



- Ensuite le **code précédant le template des numéros de page**,
- celui **des numéros de page si la page n'est pas celle en cours**, [nbpage] sera remplacé par le numéro de la page, le mot clé [link] sera remplacé par l'url de la page.
- puis le **template de numéro de page si nous sommes dans la page en cours** [nbpage] sera également remplacé par le numéro de la page en cours
- et enfin le **code qui ferme** la partie des numéros de page. Le mot clé [link] sera remplacé par l'url de la page.

**Important** : lors de la création du shortcode on peut préciser si on veut un résultat paginé, si oui le tag spécial sera remplacé par la pagination de la manière suivante : **code** page précédente, **code** précédant les numéros de page, répétition du code du **numéro de page** en fonction de si nous sommes dans la **page en cours** ou **pas**, **code** fermant la numérotation et enfin le **code** du lien suivant. Si on ne précise pas qu'on veut un résultat paginé, le tag spécial sera remplacé par du **vide**.

**Une seule liste paginée est sensée être affichée par page ! En mettre plusieurs c'est occasionner des bugs à coup sûr !**

Dans l'exemple donné ici nous avons le lien vers la page précédente (s'il y en a une), suivi d'une liste de liens vers les différentes pages, sauf pour la page en cours qui affiche simplement le numéro sans lien, puis le lien vers la page suivante (s'il y en a une).

Le code des pages peut être généré dans l'**entête** et le  **pied**, ce dernier est enveloppé dans un div dont l'id est "wp84apidae-pages" pour faciliter les manipulations javascript.

#### code de la liste

Si la liste est paginée celle-ci est enveloppée dans un div dont l'id est wp84apidae-list pour faciliter les manipulations javascript.

La donnée reçue de l'API Apidae est au format JSON, pour extraire ces données, le moteur de template utilise 3 type de tags basés sur l'implémentaion [JSONPath de Skyscanner](#). Pour vous aider vous avez dans ce module une liste de JSONPath et un onglet permettant de tester les JSONParth sur l'objet touristique de votre choix. Ces 3 types de "tags" sont entre double accolades:

- Préciser un JSONPath directement entre double accolade retournera le **premier élément trouvé** par celui-ci :  
ex. {{\$.nom.libelleFr}} retournera le nom français d'un objet touristique
- Les JSONPath peuvent retourner plusieurs réponses pour cela il est possible d'utiliser la syntaxe suivante :  
{{foreach||\$.illustrations.\*.nom.libelleFr||<p>[val] ([count])</p>}}  
Chaque élément est séparé par **des triples "|"** le mot clé **foreach**, le **jsonpath** et le **microtemplate html**.  
la valeur retournée par chaque itération du JSONPath remplacera le mot clé [val]. Un mot clé spécial [count] est disponible, c'est un simple compteur il affichera 1 pour la première valeur, deux pur la deuxième, etc...  
Cet exemple affiche pour chaque description d'illustration un paragraphe avec la description et entre parenthèses le compteur.  
ex (code généré):  
<p>Description de ma première image (1)</p>  
<p>Description de ma deuxième image (2)</p>

Parfois il est nécessaire d'évaluer plusieurs JSONPath en même temps, par exemple si on veut afficher toutes les images avec leur description:  
{{foreach||\$.illustrations.\*.nom.libelleFr;\$.illustrations.\*.traductionFichiers.\*.urlListe||}}  
Les différents JSONPath sont séparés par **deux ";"**. Attention les JSONPath doivent retourner le même nombre de résultats !! Dans le **microtemplate**, la valeur retournée par le premier JSONPath sera accessible via le mot clé [val0], le deuxième via [val1], le troisième [val2] etc...  
Dans le cas présent ce tag générera une image avec description dans la balise alt pour chaque illustration de l'objet touristique.

