  },
  "PRICE_FINAL" : {
    //name: "&PRICE_FINAL &currency <s>&PRICE_CUT &currency </s>",
    name : function(item){
      if(item.PRICE_CUT != undefined && item.PRICE_FINAL < item.PRICE_CUT){
        return item.PRICE_FINAL+" "+item.currency+" instead of
<s>"+item.PRICE_CUT+" "+item.currency+"</s>" ;
      }
      return item.PRICE_FINAL+" "+item.currency ;
    },
  },
  "title" : {
    name:"Titre : ",
  },
}

```



```

    },
    "currency" : {
        defaultvalue : "&currency",
    },
    "PRICE_CUT" : {
        defaultvalue : "&PRICE_FINAL &currency",
    },
}

```

La hiérarchie des facettes

Les facettes sont organisées hiérarchiquement, la présence d'une valeur implique celle de ses parentes. Nous permettons donc de choisir quels niveaux de la hiérarchie vous souhaitez afficher. Pour chaque type de facette, il faut choisir à partir de quel niveau on souhaite afficher la valeur, et à partir de quel niveau on ne souhaite plus l'afficher. Il y a donc deux paramètres :

- beginning, qui le donne le premier niveau d'affichage.
- end, qui donne le dernier niveau d'affichage.

On peut utiliser -1 et décrémenter pour compter à partir du dernier niveau (-2 correspond alors à l'avant dernier niveau).

Exemple de paramétrage :

```

_facet_hierarchy={
  "format" : {
    beginning : 0,
    end : 2,
  },
  "classification" : {
    beginning : -1,
    end : -1,
  },
  "brand" : {
    beginning : 0,
    end : -1,
  },
}

```

Si le type d'une facette n'est pas paramétré, elle sera affichée quelque soit son niveau.

Séparer en plusieurs colonnes

Le template 4 permet de séparer la zone d'affichage en plusieurs colonnes. On peut choisir les feeds que l'on insère dans chaque colonne. Chaque numéro correspond à l'ordre de la colonne. On associe à ce numéro les feeds que l'on souhaite insérer dedans.

Exemple de paramétrage :

```

Columns={
  2 : ["brand", 'format', "classification", "Marque"],
  1 : ["Catalog", "Categories", "Promote"],
}

```

Modifier le CSS

Notre auto-complétion utilise notamment le widget d'auto-complétion de jQuery. Pour modifier le CSS, il faut donc modifier des classes de jQuery-ui et des classes d'Antidot.

Classe	Explication	Templates utilisant la classe
ui-autocomplete-category	Le CSS de du nom de la catégorie.	Tous.
ui-autocomplete	Le CSS de l'auto-complète dans son ensemble.	Tous.
ui-state-focus	Le CSS des items qui sont focus.	Tous.
ui-menu-item	Le CSS des items du menu.	Tous.
Antidot-ItemSearchAcp	Le CSS des items lorsqu'une image est trouvée.	Template 3.
Antidot-Lilitem	CSS commun au bloc d'image et options d'un item.	Template 3.
Antidot-picture	CSS de l'image lorsqu'il y en a une. (Permet notamment de définir la taille).	Template 3-4.
Antidot-Lilimage	CSS content l'image classique.	Template 3-4.
Antidot-LiBanner	CSS spécifique contenant l'image Promote en mode bannière.	3-4.
Antidot-LiTexte	CSS commun à l'ensemble des attributs lorsqu'il y a une image.	Template 3-4.
Antidot-attributesli	CSS commun l'ensemble des attributs.	Template 2-3-4.
Antidot-ulAcp	CSS contenant tous les éléments.	Tous.

Configurer son proxy avec MAMP

L'utilisation de requête ajax cross-domain peut poser problème, il faut aller configurer un proxy.

Dans le fichier de configuration de votre serveur proxy, rajoutez ces deux lignes :

```

ProxyPass /afs Votrenomdedomaine
ProxyPassReverse /afs Votrenomdedomaine

```

Et changez votre nom de domaine dans vos paramètres en :

```

"domain" : "Votreadressedeproxy/afs"

```