Contenu Afficher

Objectif

L'objectif est de disposer d'une liste de valeurs des codes postaux/Communes dans un formulaire afin de contrôler la saisie du nom de la commune et vérifier la cohérence avec le code postal. On profite des améliorations importantes qui ont été apportées sur les listes de valeurs dans la version 19.1 et 19.2 d'Oracle APEX, avec notamment le support de plusieurs colonnes en réception.

Approches

La première approche triviale consiste à importer la liste des communes depuis un site officiel puis à la charger dans une table.

La deuxième approche a recours à un web service REST.
L'intérêt est d'éviter la maintenance supplémentaire d'un objet dans la database. Nous allons utiliser cette méthode.

Le site https://geo.api.gouv.fr/ fournit une API en open data qui répond au besoin. Le problème est que la liste des codes postaux pour une commune est fournie sous la forme d'un tableau (*nested array*). Or dans la version 19.2, un *web module* ne prend pas en compte les *nested array*. Il faudrait par exemple un proxy pour *aplatir* la structure json.

La poste, à travers son portail datanova, fournit également une API en open data. le résultat obtenu est exploitable avec APEX et c'est ce web service que nous allons utiliser dans la suite.

Voici un exemple d'appel avec, comme critère de recherche, les noms de

communes qui commencent par « toul »

```
https://datanova.laposte.fr/api/records/1.0/search/?dataset=laposte_hexasmal&q=toul
```

Pour information, il existe un autre site *https://vicopo.selfbuild.fr* qui fournit une API ne présentant pas l'inconvénient cité dans le site précédent. Sa forme est très simple: *https://vicopo.selfbuild.fr/cherche/{critere}*

Implémentation

C'est la version d'APEX 19.2 qui est utilisée dans ce qui suit. C'est à partir de celle-ci que l'on peut baser une LOV sur un web module.

Création d'un web module

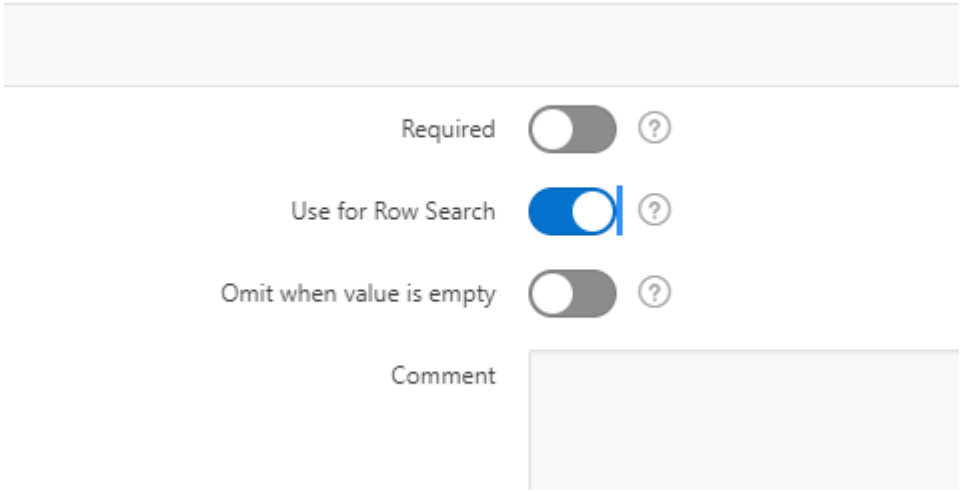
On crée dans Oracle APEX un web module basé sur le endpoint https://datanova.laposte.fr/api/records/1.0/search/ et on ajoute deux paramètrex de type *Query String variable* :

- dataset valeur fixe= laposte_hexasmal
- q valeur dynamique

Important: On précise au niveau du second paramètre (q) que sa valeur sera préférentiellement utilisée pour une recherche. (il s'agit d'une nouveauté dans la version 19.2): *Use for Row Search*

L'impact de ce réglage est que APEX remplira automatiquement ce paramètre

selon le contexte de la recherche: Si le web module est utilisé pour un report, alors ce qui est saisi dans la zone *Search* alimentera le paramètre q.
Si le web module est utilisé pour une LOV, alors ce qui est saisi dans le champs associé à cette LOV sera utilisé comme valeur de paramètre q.



Création d’une liste de valeurs multi-colonnes

On indique que la source de données de la nouvelle liste de valeurs est le web module créé précédemment.