- On peut avoir besoin d'afficher un code précis pour une ou des valeurs d'un JSONPath, la syntaxe est similaire à celle du tag foreach, on utilise alors **forif**:

**ATTENTION**: si le principe est le même que foreach et que l'on peut préciser plusieurs JSONPath séparés par **deux ";"**, la valeur sera testée **uniquement sur le premier JSONPath**.

Ce tag s'utilise de la manière suivante:

```
{{forif||$.illustrations.*.type;$.illustrations.*.nom.libelleFr||IMAGE1%%<p>[val1] ([count]) : <b>[val0]</b></p>;else%%<p>default</p>}}
```

Dans la partie de **microtemplating**, on précise la valeur que le JSONPath est sensé avoir, puis une séparation avec %% puis le **microtemplate** associé.

Une valeur spéciale **else**, non obligatoire, permet d'afficher un template par défaut à chaque itération si aucune valeur n'est trouvée, si on ne précise pas de else, seules les valeurs spécifiées provoqueront l'affichage du microtemplate associé. Cela peut-être utile pour n'afficher que certains labels

**CONSEILS** : pour les **forif** utilisez pour les **valeurs à tester** des références ou des identifiants sinon le code risque de ne pas fonctionner correctement. Ayez une bonne connaissance de JSONPath cela évitera du code inutilement complexe ex :

```
{{foreach||$.illustrations[0].nom.libelleFr;$.illustrations[0].traductionFichiers[0].urlFiche||<p></p>}}
```

Ce code affichera la première image d'un objet touristique avec sa description en alt, le JSONPath ici est limité au premier résultat.

Un code simple est ce qui permettra de meilleures performances et évitera les erreurs.

#### LES TAGS NE PEUVENT PAS S'IMBRIQUER LES UNS DANS LES AUTRES !

Une liste non exhaustive de JSONPath est à votre disposition à la fin de cette documentation afin de vous faciliter la tâche.

Maintenant que nous avons vu comment fonctionnent la partie template nous allons pouvoir aborder comment générer un lien vers le détail d'une fiche. Pour cela nous partirons de l'exemple suivant que vous pourrez adapter :

```
<a href="/apiref/{tourl:$.localisation.adresse.commune.nom}"/>{tourl:$.nom.libelleFr}/{$.id}-{detailid}">...</a>
```

Notez que le lien commence par **/apiref** c'est le mot clé indispensable qui nous permet de savoir que nous allons dans une page de détail.

**/hotel** correspond au type d'objet, et vu que nous nous basons sur des sélections et qu'il est préférable qu'elles soient pour un même type d'objet le mot clé hotel est indiqué "en dur" il est possible de le déterminer de manière dynamique mais il vous faudra faire un forif sur les valeurs de \$.type qui correspond au type d'objet touristique.

```
{{tourl:$.localisation.adresse.commune.nom}}
```

la partie suivante est un jsonpath qui commence par le mot clé **tourl**: ce mot clé permet de transformer du texte pour qu'il soit utilisable dans une url, ensuite le jsonpath correspond au nom de la commune. Tout jsonpath précédé de **tourl**: sera transformé pour être "url compliant".

```
{{tourl:$.nom.libelleFr}}
```

même chose que précédemment on transforme ici le nom de l'objet touristique en français en version pour les url.

A noter il existe dans le même registre que tourl: le préfixe **nl2br**: qui transforme les retours chariots en sauts de ligne HTML.

`{{$.id}}` correspond à l'id Apidae, il permet de charger le bon objet touristique

`-[detailid]`

`[detailid]` est un mot clé qui sera remplacé par l'identifiant du template de détail, il est également essentiel car il conditionne les noeuds d'informations qui seront appelés et la langue.

Pour récapituler nous avons un lien du type:

`/api/type/objet/commune/nom de l'objet/identifiant apidae de l'objet-identifiant du template de détail`

Cette réécriture d'url est mise en place avec le plugin et elle est supprimée si vous le désinstallez.

