
Documentation Technique

Outils PHP

Accès aux données des véhicules à partir du Webservice

CATDATA

11 rue de la Haye

Espace Européen de l'Entreprise

67300 Schiltigheim

France

Août 2016

Sommaire

<u>Introduction</u>	p. 1
<u>Installation</u>	p. 2
<u>Architecture</u>	p. 2
Arborescence.....	p. 2
Descriptif des bases de données.....	p. 3
<u>Fonctionnement</u>	p. 3
Utilisation.....	p. 3
Contenu des fichiers.....	p. 7
Dans le sous – dossier ‘ajax’.....	p.7
Dans le sous – dossier ‘fct_conn’.....	p.9
Dans le dossier cible.....	p.29

Introduction

Outils PHP fournit d3es outils permettant d'accéder aux données du webService ou de AAA d'un ou plusieurs véhicules à partir d'un ou plusieurs identifiants.

Les bases de données utilisées sont françaises, portugaises et italiennes.

Les identifiants utilisables sont l'immatriculation (immat), le VIN (vehicle identification number), le Ktypnr (identifiant propre à l'entreprise) et le Typeld (identifiant du type de véhicule).

Les données sont affichées sous forme de tableau (horizontal ou vertical).

Les pages sont écrites en php. Elles contiennent également du code html et javascript (jquery et ajax). Seule la page de menu ne contient que du html.

Installation

Copier le dossier 'outils_php' sur le disque puis l'envoyer au serveur ainsi que tout son contenu.

Architecture

Arborescence

Le dossier 'outils_php' comporte les sous – dossiers suivants :

- outils_php
 - ajax
 - css
 - fct_conn
 - img
 - js
 - test

Le dossier source contient tous les fichiers .php avec les outils et le fichier 'menu.html', point de départ de l'outil.

Le sous – dossier 'ajax' contient les fichiers nécessaires aux appels ajax pour les outils 'immat' utilisant l'immatriculation. Ils permettent l'affichage d'un tableau contenant des données après appel au VinFilter.

Le sous – dossier 'css' contient le fichier 'tableaux.css' gérant la mise en forme des pages et l'affichage des tableaux de résultat.

Le sous – dossier ‘fct_conn’ contient les fichiers .php avec les éléments de connexion (identifiant, compagnie, mot de passe...), les variables globales et les fonctions appelées par les fichiers outils.

Le sous – dossier ‘img’ contient les images utilisées (logo, loader, fond d’écran).

Le sous – dossier ‘js’ contient le fichier jquery-3.1.0min permettant l’insertion et l’utilisation des fonctions jquery dans les fichiers outils.

Le sous – dossier ‘test’ contient tous les fichiers outils en version test (possibilité de faire des tests sans modifier directement les fichiers outils).

Descriptif des bases de données

Trois tables sont appelées dans la base de données cattools (les identifiants et les mots de passes sont dans le fichier ‘pdo.php’ dans le sous – dossier ‘fct_conn’ :

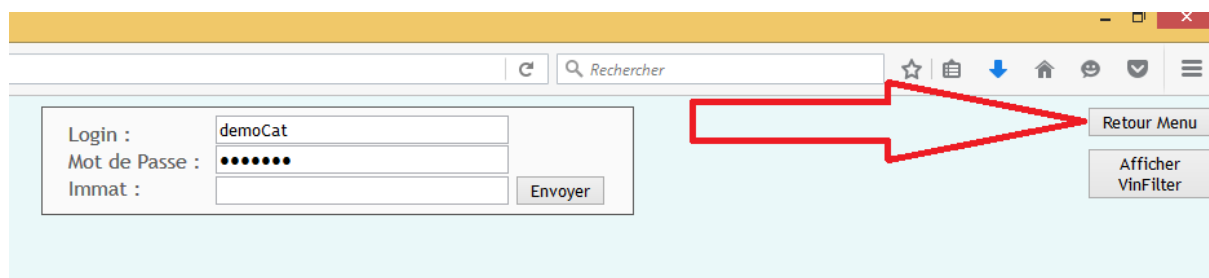
- w_veh_vl : base de données française sur les véhicules légers
- w_veh_pl vl : base de données française sur les poids lourds
- w_tec_vl_eu : base de données portugaise sur les véhicules légers

Le libellé et le nombre de champs diffèrent d’une table à l’autre.

Fonctionnement

Utilisation

L’accès au menu se fait à partir de l’url ‘http://cattools.catdata.com/outils_php/menu.html’. A partir de là il est possible de choisir un outil en cliquant dessus. Chaque page d’outil possède un bouton ‘Retour menu’ pour revenir à la page d’accueil.



Bouton Retour Menu

Les outils sont partagés en trois colonnes :



Page d'accueil

- Pour les outils du **SIV** : il faut saisir un login puis un mot de passe, ensuite un vin ou une immatriculation. Par défaut, un login et un mot de passe sont déjà saisis. Après validation, un tableau contenant les données provenant du Webservice s'affiche. Spécifiquement pour l'outil 'immat_vin_catdata_fr', un second tableau s'affiche parallèlement au premier avec les données AAA, puis un troisième tableau, toujours parallèle au premier, avec les données DATANEO, et enfin un quatrième tableau horizontal, sous les deux premiers, contenant les informations de la base de données à partir des ktypnr issus des services CATDATA et AAA.

Login :	<input type="text" value="demoCat"/>	<input type="button" value="Envoyer"/>
Mot de Passe :	<input type="password" value="••••••••"/>	
Immat :	<input type="text"/>	

Formulaire

Pour l'outil italien, le login n'est pas indispensable. Par défaut il est donc laisser vide.

Login :	<input type="text"/>	<input type="button" value="Envoyer"/>
Mot de Passe :	<input type="password" value="••••••••"/>	
Compagnie :	<input type="text" value="tecalliance"/>	
Immat :	<input type="text"/>	

Formulaire pour l'outil italien

Un bouton 'Afficher VinFilter' permet d'afficher ou de masquer un autre tableau avec les données issues du VinFilter.