A partir de la version 19.2, il est possible d’afficher plusieurs colonnes dans une popup LOV et on peut également injecter ces valeurs de colonnes dans plusieurs champs simultanément. Nous mettons à profit cette fonctionnalité pour alimenter à la fois le nom de la ville et la valeur du code postal.
Bien vérifier que les colonnes soient visibles dans le tableau des colonnes retournées par la LOV.

Prise en compte dans un formulaire

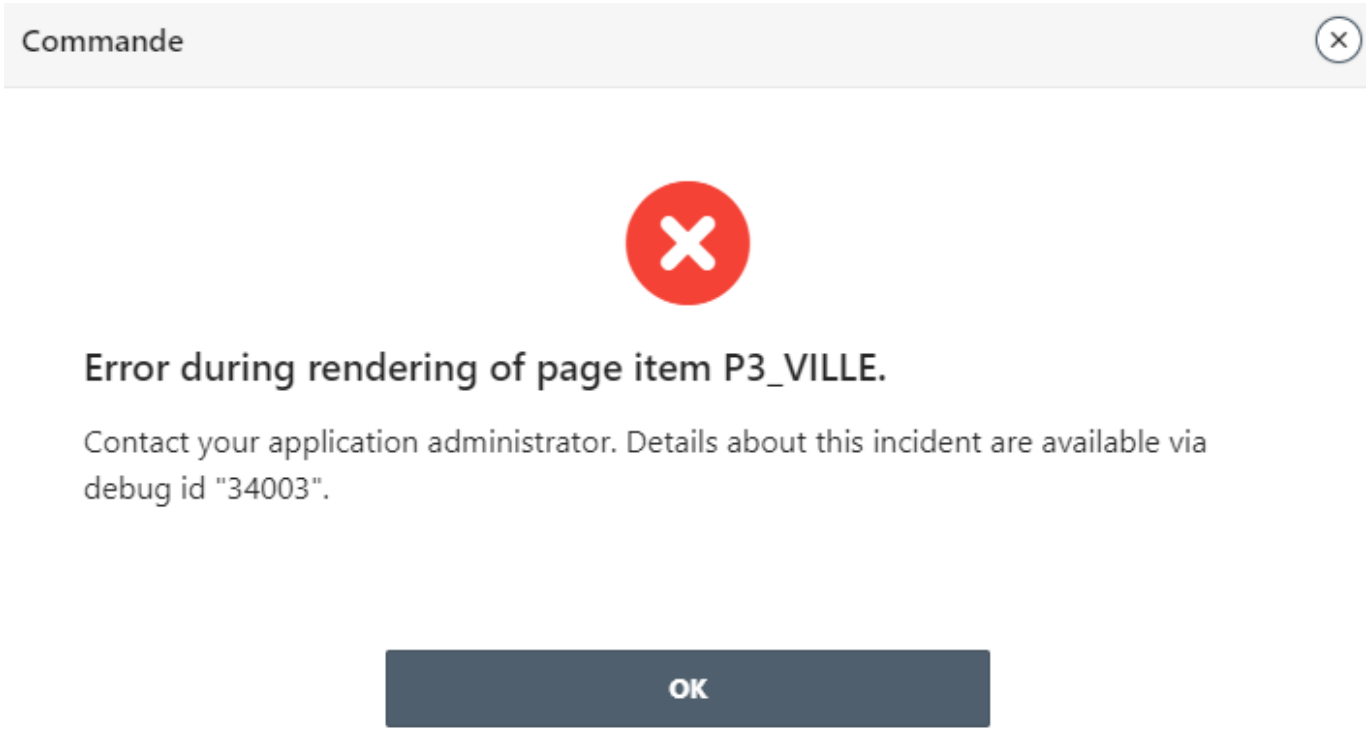
- indiquer que le champs VILLE est basé sur une Popup Lov
- Choisir la liste de valeur.
- Préciser le nombre de caractère minimum à entrer avant de déclencher une recherche dans la LOV, autrement dit, avant de déclencher un appel du Web Module.
- Designer les autres colonnes qui seront alimentées par la LOV.