## Les templates de détail

- Vous aurez à préciser un nom afin de le retrouver dans la partie fabrique à shortcode.
- Vu que le template de détail est lié aux informations retournées par l'API, vous pourrez préciser, de la même manière que pour les listes, les champs (ou noeuds d'information retournés) et choisir la langue.
- Si vous le désirez, au travers d'une configuration avancée et si vous avez bien [pris connaissance de la documentation technique](#) sur la récupération de la donnée pour un objet touristique, vous pouvez préciser les variables en get envoyée lors de la requête de récupération comme ceci : `responseFields=id,nom&locales=fr,en`

Dans cet exemple-ci le `responseFields` est inutile car précisé avant mais c'est pour montrer la logique. L'intérêt serait de pouvoir récupérer plusieurs langues au besoin, donc vous pourriez simplement préciser :

```
locales=fr,en
```

Si vous avez un site anglais et que le nom de l'objet n'est pas traduit vous pourriez alors afficher le nom de l'objet touristique soit en récupérant le nom anglais s'il existe sinon le français avec le tag : `{{$.nom['libelleEn','libelleFr']}}`

- Vous pourrez en utilisant les 3 types de tag précisés dans la partie Template de liste indiquer le contenu de la balise Title. Faites bien attention de ne pas mettre de balises HTML et de **rester sur du texte pur** sinon vous casserez le code de votre page !
- Vous pourrez ensuite préciser des fichiers css puis js à rajouter dans votre page, chaque fichier doit être séparé par des virgules.
- Pour le code du template, il est similaire à celui du template de liste avec les 3 types de tag (les JSONPath direct, les foreach et les forif). Il y en a un supplémentaire et spécifique pour le lien de retour. Si on vient d'une liste, la page de détail se souviendra de la page de liste.  
`{{returnlink}}<p><a href="[retlink]">retour à la liste</a></p>|||<p>pas de lien de retour</p>}}`  
Après le mot clé `returnlink` qui indique que nous sommes sur un lien de retour nous avons ensuite un microtemplate avec le mot clé `[retlink]` qui sera remplacé par l'url de provenance, et si la page a été accédée directement **sans passer par une liste, ce sera le 2e microtemplate qui s'affichera**. Si on ne veut pas de template si on ne vient pas d'une liste, on peut faire de la manière suivante : `{{returnlink}}<p><a href="[retlink]">retour à la liste</a></p>|||}}` et vu que le 2e microtemplate est vide, rien ne s'affichera.

## Moteurs

Comme pour les autres éléments on précisera un nom pour pouvoir le sélectionner dans la fabrique à shortcode.

Le template pour le moteur se compose de code HTML et de deux tags spécifiques, toujours sur la même base de deux accolades `{{...}}` avec les paramètres séparés par des triples pipes `|||`:

- `{{recap|||<a href="[rmlink]">supprimer le critère [label]</a>}}`  
Ce tag comme le mot clé `recap` l'indique, récapitule les critères sélectionnés et permet leur suppression au **travers du microtemplate** `[rmlink]` sera remplacé par un lien sans le critère et `[label]` par le libellé du critère. Ce microtemplate se répète pour chaque critère.
- `{{recherche|||classement|||2 étoiles|||2-etoiles|||"criteresQuery"."critere:HotellerieClassement_2749"|||<a href="[link]">2 étoiles</a>|||<a href="[link]">enlever le critère 2 étoiles</a>}}`  
Ce tag génère un lien vers la liste en prenant en compte un nouveau critère de recherche, ou s'il est déjà sélectionné peut générer un lien qui le désélectionne.
  - `recherche` est le mot clé qui indique que nous allons déclarer un critère de recherche
  - `le deuxième paramètre` précise la **famille** du critère recherche, il ne peut y avoir qu'un critère sélectionné par famille. Par exemple pour le classement d'un hôtel nous avons de 1 à 5 étoiles mais ces critères ne peuvent pas se cumuler car un hôtel ne peut pas être 2 et 3 étoiles en même temps. Donc le lien d'exemple est de la famille `classement` et si nous avons choisi les 2 étoiles, si nous cliquons sur 3 étoiles avec la même famille : `classement` ; le critère 2 étoiles sera remplacé par 3 étoiles. Il vous faut choisir un mot clé pour chaque famille de critère (donc des critères qui ne peuvent se cumuler entre eux). Préférez un mot clé en minuscule, sans espace, ni accent.
  - `Le troisième paramètre` est le label qui va s'afficher.
  - `Le quatrième paramètre` est la version du label pour l'url, **il doit être unique**. Vu que ce paramètre sera dans l'url quand le critère sera sélectionné il doit se composer uniquement des lettres minuscules non accentuées sans espace. On peut mettre des tirets 3-etoiles.
  - `Le cinquième paramètre` correspond au code complémentaire JSON pour filtrer la sélection de manière à obtenir ce que l'on souhaite. Par exemple ici pour filtrer notre sélection d'hôtel pour obtenir les 3 étoiles il faut ajouter à la requête de base `"criteresQuery"."critere:HotellerieClassement_2749"` pour filtrer sur une commune par exemple avignon il faudrait plutôt :

```
"communeCodesInsee": [ "84007" ]
```

**Attention**, Les éléments de `criteresQuery` se concatènent et les autres se remplacent, on peut préciser plusieurs critères séparés par des virgules dans la logique du format JSON. Pour plus d'information [veuillez regarder la documentation](#). Pour les `criteresQuery` les éléments de référence de votre compte Apidae vous seront indispensables (Diffuser > Outils développeurs > Référentiels > Éléments de référence). Par exemple ici pour les 3 étoiles :

Nom	Code TIF	Type	Famille	Parent	Ordre
3 étoiles	2750	06.04.01.03.03	HotellerieClassement		3

on voit que le code associé est 2750 et qu'il s'agit d'un type `HotellerieClassement` ce qui donne, pour filtrer sur les 3 étoiles, la requête que nous avons en exemple.

Si vous [regardez la documentation](#) et les **critères internes** dans le référentiel Apidae, vous pourrez également formuler le code nécessaire pour filtrer sur ceux-ci.

- `Le sixième paramètre` est le microtemplate de lien si le critère n'est pas sélectionné, le mot clé `[link]` sera remplacé par le lien vers la liste prenant en compte le critère.
- Enfin `le septième paramètre` est le microtemplate si le critère est sélectionné et le mot clé `[link]` pointe vers l'url **sans le critère**.

**Note** :Il est possible de faire un filtrage sur les **dates**, le filtrage se fera grâce aux variables d'url **datedebut** et **datefin** ce qui correspond [aux paramètres dateDebut et dateFin de la recherche via l'API](#).

Pour utiliser cette fonctionnalité vous pourrez, par exemple, dans le code de votre moteur ajouter deux champs en input text et grâce à un datepicker javascript de votre choix (et compatible avec les librairies js chargées par votre worpress) modifier l'url de la page de liste en ajoutant les variables d'url au format YYYY-mm-dd (ex. 2017-08-01 pour le 1er août 2017), comme pour l'API.

**Attention** : Prévoyez de supprimer **le numéro de page** s'il est présent ! Sinon la recherche essaiera de trouver la page demandée selon les nouveaux critères de recherche (avec les dates) et cette page n'existera peut-être pas, ce qui provoquera un dysfonctionnement !

ex. : <http://monsite.domain/fetes-et-manifestations/2/> >> (je selectionne des dates) >> <http://monsite.domain/fetes-et-manifestations/?datedebut=2017-08-01&datefin=2017-08-12>  
Le numéro de page a disparu et les variables d'url datedebut et datefin ont été rajoutées.