Login :
Mot de Passe :
Compagnie :
Immat :

Envoyer

Retour Menu
Afficher VinFilter

Bouton VinFilter

Immat VinFilter fr

cattools.catdata.com/outils_php/immat_vin_catdata_fr.php

Rechercher

CATDATA

SIV IMMAT / VIN CATDATA FR

Login : demoCat
Mot de Passe :
Immat : 564BAF67
Envoyer

SIV CATDATA

Libellé	Valeur
CARROSSERIE	BERLINE
CARROSSERIE_CG	DERIV VP
CO2	120
CODE_MOTEUR	K9KM7
CODIF_VIN_PR	VF1CR1G0H38830152
CONS_EXURB	4.1
CONS_MIXTE	4.5
CONS_URB	5.3
COULEUR_VEHIC	BLANC
CYLINDREE	1461
DATE_1ER_CIR	20071212
DATE_DCG	20071212
DEPOLLUTION	NON
EMPAT	258
ENERGIE	GAZOLE
GENRE_V	CAM
GENRE_V_CG	CTTE

SIV AAA

Libellé	Valeur
carrosserie	BERLINE
carrosserieCG	DERIV VP
co2	120
codeMoteur	K9KM7
codifVin	VF1CR1G0H38830152
consExurb	4.1
consMixte	4.5
consUrb	5.3
couleurVehic	BLANC
cylindree	1461
date1erCir	jour ---12 mois --12-- annee 2007
dateDCG	jour ---12 mois --12-- annee 2007
depollution	NON
empat	258
energie	GAZOLE
genreV	CAM
genreVCG	CTTE

SIV DATA NEO

Libellé	Valeur
NB_PL_ASSISES_SIV	2
NRJ_SIV	GO
MASSE_F2_SIV	1645
PUISS_FISC_SIV	6
PUISS_KW_SIV	50
CNIT_SIV	
TVV_SIV	URE5021E7572
MOD_SIV	CLIOSOCIETE
CYLINDREE_SIV	1461
DT_PREM_IMMAT_SIV	200712
GENRE_DATANEO	VU
GENRE_SIV	CTTE
VIN_SIV	VF1CR1G0H38830152
IMMAT_SIV	564BAF67
CARROSS_SIV	DERIV_VP
CO2_SIV	0
MRO_SIV	RFNAUIT

Exemple 1 d'utilisation de l'outil immat

Immat VinFilter fr

cattools.catdata.com/outils_php/immat_vin_catdata_fr.php#div/immatVIN

Rechercher

Masquer VinFilter

TYPE_VVP	
VERSION	STE
HerNr_TD	93
KModNr_TD	5462
TYPVEH	T120 19051

turboCompr	TURBO
type	URE5021E7572
typeVarVersPrf	
typeVinCG	CR1G0H
version	STE
pneus	dimension 185/60 R15 84 H 195/50 R16 88 V
k_type	19051

Données Véhicules légers

N°	WS	Ktypnr	Type Id	hernr	kmodnr	Marque	Modèle	Version	Début construction	Fin construction	kw	ps	cyl_impot	cyl	nb_cyl	Libcar	Libene	Codes Moteur	libenemix	libprop
1	CATDATA	19051	38187	93	5462	RENAULT	CLIO III (BR0/1, CR0/1)	1.5 dCi (C/BR0G, C/BR1G)	200506	0	50	68	0	1461	4	3/5 portes	Diesel	K9K 714, K9K 768	Injection directe	Traction avant
1	AAA	19051	38187	93	5462	RENAULT	CLIO III (BR0/1, CR0/1)	1.5 dCi (C/BR0G, C/BR1G)	200506	0	50	68	0	1461	4	3/5 portes	Diesel	K9K 714, K9K 768	Injection directe	Traction avant

SIV VinFilter Véhicule léger (BDD France)

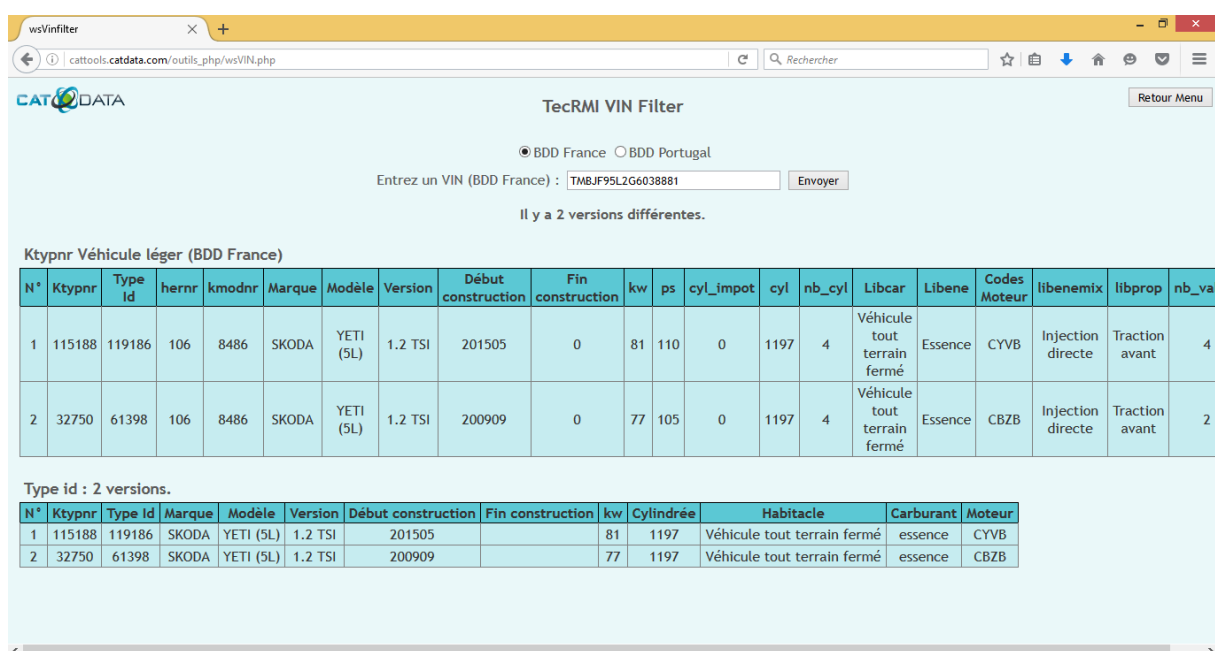
N°	Ktypnr	Type Id	hernr	kmodnr	Marque	Modèle	Version	Début construction	Fin construction	kw	ps	cyl_impot	cyl	nb_cyl	Libcar	Libene	Codes Moteur	libenemix	libprop	nb_
1	19051	38187	93	5462	RENAULT	CLIO III (BR0/1, CR0/1)	1.5 dCi (C/BR0G, C/BR1G)	200506	0	50	68	0	1461	4	3/5 portes	Diesel	K9K 714, K9K 768	Injection directe	Traction avant	nb_

Exemple 2 d'utilisation de l'outil immat

- Pour les outils **TecRMI** : l'outil VinFilter permet la saisie d'un VIN et de préciser la base de données (française ou portugaise). Le tableau du bas affichera les données du véhicule. Attention, le résultat peut afficher soit un ou des véhicules, soit des modèles, soit des marques. Les véhicules sont triés par ktypnr décroissant. Le tableau du haut n'affiche de résultats que s'il y a un ou plusieurs véhicules car il appelle la base de données.