L'alimentation du code postal se déclare en remplissant la propriété « champs supplémentaire » sous la forme de couples séparés par un point-virgule:

```
"colonne LOV: champs; colonne LOV: champs; ... "
```

Limitations

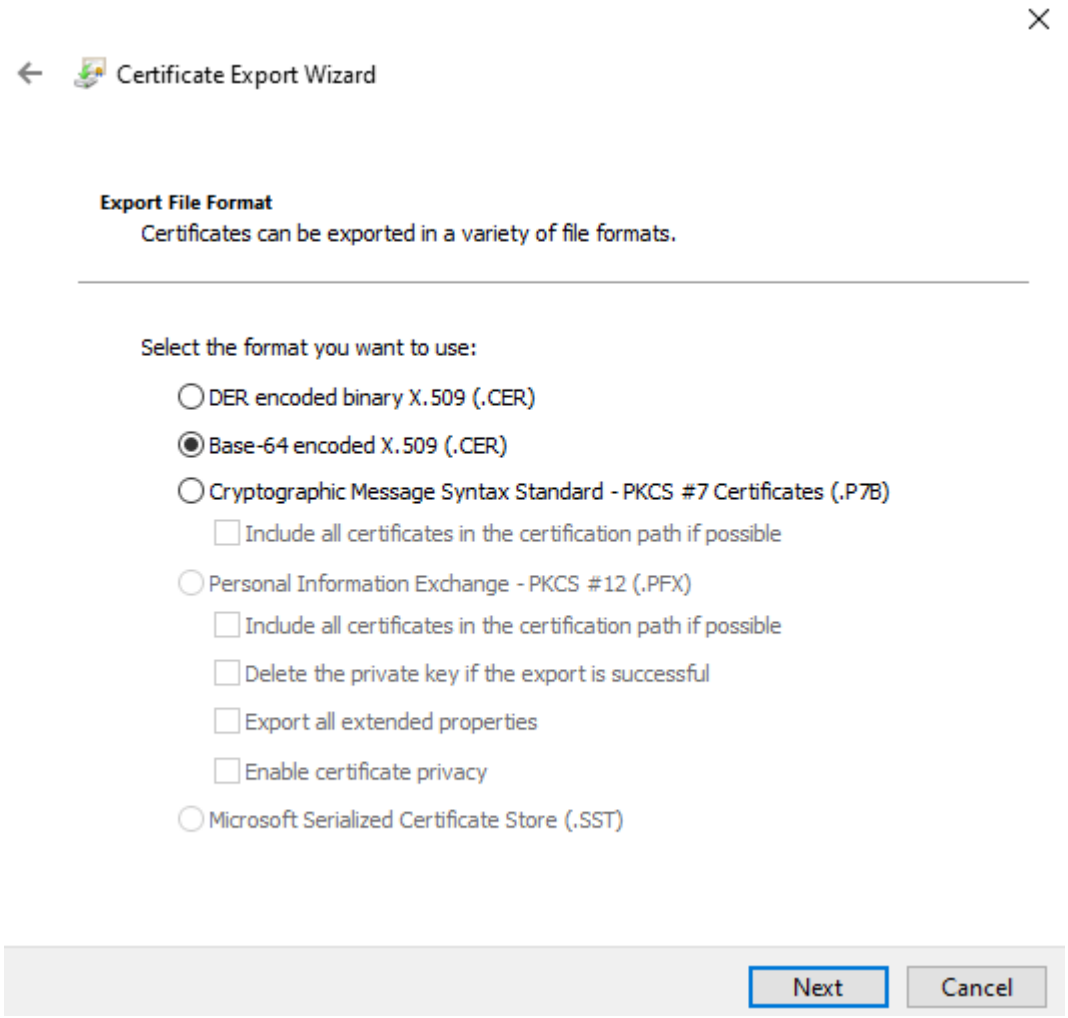
Si la LOV est de type *Popup Lov*, Il faut activer l'option *Entrée manuelle*. La conséquence que la popup se comportera comme un *combo box*. Sans cela, la lecture d'un enregistrement existant tombe en erreur. C'est dommage car cela oblige, d'un point de vue ergonomique, à cliquer sur la flèche vers le bas pour faire apparaître la liste de valeurs.



Remarque sur les certificats

Sur une instance APEX installée manuellement, il faut que le certificat du site appelé en REST soit enregistré dans le *wallet* de la database. Dans le cas contraire, on reçoit une erreur de ce type:

Remarque: Lors de la récupération du certificat, il a fallu que je choisisse la version du certificat encodé en Base64, sinon cela ne fonctionnait pas sur une database Oracle 18c express.



Annexes

Il est possible de récupérer le certificat directement sur le serveur avec une commande openssl, ce qui évite le recours à un navigateur. Cependant, je n'ai pas pu obtenir un format adapté au wallet Oracle. Peut-être s'agit il du format Base64 non respecté ?

```
openssl x509 -in cert.pem -text -noout
```

Author

- 

  Patrick

  GPM Factory