La fabrique à shortcode

Le principe du module est donc de générer un shortcode qui sera remplacé par une liste de résultat. Cet onglet vous permet de générer le shortcode en sélectionnant la liste, le template de liste le template de détail et éventuellement le moteur. La fabrique est inutilisable tant qu'il n'y a pas au moins un élément dans les 3 premiers et la fabrique à shortcode ne voudra pas vous retourner un code.

Vous pouvez cocher la case liste paginée mais **une seule liste paginée doit être insérée par page**. Pour info, lorsque la case paginée n'est pas cochée le mot clé [moteur] dans l'entête de liste est remplacé par du vide, de même pour le tag de pagination.

Il est possible de spécifier un nombre de réponses, par défaut il y en aura 20. Cette option est utile pour faire remonter, par exemple, un pavé de 3 offres aléatoires.

Pour plus de facilité on peut préciser des fichiers css et js supplémentaires selon les besoins.

Liste non exhaustive de JSONPath

Afin de vous aider à ne pas perdre trop de temps et à vous donner des bases de travail pour générer le ou les JSONPath dont vous avez besoin en voici une liste.

A noter : Les objets touristiques dans Apidae ont des noeuds d'informations spécifiques à leur type. Nous allons donc présenter d'abord des JSONPath généraux puis par type d'objet. Il y a souvent libelleFr dans les JSONPath ce qui correspond au libellé français, libelleEn est pour l'anglais, ... pour les moyens de contact (et d'autres noeuds d'informations) il peut y avoir aussi des informations destinée à une lange par exemple : \$.informations.moyensCommunication[?(@.type.id == 204)].coordonnees.fr correspond à l'email pour les français, comme pour les libellés, changer le code de la langue donnera accès à l'information localisée à condition bien sûr qu'elle soit saisie, retrouvez [plus d'infos ici sur les langues](#).

Dans la colonne description de cette liste, le mot référence correspond à l'identifiant Apidae du critère correspondant.

Vous pouvez partir de cette base pour **tester les JSONPath dans l'onglet correspondant du module** : chargez un objet touristique grâce à son identifiant et testez la valeur de retour de vos JSONPath

JSONPath généraux

JSONPath	Description
\$.id	identifiant Apidae
\$.type	type d'objet ( <a href="#">voir la documentation</a> )
\$.nom.libelleFr	nom en français
\$.nom["libelleEn",'libelleFr']	nom en anglais puis en français
\$.ouverture.periodeEnClair.libelleFr	périodes d'ouvertures en français
\$.illustrations.*.traductionFichiers.*.urlDiaporama	images en 900px de large
\$.illustrations.*.traductionFichiers.*.urlFiche	images en 210px de large
\$.illustrations.*.traductionFichiers.*.urlListe	images en 100px de large
\$.illustrations.*.nom.libelleFr	nom des images
\$.illustrations.*.copyright.libelleFr	copyright des images
\$.prestations.languesParlees.*.libelleFr	langues parlées
\$.prestations.services.*.libelleFr	services
\$.prestations.conforts.*.libelleFr	éléments de confort
\$.prestations.equipements.*.libelleFr	équipements
\$.*.activites.libelleFr	activités
\$.informations.structureGestion.nom.libelleFr	nom du propriétaire
\$.informations.structureGestion.civilite.libelleFr	civilité du propriétaire
\$.localisation..adresse.adresse1	adresse1
\$.localisation..adresse.adresse2	adresse2
\$.localisation..adresse.adresse3	adresse3
\$.localisation.adresse.commune.codePostal	code postal
\$.localisation.adresse.commune.nom	ville
\$.localisation.adresse.commune.code	code insee de la ville
\$.informations.moyensCommunication[?(@.type.id == 204)].coordonnees.fr	email (pour fr)
\$.informations.moyensCommunication[?(@.type.id == 205)].coordonnees.fr	site web
\$.informations.moyensCommunication[?(@.type.id ==	téléphone (tous)