L'outil 'typeid->ktypnr' demande la saisie d'un type id puis affiche les données du véhicule dans deux tableaux : l'un à partir du type id, l'autre à partir du ktypnr récupéré grâce au type id et qui permet l'appel à la base de données.

L'outil 'ktypnr->typeid' fait l'inverse. Il est de plus possible de préciser la base de données (française ou portugaise).



TecRMI VIN Filter

● BDD France ○ BDD Portugal

Entrez un VIN (BDD France) :

Il y a 2 versions différentes.

Ktypnr Véhicule léger (BDD France)

N°	Ktypnr	Type id	hernr	kmodnr	Marque	Modèle	Version	Début construction	Fin construction	kw	ps	cyl_impot	cyl	nb_cyl	Libcar	Libene	Codes Moteur	libenemix	libprop	nb_val
1	115188	119186	106	8486	SKODA	YETI (5L)	1.2 TSI	201505	0	81	110	0	1197	4	Véhicule tout terrain fermé	Essence	CYVB	Injection directe	Traction avant	4
2	32750	61398	106	8486	SKODA	YETI (5L)	1.2 TSI	200909	0	77	105	0	1197	4	Véhicule tout terrain fermé	Essence	CBZB	Injection directe	Traction avant	2

Type id : 2 versions.

N°	Ktypnr	Type id	Marque	Modèle	Version	Début construction	Fin construction	kw	Cylindrée	Habitacle	Carburant	Moteur
1	115188	119186	SKODA	YETI (5L)	1.2 TSI	201505		81	1197	Véhicule tout terrain fermé	essence	CYVB
2	32750	61398	SKODA	YETI (5L)	1.2 TSI	200909		77	1197	Véhicule tout terrain fermé	essence	CBZB

Exemple d'utilisation du VinFilter

- Pour les outils **TecDoc** : les outils ktynr/ntypnr FR et ktypnr PT permettent l'affichage des données dans la base de données française ou portugaise à partir d'un ou plusieurs ktypnr. Une saisie multiple est ici possible. La version française permet la distinction entre véhicule léger et poids lourd.

Informations du Ktypnr ou du ...

cattools.catdata.com/outils_php/ktypnrFR.php

TecDoc Ktypnr / Ntypnr FR

☒ ktypnr (véhicule léger)
 ☐ ntypnr (poids lourd)

Entrez un ktypnr : Envoyer

Ktypnr Véhicule léger (BDD France)

N°	Ktypnr	Type Id	hernr	kmodnr	Marque	Modèle	Version	Début construction	Fin construction	kw	ps	cyl_impot	cyl	nb_cyl	Libcar	Libene	Codes Moteur	lib
1	11786	70289	5	9730	AUDI	A1 Sportback (8XA, 8XF, 8XK)	1.6 TDI	201111	0	66	90	0	1598	4	3/5 portes	Diesel	CAYB	Inject
2	9978	68613	138	9589	CHEVROLET	CAMARO Décapotable	6.2	201101	0	298	405	0	6162	8	Décapotable	Essence	L99	Inject co d'admissic
3	458	401	74	18	MERCEDES-BENZ	A trois volumes (W124)	300 D (124.130)	199006	199306	81	110	2975	2996	6	A trois volumes	Diesel	OM 603.912	Moteur in
4	236	205	2	291	ALFA ROMEO	33 (905)	1.5 4x4 (905.AZM, 905.AZT)	198609	198912	77	105	1474	1490	4	3/5 portes	Essence	AR 30146, AR 30588	Inject co d'admissic

Exemple d'utilisation de l'outil Ktypnr

Contenu des fichiers

- Dans le sous – dossier ‘ajax’ :

Chaque fichier outil fait appel aux fichiers suivants :

- config.php (pour les identifiants, les variables globales, les url pour l'appel au Webservice...)
- pdo.php (pour l'appel à la base de données)
- ws_vinfilter.php (pour l'appel des fonctions du Webservice)
- fonctions.php (pour l'appel des fonctions spécifiques)

- **immat_vinfilter_fr**

Suite au clic sur le bouton ‘Afficher VinFilter’, les données utiles sont récupérées (immat, login, mot de passe).

Ex :

// récupération de l'immat

```

if(empty($_POST['recupImmat'])) {
    $immat = _IMMAT_DEPART;
} else {
    $immat = $_POST['recupImmat'];
}

```


Ensuite on récupère les informations du webservice pour chercher le typeld et le ktypnr. On s'appuie sur la fonction getDonneesWS pour extraire le vin puis les fonctions appelwsVINmodif pour le typeld et appelwsKtypnrFromTypeld pour le ktypnr.

Ex :

```
$resWS = getDonneesWS($login, $motDePasse, $immat);
$resVIN = appelwsVINmodif($vin, _COMPANY, $pays, $langue, _PASSWORD, _USER);

$tabKtypnr[] = array(
    "Ktypnr" => appelwsKtypnrFromTypeld($element->Typeld, _COMPANY,
    _PASSWORD, _USER)->GetVtTcdTypeAndClassByAdcTypeldResult->TcdTypeld
);
```

Le typeld et le ktypnr sont stockés dans des arrays pour envoi en paramètres dans les fonctions d'affichage. Chaque array contient un array qui contient les informations. On obtient alors un array par ktypnr ou typeld.

On teste ensuite le contenu de l'immat. S'il est vide on n'affiche rien. Sinon on cherche le classld avec la fonction appelwsKtypnrFromTypeld pour ensuite appeler la bonne fonction d'affichage. S'il n'y a pas de typeld on affiche un message d'erreur.

```
$classld = appelwsKtypnrFromTypeld($tabTypeld[0]['Type Id'], _COMPANY, _PASSWORD,
_USER)->GetVtTcdTypeAndClassByAdcTypeldResult->TcdClassld;
```

```
// affiche l'en – tête du tableau en fonction du pays et du classld
afficheEnTeteHorizPaysClassld($pays, $classld);
```

```
// affiche les données
afficheDonneesKtypnr($tabTypeld, $classld, $tabKtypnr, $conn, $pays);
```

- [immat_vinfilter_pt](#)

Même contenu que précédemment sauf qu'on utilise la base de données portugaise. Les données sont récupérées à partir de la fonction wslImmatPortugal :

```
$resWS = wslImmatPortugal($login, $motDePasse, $immat);
```

Pour l'affichage le classld est fixé à 1 car il n'y a pas de poids lourds.

- [immat_vinfilter_it](#)

Même contenu que précédemment. Il faut récupérer en plus la compagnie :

```
// récupération de la compagnie
```

```
if(empty($_POST['recupCompagnie'])) {  
    $company = _COMPANY_IT;  
} else {  
    $company = $_POST['recupCompagnie'];  
}
```

On utilise le webservice italien avec la fonction GetVehicleDataIT mais la base de données française pour le vinfilter :

```
$vin = GetVehicleDataIT($immat, $company, $motDePasse, _PRODUCT_IT, $login)-  
>GetRTCompletoDaTargaMinTecDocResult->Telaio;
```

Là aussi le classId est fixé à 1.

- Dans le sous – dossier ‘fct_conn’ :

- config.php

Ce fichier contient sous forme de define :

- les url pour les appels aux webservices ou aux services AAA et DATANEO
- les éléments de connexion aux webservices (login, mot de passe, compagnie, producteur pour l’outil italien), à AAA et à DATANEO (siret du client, son login et son mot de passe)
- les variables initialement définies (langue, pays, VIN, typeld, typnr, classId, immat)

- fonctions.php

Ici se trouve les fonctions d’affichage des données, les fonctions spécifiques d’appel aux webservices et des fonctions techniques.

```
/* Fonction d'affichage d'erreur et de suivi */
```

```
function print_t($input = "") {  
    echo "<pre>";  
    print_r($input);  
    echo "</pre>";  
}
```

```
// fonction wsVin modifiée pour renvoyer tous les modèles ou toutes les marques
```

```
function wsVINmodif($parametres=array())
```

Il s'agit d'une fonction modifiée de la fonction wsVIN. Contrairement à cette dernière, si l'appel s'arrête aux marques ou aux modèles, on renvoie tous les modèles ou toutes les marques trouvées.

Les paramètres sont identiques à la fonction wsVIN.

// fonction qui appelle wsVINmodif

```
function appelwsVINmodif($vin, $company, $pays, $langue, $password, $user) {  
    return wsVINmodif(array(  
        "vin" => $vin,  
        "vt_company" => $company,  
        "vt_country" => $pays,  
        "vt_language" => $langue,  
        "vt_password" => $password,  
        "vt_user" => $user,  
        "vt_cars" => 1,  
        "vt_trucks" => 1  
    ));  
}
```

Fonction qui prend en paramètres les données nécessaires pour l'appel de la fonction wsVINmodif et retourne le résultat de cette fonction.

// fonction qui appelle wsTypeld

```
function appelwsTypeld($company, $user, $password, $typnr, $classId)
```

Même principe que la fonction précédente mais spécifique à la fonction wsTypeld.

// fonction qui appelle wsVehiculeName

```
function appelwsVehiculeName($company, $password, $user, $typeld)
```

Même principe que la fonction précédente mais spécifique à la fonction wsVehiculeName.

// fonction qui appelle wsKtypnrFromTypeld

```
function appelwsKtypnrFromTypeld($typeld, $company, $password, $user)
```

Même principe que la fonction précédente mais spécifique à la fonction wsKtypnrFromTypeld.

// fonction qui appelle wsGetVtMakeList

```
function appelwsGetVtMakeList($vin, $company, $pays, $langue, $password, $user)
```

Même principe que la fonction précédente mais spécifique à la fonction wsGetVtMakeList.

```
// fonction qui appelle wsGetVtRangeList
```

```
function appelwsGetVtRangeList($vin, $company, $pays, $langue, $makeld, $password, $user)
```

Même principe que la fonction précédente mais spécifique à la fonction wsGetVtRangeList.

```
// fonction qui appelle wsGetVtTypeList
```

```
function appelwsGetVtTypeList($vin, $company, $pays, $langue, $rangeld, $password, $user)
```

Même principe que la fonction précédente mais spécifique à la fonction wsGetVtTypeList.

```
// procédure pour l'affichage des données à partir d'un tableau sur les VtTypeAdditionalDetail
```

```
function afficheDonneesTab($tab) {  
    foreach($tab as $element1){  
        $intitule = "";  
        switch(strval($element1->AddInfoKeyId)){  
            case '3' :  
                $intitule = 'Début construction';  
                break;  
            case '4' :  
                $intitule = 'Fin construction';  
                break;  
            case '5' :  
                $intitule = 'kw';  
                break;  
            case '6' :  
                $intitule = 'Cylindrée';  
                break;  
            case '7' :  
                $intitule = 'Habitacle';  
                break;  
            case '8' : // pour les camions  
                $intitule = 'kg';  
                break;  
        }  
    }  
}
```

```

case '9' : // pour les camions
    $intitule = 'Config Essieux';
    break;
case '10' :
    $intitule = 'Carburant';
    break;
case '25' : // pour les camions
    $intitule = 'Id Fabricant';
    break;
case '-4' :
    $intitule = 'Moteur';
    break;
case '-5' :
    $intitule = 'Boîte vitesse';
    break;
case '-6' : // pour les camions
    $intitule = 'Essieux';
    break;
default:
    break;
}
?><tr><td><?php
    print_r($intitule);
?></td><td><?php
    if (!isset($element1->AddInfoKeyValue)) {
        print_r(""); // car erreur sur les dates de fin non communiquées
(élément inexistant)
    } else {
        print_r($element1->AddInfoKeyValue);
    }
?></td></tr><?php
}
}

```

// procédure pour l'affichage des données à partir d'une ligne de requête ou d'un tableau

```
function AfficheDonnees($ligne) {
    foreach($ligne as $cle => $valeur) {
        ?><tr><td><?php
            print_r($cle);
        ?></td><td><?php
            print_r($valeur);
        ?></td></tr><?php
    }
}
```