201)].coordonnees.fr	
\$.informations.moyensCommunication[?(@.type.id == 201 and @.coordonnees.fr =~ /^(?!06 07).*\$/)].coordonnees.fr	téléphone (fixe)
\$.informations.moyensCommunication[?(@.type.id == 201 and @.coordonnees.fr =~ /^(?!06 ^07)/)].coordonnees.fr	téléphone (mobile)
\$.informations.moyensCommunication[?(@.type.id == 202)].coordonnees.fr	fax
\$.informations.moyensCommunication[?(@.type.id == 207)].coordonnees.fr	facebook
\$.descriptionTarif.modesPaiement.*.libelleFr	moyens de paiement libellés
\$.prestations.labelsTourismeHandicap.*.id	références des labels tourisme et handicap
\$.criteresInternes.*.id	références critères internes
\$.criteresInternes.*.libelle	label critères internes
\$.localisation.geolocalisation.geoJson.coordinates[1]	latitude
\$.localisation.geolocalisation.geoJson.coordinates[0]	longitude
\$.descriptionTarif.tarifsEnClair.libelleFr	texte des tarifs
\$.multimedias[?(@.traductionFichiers.*.extension=="kml" and @.traductionFichiers.*.locale=="fr")].traductionFichiers.*.url	url fichiers KML
\$.multimedias[?(@.traductionFichiers.*.extension=="kml" and @.traductionFichiers.*.locale=="fr")].nom.libelleFr	libellé des fichiers KML
\$.multimedias[?(@.type=="DOCUMENT")].traductionFichiers[?(@.locale=="fr")].url	url des documents liés
\$.multimedias[?(@.type=="DOCUMENT")].nom.libelleFr	nom des documents liés
\$.donneesPrivees[?(@.nomTechnique == 'identifiant du nom technique')].descriptif.libelleFr	libellé d'une donnée privée "identifiant du nom technique" est bien sûr à remplacer par l'identifiant que l'on trouve dans le référentiel Apidae
\$.prestations.nombrePersonnesMinimum	nombre minimum de personne pour une prestation
\$.prestations.nombrePersonnesMaximum	nombre maximum de personne pour une prestation
\$.prestations.ageMin	age minimum pour une prestation
\$.prestations.ageMax	age maximum pour une prestation
\$.presentation.descriptifCourt.libelleFr	descriptif court
\$.presentation.descriptifDetaille.libelleFr	descriptif détaillé

## JSONPath ACTIVITE

JSONPath	Description
\$.informationsActivite.activiteType.id	type d'activité référence
\$.informationsActivite.activiteType.libelleFr	type d'activité libellé
\$.informationsActivite.dureeSeance	durée de l'activité

## JSONPath COMMERCE\_ET\_SERVICE

JSONPath	Description
\$.informationsCommerceEtService.commerceEtServiceType.id	référence du type de commerce
\$.informationsCommerceEtService.commerceEtServiceType.libelleFr	libellé du type de commerce

## JSONPath DEGUSTATION

JSONPath	Description
\$.informationsDegustation.typesProduit.id	type de dégustation référence
\$.informationsDegustation.typesProduit.libelleFr	type de dégustation libellé
\$.informationsDegustation.chartreQualite.id	référence de la charte qualité
\$.informationsDegustation.chartreQualite.libelleFr	libellé de la charte qualité
\$.informationsDegustation.labelsChartesQualite.*.libelleFr	labels libellés
\$.informationsDegustation.labelsChartesQualite.*.id	références labels
\$.informationsDegustation.aopAocIgps.*.libelleFr	aop aoc libellés
\$.informationsDegustation.aopAocIgps.*.id	références aop aoc