// procédure pour l'affichage des données à partir d'une ligne de requête ou d'un tableau

// spécifique à l'outil immat car test sur la présence d'un ou plusieurs ktypnr et gestion de cas spécifiques

```
function afficheDonneesImmat($ligne)
```

// procédure pour l'affichage des données à partir d'une ligne de requête ou d'un tableau

// spécifique à l'outil immat it car gère les différents cas et s'arrête à l'objet Allestitenti pour affichage de tableaux complémentaires

// input : object

```
function afficheDonneesImmatIt($ligne)
```

// procédure qui affiche l'en - tête des tableaux

```
function afficheEnTete() {
    ?><tr>
        <th>Libellé</th>
        <th>Valeur</th>
    </tr><?php
}
```

// procédure qui affiche l'en - tête des tableaux

// spécifique à l'outil immat car plusieurs colonnes dont 6 fusionnées

```
function afficheEnTeteImmat() {
    ?><tr>
        <th>Libellé</th>
```

```

        <th colspan=6>Valeur</th>
    </tr><?php
}

// procédure qui affiche l'en - tête des tableaux VIN horizontalement
function afficheEnTeteHoriz() {
    ?><tr>
        <th>N°</th>
        <th>Ktypnr</th>
        <th>Type Id</th>
        <th>Marque</th>
        <th>Modèle</th>
        <th>Version</th>
        <th>Début construction</th>
        <th>Fin construction</th>
        <th>kw</th>
        <th>Cylindrée</th>
        <th>Habitacle</th>
        <th>Carburant</th>
        <th>Moteur</th>
    </tr><?php
}

/* Procédure afficheEnTeteHorizPaysClassId qui affiche l'en - tête du tableau selon le pays et
le classId
@param $pays
@param $classId
@param $siv permet l'affichage ou non d'une colonne supplémentaire spécifique à l'outil siv
immat fr
*/
function afficheEnTeteHorizPaysClassId($pays, $classId, $siv = "")

// procédure qui affiche les données horizontalement à partir d'une ligne de requête ou d'un
tableau
function afficheDonneesHoriz($ligne) {

```

```
foreach($ligne as $cle => $valeur) {
    ?><td><?php
    print_r($valeur);
    ?></td><?php
}
}
```

// procédure qui affiche les clés des données horizontalement à partir d'une ligne de requête ou d'un tableau

```
function afficheClesDonneesHoriz($ligne)
```

// procédure qui affiche les clés et les valeurs horizontalement d'un tableau à partir d'une ligne de requête ou d'un tableau

```
function afficheTabHoriz($tab)
```

```
/*
```

Fonction qui remplit un tableau avec les données VtTypeAdditionalDetail

Si plusieurs infos concernent le même indice, les infos sont concaténées dans le même indice

@param \$tab de type array VtTypeAdditionalDetails

@return un array

```
*/
```

```
function ajoutDetail($tab) {
```

```
    $tabRempli = array();
```

```
    foreach($tab as $element) {
```

```
        switch(strval($element->AddInfoKeyId)){
```

```
            case '3' :
```

```
                $tabRempli['Début construction'] = $element->AddInfoKeyValue;
```

```
                break;
```

```
            case '4' :
```

```
                if (!isset($element->AddInfoKeyValue)) {
```

non communiquées (élément inexistant)

```
                } else {
```

```
                    $tabRempli['Fin construction'] = $element->AddInfoKeyValue;
```



```

    }
    break;
case '5' :
    $tabRempli['kw'] = $element->AddInfoKeyValue;
    break;
case '6' :
    $tabRempli['Cylindrée'] = $element->AddInfoKeyValue;
    break;
case '7' :
    if (!isset($element->AddInfoKeyValue)) {
        $tabRempli['Habitacle'] = ""; // car erreur si infos non
communiquées (élément inexistant)
    } else {
        $tabRempli['Habitacle'] = $element->AddInfoKeyValue;
    }
    break;
case '10' :
    if (!isset($element->AddInfoKeyValue)) {
        $tabRempli['Carburant'] = ""; // car erreur si infos non
communiquées (élément inexistant)
    } else {
        $tabRempli['Carburant'] = $element->AddInfoKeyValue;
    }
    break;
case '-4' :
    if(!isset($element->AddInfoKeyValue)) {
        $tabRempli['Moteur'] = "";
    } else {
        if(empty($tabRempli['Moteur'])) {
            $tabRempli['Moteur'] = $element-
>AddInfoKeyValue;
        } else {
            $tabRempli['Moteur'] .= '<br>'.($element-
>AddInfoKeyValue);
        }
    }

```

```

    }
    break;
default:
    break;
}
}
// vérification de l'existence de l'objet moteur (addinfokeyid = -4)
$present = false;
foreach($stab as $element) {
    $vars = get_object_vars($element);
    $i = 0;
    while(!$present && $i < sizeof($vars)) {
        if(in_array('-4', $vars)) {
            $present = true;
        }
        $i++;
    }
}
if(!$present) {
    // cas où l'élément moteur manque
    $stabRempli['Moteur'] = "";
}
return $stabRempli;
}

```

/* procédure afficheTabVide qui affiche un tableau vide de même longueur que le nombre de champs de la requête

@param \$tab1 qui contient le ktypnr et le typeld

@param \$tab2 qui contient les champs de la requête

*/

```
function afficheTabVide($tab1, $tab2) {
```

// modification de \$tab2 pour obtenir des clés correspondants aux champs et des valeurs vides

```
    $tab2Vide = array_fill_keys($tab2, "");
```

```
    afficheDonneesHoriz($tab1);
```

```
afficheDonneesHoriz($tab2Vide);
}

/* procédure afficheDonneesTypnr qui affiche les résultats d'une requête à partir du typnr et
du classId, selon le pays.
Pas de titre ni d'en - tête !
Peut être placé dans une boucle
@param $indice pour le compteur
@param $typeld
@param $classId
@param $typnr
@param $conn objet de connexion à la BDD, provient de PDO
@param $pays (fr ou pt)
*/
function afficheDonneesTypnr($indice, $typeld, $classId, $typnr, $conn, $pays)

// procédure afficheDonneesTypnrTab qui affiche le résultat d'une ou plusieurs requêtes à la
BDD sur le ktypnr avec stockage des résultats dans un tableau avant affichage.
// Ne pas mettre dans une boucle ! La fonction en contient déjà une.
// Adaptée à l'outil VIN Filter, contient un compteur.
// Tri sur le ktypnr
/*
@param $typeld array
@param $classId array
@param $typnr array
@param $conn objet de connexion à la BDD, provient de PDO
@param $pays (fr ou pt)
@param $siv permet l'affichage ou non d'une colonne supplémentaire spécifique à l'outil siv
immat fr
*/
function afficheDonneesTypnrTab($tabTypeld, $classId, $tabKtypnr, $conn, $pays, $siv = "")

// procédure afficheDonneesTypnrTab qui affiche le résultat d'une ou plusieurs requêtes à la
BDD sur le ktypnr avec stockage des résultats dans un tableau avant affichage.
// Ne pas mettre dans une boucle ! La fonction en contient déjà une.
```

```
// Adaptée à l'outil ktypnr FR et PT. Contient un compteur.
// Tri sur le typeld
/*
@param $typeld array
@param $classId int
@param $typnr array
@param $conn objet de connexion à la BDD, provient de PDO
@param $pays (fr ou pt)
*/
function afficheDonneesKtypnr($tabTypeld, $classId, $tabKtypnr, $conn, $pays)

/*
fonction lettreOut qui teste la présence de lettres dans l'objet
@param $var l'objet à tester
@return 0 s'il y a au moins une lettre, sinon 1
*/
function lettreOut($var) {
    return (!preg_match('/[a-zA-Z]/', $var));
}

/**
 * Fonction qui formate une immatriculation ou un VIN à partir de n'importe quelle entrée
 * Input :
 *      - variable de connexion (object) ---->> Retirée
 *      - ID du compte d'accès au webservice (int) ---->> Retirée
 *      - immatriculation ou VIN (string)
 * Output : immatriculation formatée ou VIN nettoyé (string)
 * PS : la fonction a été raccourcie
 */
function formatImmatriculationVIN($input = "") {

    // Passage en majuscule
    $input_nettoyé = strtoupper(trim($input));
```

```
// On supprime tous les caractères différents d'un chiffre ou d'une lettre
$input_nettoyé = preg_replace("#^[^0-9A-Z]#", "", $input_nettoyé);

// On vérifie si il s'agit d'un VIN
if (strlen($input_nettoyé) == 17) {
    // On supprime les caractères interdits
    $input_nettoyé = str_replace(array('I', 'O', 'Q'), array('1', '0', '0'), $input_nettoyé);
}

// Dans le cas d'une immatriculation
else {
    // Cas d'une nouvelle immatriculation
    if (preg_match("#^[A-Z]{2}[0-9]{3}[A-Z]{2}$#", $input_nettoyé)) {
        // rien
    }
    // Cas d'une ancienne immatriculation dans le sens SIV
    elseif (preg_match("#^[0-9]{1,2}[0-9A-Z]{1}[A-Z]{1,3}[0-9]{4}$#",
$input_nettoyé)) {
        // rien
    }
    // Cas d'une ancienne immatriculation dans le sens naturel
    elseif (preg_match("#^[0-9]{1,4}[A-Z]{1,3}[0-9]{1,2}[0-9A-Z]{1}$#",
$input_nettoyé)) {
        $part1 = "";
        $part2 = "";
        $part3 = "";
        for ($i = 0; $i < strlen($input_nettoyé); $i++) {
            if ($part2 == "" && preg_match("#[0-9]#", $input_nettoyé[$i])) {
                $part1.= $input_nettoyé[$i];
            }
            if ($part3 == "" && preg_match("#[A-Z]#", $input_nettoyé[$i])) {
                $part2.= $input_nettoyé[$i];
            }
            if ($part2 != "" && preg_match("#[0-9]#", $input_nettoyé[$i])) {
                $part3 = substr($input_nettoyé, $i);
                break;
            }
        }
    }
}
```

```

    }
}
switch (strlen($part1)) {
    case 1 : $part1 = '000'.$part1; break;
    case 2 : $part1 = '00'.$part1; break;
    case 3 : $part1 = '0'.$part1; break;
}
$input_nettoyé = $part3.$part2.$part1;
}
// Entrée non-conforme
else {
    // rien
}
}
return $input_nettoyé;
}

```

/**

* Fonction qui formate une immatriculation ou un VIN à partir de n'importe quelle entrée

* Input :

* - immatriculation ou VIN (string)

* Output : immatriculation formatée ou VIN nettoyé (string)

PS : spécifique au SIV DATANEO

**/

function formatImmatriculationVINsivDATANEO(\$input = "")

Même fonctionnement que la fonction précédente, à l'exception que sur les anciennes immatriculations, le département doit rester à la fin au lieu de passer à l'avant.

/**

* Fonction qui formate une immatriculation PT ou un VIN à partir de n'importe quelle entrée

* Input :

* - immatriculation ou VIN (string)

* Output : immatriculation formatée ou VIN nettoyé (string)

PS : spécifique à une immat pt et it

*/

```
function formatImmatriculationVINpt($input = "") {  
    // Passage en majuscule  
    $input_nettoyee = strtoupper(trim($input));  
    // On supprime tous les caractères différents d'un chiffre ou d'une lettre  
    $input_nettoyee = preg_replace("#[^0-9A-Z]#", "", $input_nettoyee);  
    // On vérifie s'il s'agit d'un VIN  
    if (strlen($input_nettoyee) == 17) {  
        // On supprime les caractères interdits  
        $input_nettoyee = str_replace(array('I', 'O', 'Q'), array('1', '0', '0'), $input_nettoyee);  
    }  
  
    return $input_nettoyee;  
}
```

/**

* Fonction d'appel des données CATDATA BDD fr

// Fonction modifiée

* Input :

* - immatriculation (formatée dans la fonction) ou VIN (nettoyé dans la fonction) (string)

* - login utilisateur (string)

* - mot de passe utilisateur (string)

* Output : array

* - code erreur (string)

* - données véhicule de CATDATA (object of object)