\$.informationsDegustation.aocDescriptif.libelleFr	Aoc libellés
\$.informationsDegustation.aocDescriptif.id	Aoc références
\$.informationsDegustation.regionProduction.libelleFr	région de production libellé
\$.informationsDegustation.regionProduction.id	région de production référence
\$.informationsDegustation.typesProduit.*.libelleFr	types de produits libellés
\$.informationsDegustation.typesProduit.*.id	types de produits références

#### JSONPath EQUIPEMENT

JSONPath	Description
\$.informationsEquipement.activites.id	référence d'activité
\$.informationsEquipement.activites.libelleFr	libellé de l'activité
\$.informationsEquipement.rubrique.id	référence du type
\$.informationsEquipement.rubrique.libelleFr	libellé du type
\$.informationsEquipement.itineraire.distance	distance d'un itinéraire
\$.informationsEquipement.itineraire.denivellation	dénivellé d'un itinéraire
\$.informationsEquipement.itineraire.dureeJournaliere	durée d'un itinéraire

#### JSONPath FETE\_ET\_MANIFESTATION

JSONPath	Description
\$.informationsFeteEtManifestation.typesManifestation.id	référence du type de FMA
\$.informationsFeteEtManifestation.typesManifestation.libelleFr	libellé du type de FMA
\$.informationsFeteEtManifestation.categories.*.id	référence de la catégorie de FMA
\$.informationsFeteEtManifestation.categories.*.libelleFr	libellé de la catégorie de FMA
\$.informationsFeteEtManifestation.evenementGenerique.id	référence du sous type d'évènement (fête de la musique, etc.)
\$.informationsFeteEtManifestation.evenementGenerique.libelleFr	libellé du sous type d'évènement
\$.informationsFeteEtManifestation.portee.id	référence de la portée de la FMA
\$.informationsFeteEtManifestation.portee.libelleFr	libellé de la portée

#### JSONPath HEBERGEMENT\_COLLECTIF

JSONPath	Description
\$.informationsHebergementCollectif.hebergementCollectifType.id	référence du type d'Hébergement Collectif
\$.informationsHebergementCollectif.hebergementCollectifType.libelleFr	libellé du type d'Hébergement Collectif
\$.informationsHebergementCollectif.typesHabitation.*.id	référence du type d'habitation
\$.informationsHebergementCollectif.agrements.*.type.libelleFr	libellé du type d'agrément
\$.informationsHebergementCollectif.agrements.*.numero	numéro de l'agrément
\$.informationsHebergementCollectif.labels.*.id	référence du label
\$.informationsHebergementCollectif.labels.*.libelleFr	libellé du label
\$.informationsHebergementCollectif.chaineEtLabel.id	référence de la chaine
\$.informationsHebergementCollectif.chaineEtLabel.libelleFr	libellé de la chaine
\$.informationsHebergementCollectif.classementPrefectoral.id	référence du classement préfectoral
\$.informationsHebergementCollectif.classementPrefectoral.libelleFr	libellé du classement préfectoral

#### JSONPath HEBERGEMENT\_LOCATIF

JSONPath	Description
\$.informationsHebergementLocatif.hebergementLocatifType.id	référence du type d'Hébergement Locatif
\$.informationsHebergementLocatif.hebergementLocatifType.libelleFr	libellé du type d'Hébergement Locatif
\$.informationsHebergementLocatif.labels.*.libelleFr	libellé du label
\$.informationsHebergementLocatif.labels.*.id	référence du label
\$.informationsHebergementLocatif.classementPrefectoral.libelleFr	libellé du classement préfectoral
\$.informationsHebergementLocatif.classementPrefectoral.id	référence du classement préfectoral

JSONPath HOTELLERIE

JSONPath	Description
\$.informationsHotellerie.hotellerieType.id	référence du type d'hôtel
\$.informationsHotellerie.hotellerieType.libelleFr	libellé du type d'hôtel
\$.informationsHotellerie.classement.libelleFr	libellé du classement préfectoral
\$.informationsHotellerie.classement.id	référence du classement préfectoral
\$.informationsHotellerie.labels.*.id	référence du label
\$.informationsHotellerie.labels.*.libelleFr	libellé du label