*/

```
function getDonneesWS($login, $motDePasse, $immat) {  
    // URL D'APPEL DU WS JSON  
    $url = _WS_CATDATA_URL_FR;  
  
    // PARAMETRES DE LA FONCTION  
    $Info_Vehicule = array();  
    $Info_Vehicule['pParam1'] = formatImmatriculationVIN($immat);
```

```
$Info_Vehicule['pParam2']
"<UTILISATEUR><CODUTIL>".$login."</CODUTIL><MDPUTIL>".$motD
ePasse."</MDPUTIL></UTILISATEUR>";
```

```
// CONSTRUCTION DE LA REQUETE POST
```

```
$post = http_build_query(
    array(
        'Info_Vehicule' => json_encode($Info_Vehicule)
    )
);
```

```
// HTTP : EN TETE DE LA REQUETE
```

```
$options = array(
    'http' => array(
        'header' =>
            "Content-type:application/x-www-form-urlencoded",
        'method' => "POST",
        'content' => $post
    )
);
```

```
// STREAM : CREATION DU FLUX D'ENVOI
```

```
$context = stream_context_create($options);
// print_r(formatImmatriculationVIN($immat));
```

```
$res = "";
```

```
// ENVOI + ANALYSE DU RETOUR
```

```
if(!$Info_Vehicule_Result = file_get_contents($url, false, $context)) {
    $res = "<b style='color:red'>Echec de l'envoi de la requête</b>";
} else {
    $res = $Info_Vehicule_Result;
}
```

```
$res = json_decode($res);
return $res;
```



```
}
```

```
/**
```

```
* Fonction d'appel des données AAA
```

```
// Fonction modifiée
```

```
* Input :
```

```
* - siret client (string)
```

```
* - id client (string)
```

```
* - password client (string)
```

```
* - immatriculation (formatée dans la fonction) ou VIN (nettoyé dans la fonction) (string)
```

```
* Output : array
```

```
* - code erreur (string)
```

```
* - données véhicule de AAA (object of object)
```

```
**/
```

```
function getDonneesAAA($client_siret, $client_id, $client_password, $input = "")
```

Le fonctionnement de la fonction est similaire à la précédente sauf qu'on fait appel aux services AAA.

```
/**
```

```
* Fonction d'appel des données CATDATA BDD pt
```

```
// Fonction modifiée
```

```
* Input :
```

```
* - immatriculation (formatée dans la fonction) ou VIN (nettoyé dans la fonction) (string)
```

```
* - pays
```

```
* - login
```

```
* - password
```

```
* Output : array
```

```
* - code erreur (string)
```

```
* - données véhicule de ws (object of object)
```

```
**/
```

```
function wsImmatPortugal($login_pt, $password_pt, $immat = "")
```

Fonction similaire à la fonction getDonneesWS sauf qu'on fait appel aux webservices portugais.

```
/**
```

```
* Fonction d'appel des données ws dans la base de données iterator_apply
* Input :
* - immatriculation (formatée dans la fonction) ou VIN (nettoyé dans la fonction) (string)
* - company
* - password
* - product
* - client
* Output : object
* - données du véhicule
* - message d'erreur
*/
```

```
function GetVehicleDataIT($input = "", $company = "", $password = "", $product = "", $client = "")
Fonction similaire à la fonction getDonneesWS sauf qu'on fait appel aux webservices italiens.
```

```
/** Fonction d'appel des données au webservice DATANEO
```

```
* La fonction formate l'input en appelant la fonction formatImmatriculationVINsivDATANEO
* input :
* immat ou vin
* - login
* - mot de passe
* - siret
* output : objet
*/
```

```
function appelDataNeo($input, $login, $password, $siret) {
    // formatage de l'input
    $input = formatImmatriculationVINsivDATANEO($input);

    if(strlen($input) == 17) {
        // cas d'un VIN
        $url = "https://websiv.dataneo.fr/vehicule/vin/".$input;
    } else {
        // cas d'un immat
        $url = "https://websiv.dataneo.fr/vehicule/immatriculation/".$input;
    }
}
```

```
$return = exec('curl -XGET ' . $url . ' -d \{'
    "Login": " . $login . '",
    "Password": " . $password . '",
    "Siret": " . $siret . '"
    }\');

return json_decode($return);
}
```

- [pdo.php](#)

Ce fichier possède les éléments de connexion à la base de données : nom de la table, chaîne de connexion, hôte, identifiant et mot de passe. L'ensemble est stocké dans la variable \$conn.

- [ws_vinfilter.php](#)

Dans ce fichier figures les fonctions d'appel aux webservices et les paramètres nécessaires :

- `function wsGetVtMakeList($parametres=array())`

Fonction qui cherche la marque d'un véhicule à partir d'un VIN.

Renvoie un objet contenant une ou plusieurs marques (id et libellé) ou rien.

Prend en paramètre un array. Ex :

```
$retour = wsGetVtMakeList(array(
    "vin"                => "VNKKC96380A191500",
    "vt_company"         => "TecAllianceFRweb",
    "vt_country"         => "fr",
    "vt_language"        => "fr",
    "vt_password"        => "Start1234!",
    "vt_user"            => "TfIWeb01",
    "vt_cars"            => 1,
    "vt_trucks"          => 1
));
```

- `function wsGetVtRangeList($parametres=array())`

Fonction qui cherche le modèle d'un véhicule à partir d'un VIN et du makeld.

Renvoie un objet contenant un ou plusieurs modèles (id et libellé) ou rien.

Prend en paramètre un array. Ex :

```
$retour = wsGetVtRangeList(array(  
    "vin"                => "VNKKC96380A191500",  
    "vt_company"         => "TecAllianceFRweb",  
    "vt_country"          => "fr",  
    "vt_language"         => "fr",  
    "vt_makeld"           => 41,  
    "vt_password"         => "Start1234!",  
    "vt_user"             => "TfIWeb01",  
    "vt_cars"             => 1,  
    "vt_trucks"           => 1  
));
```

- fonction wsGetVtTypeList(\$parametres=array())

Fonction qui cherche les informations d'un véhicule à partir d'un VIN et du rangeld.

Renvoie un objet contenant une ou plusieurs versions de véhicule ou rien.

Prend en paramètre un array. Ex :

```
$retour = wsGetVtTypeList(array(  
    "vin"                => "VNKKC96380A191500",  
    "vt_company"         => "TecAllianceFRweb",  
    "vt_country"          => "fr",  
    "vt_language"         => "fr",  
    "vt_rangeld"          => 41,  
    "vt_password"         => "Start1234!",  
    "vt_user"             => "TfIWeb01",  
    "vt_cars"             => 1,  
    "vt_trucks"           => 1  
));
```

- fonction wsVIN(\$parametres=array())

Fonction qui cherche les informations d'un véhicule à partir d'un VIN. Elle

Renvoie un objet contenant une ou plusieurs versions de véhicule si une seule marque et un seul modèle, sinon rien.