JSONPath HOTELLERIE\_PLEIN\_AIR

JSONPath	Description
\$.informationsHotelleriePleinAir.hotelleriePleinAirType.id	référence du type de camping
\$.informationsHotelleriePleinAir.hotelleriePleinAirType.libelleFr	libellé du type de camping
\$.informationsHotellerie.classement.libelleFr	libellé du classement préfectoral
\$.informationsHotellerie.classement.id	référence du classement préfectoral
\$.informationsHotelleriePleinAir.labels.*.id	référence du label
\$.informationsHotelleriePleinAir.labels.*.libelleFr	libellé du label
\$.informationsHotelleriePleinAir.chaines.*.id	référence de la chaine
\$.informationsHotelleriePleinAir.chaines.*.libelleFr	libellé de la chaine

JSONPath PATRIMOINE\_CULTUREL

JSONPath	Description
\$.informationsPatrimoineCulturel.patrimoineCulturelType.id	référence du type de patrimoine
\$.informationsPatrimoineCulturel.patrimoineCulturelType.libelleFr	libellé du type de patrimoine
\$.informationsPatrimoineCulturel.categories.*.id	références des catégories
\$.informationsPatrimoineCulturel.categories.*.libelleFr	libellés des catégories
\$.informationsPatrimoineCulturel.themes.*.id	références des catégories
\$.informationsPatrimoineCulturel.themes.*.libelleFr	libellés des catégories

JSONPath PATRIMOINE\_NATUREL

JSONPath	Description
\$.informationsPatrimoineNaturel.patrimoineCulturelType.id	référence du type de patrimoine
\$.informationsPatrimoineNaturel.patrimoineCulturelType.libelleFr	libellé du type de patrimoine
\$.informationsPatrimoineNaturel.categories.*.id	références des catégories
\$.informationsPatrimoineNaturel.categories.*.libelleFr	libellés des catégories
\$.informationsPatrimoineNaturel.themes.*.id	références des catégories
\$.informationsPatrimoineNaturel.themes.*.libelleFr	libellés des catégories

JSONPath RESTAURATION

JSONPath	Description
\$.informationsRestauration.restaurationType.id	référence du type de restauration
\$.informationsRestauration.restaurationType.libelleFr	libellé du type de restauration
\$.informationsPatrimoineNaturel.categories.*.id	références des catégories
\$.informationsPatrimoineNaturel.categories.*.libelleFr	libellés des catégories
\$.informationsPatrimoineNaturel.labels.*.id	références des labels
\$.informationsPatrimoineNaturel.labels.*.libelleFr	libellés des labels
\$.informationsRestauration.classementsGuides.*.id	références des classements de guide
\$.informationsRestauration.classementsGuides.*.libelleFr	libellés des classements de guide

JSONPath SEJOUR\_PACKAGE

--	--

JSONPath	Description
\$.informationsSejourPackage.activiteCategories.id	référence de la catégorie d'activité
\$.informationsSejourPackage.activiteCategories.libelleFr	libellé de la catégorie d'activité
\$.informationsSejourPackage.activitesCulturelles.id	référence de l'activité culturelle
\$.informationsSejourPackage.activitesCulturelles.libelleFr	libellés de l'activité culturelle
\$.informationsSejourPackage.typesHebergement.id	référence du type d'hébergement
\$.informationsSejourPackage.typesHebergement.libelleFr	libellé du type d'hébergement
\$.informationsSejourPackage.formuleHebergement.id	référence de la formule d'hébergement
\$.informationsSejourPackage.formuleHebergement.libelleFr	libellé de la formule d'hébergement
\$.informationsSejourPackage.transports.*.id	références des transports
\$.informationsSejourPackage.transports.*.libelleFr	libellés des transports