Prend en paramètre un array. Ex :

```
$retour = wsVIN(array(  
    "vin"           => "VNKKC96380A191500",  
    "vt_company"    => "TecAllianceFRweb",  
    "vt_country"     => "fr",  
    "vt_language"    => "fr",  
    "vt_password"    => "Start1234!",  
    "vt_user"        => "TflWeb01",  
    "vt_cars"        => 1,  
    "vt_trucks"      => 1  
));
```

- function wsKtypnrFromTypeId(\$parametres=array())

Fonction qui associe le typeId à un ktypnr.

Renvoie un ktypnr ou rien.

Prend en paramètre un array. Ex :

```
$retour = wsKtypnrFromTypeId(array(  
    "vt_typeid"      => "62583"  
    "vt_company"     => "TecAllianceFRweb",  
    "vt_password"    => "Start1234!",  
    "vt_user"        => "TflWeb01"  
));
```

- function wsVehiculeName(\$parametres=array())

Fonction qui renvoie des informations sur un véhicule à partir de son typeId (marque, modèle, version, informations techniques).

Renvoie un objet contenant des informations sur un véhicule.

Prend en paramètre un array. Ex :

```
$retour = wsVehiculeName(array(  
    "vt_company"      => "TecAllianceFRweb",  
    "vt_country"      => "fr",  
    "vt_language"     => "fr",  
    "vt_password"     => "Start1234!",  
    "vt_user"         => "TflWeb01",  
));
```

```

"vt_typeid"          => "62577"
));

        o function wsTypeid($parametres=array())
Fonction qui associe le ktypnr à un typeid.
Renvoie un typeid ou rien.
Prend en paramètre un array. Ex :
$retour = wsTypeid(array(
    "vt_company"      => "TecAllianceFRweb",
    "vt_user"         => "TfIWeb01",
    "vt_password"     => "Start1234!",
    "vt_typnr"        => "119518",
    "vt_classId"      => "1"
));

```

Il est possible de réaliser un affichage en dur des données pour chaque fonction.

- Dans le dossier cible :

Chaque fichier outil fait appel aux fichiers suivants :

- o config.php (pour les identifiants, les variables globales, les url pour l'appel au Webservice...)
- o pdo.php (pour l'appel à la base de données)
- o ws_vinfilter.php (pour l'appel des fonctions du Webservice)
- o fonctions.php (pour l'appel des fonctions spécifiques)
- o tableaux.css (pour la mise en forme)

- [immat_vin_catdata_fr.php](#)

La langue et le pays sont fixés à fr. Les données utiles sont récupérées dans le formulaire.

Ex :

```

// récupération du login et affichage dans la zone de saisie
<input type="text" id="txt_login" name="txt_login" size="30" value="<?
    if(empty($_POST['txt_login'])) {
        $login = _LOGIN_DEPART;
    } else {
        $login = $_POST['txt_login'];

```

```
}
```

```
echo $login; ?>" required />
```

Par défaut, les variables prennent les valeurs déterminées par les defines.

Il y a un contrôle sur l'immat pour limiter la saisie aux chiffres et aux lettres.

Il est ensuite fait appel aux services CATDATA et AAA pour récupérer les données à partir de l'immat saisie :

```
// Appel WS CATDATA
```

```
$resWS = getDonneesWS($login, $motDePasse, $immat);
```

```
// Appel AAA
```

```
$resAAA = getDonneesAAA(_CLIENT_SIRET, _CLIENT_ID, _CLIENT_PASSWORD,
$immat);
```

Enfin l'appel au service DATA NEO est exécuté pour récupérer ses données :

```
// appel au webservice DATA NEO
```

```
$resDataNeo = appelDataNeo($immat, _LOGIN_DATA_NEO, _PASSWORD_DATA_NEO,
_SIRET_DATA_NEO);
```

Les variables stockant les résultats sont envoyées dans les fonctions spécifiques pour affichage des données, après un test sur l'existence d'un retour. Un booléen permet d'enregistrer si un affichage était impossible pour lancer un message d'erreur.

Ex avec le tableau CATDATA :

```
<!-- tableau CATDATA -->
```

```
<table><?php
```

```
if($immat != "") { // si démarrage de l'outil : pas d'affichage
```

```
if(isset($resWS->VEHICULE)) {
```

```
?><caption>SIV CATDATA</caption><?php
```

```
afficheEnTeteImmat();
```

```
afficheDonneesImmat($resWS->VEHICULE);
```

```
} else {
```

```
$erreur = true;
```

```
}
```

```
}
```

```
?></table>
```

Pour le tableau DATANEO il faut en plus récupérer un second membre dans le résultat :

```
if($immat != "") { // si démarrage de l'outil : pas d'affichage
    if(isset($resDataNeo->Vehicle)) {
        ?><caption>SIV DATA NEO</caption><?php
        afficheEnTete();
        AfficheDonnees($resDataNeo->Vehicle);
        // rajout du dernier membre dans un array pour la fonction d'affichage
        $tab = array('Responseld' => $resDataNeo->Responseld);
        AfficheDonnees($tab);
    } else {
        echo "<p style='color:red'><strong>Pas d'informations DATANEO sur ce
véhicule IMMAT / VIN = ".$immat."</strong></p>";
    }
}
```

Les tableaux CATDATA, AAA et DATANEO sont placés dans une même balise div.

Une autre balise div contient un affichage horizontal des données issues des ktpynr récupérés dans le tableau CATDATA puis dans le tableau AAA. Les ktpnrs sont récupérés et stockés dans des arrays, après avoir tester la présence ou non d'un ktpynr non nul :

```
// récupération des données CATDATA
$tabTypVeh = $resWS->VEHICULE->LTYPVEH->TYPVEH;
// récupération des ktpynr CATDATA
$tabTypeId = array();
$tabKtpynr = array();
if(count($tabTypVeh) > 1) {
    foreach($tabTypVeh as $element) {
        $tabTypeId[] = array(
            "Type Id" => appelwsTypeId(_COMPANY, _USER, _PASSWORD,
substr(($element->IDTYPVEH), 2, -3), substr(($element->IDTYPVEH), 0, 1))-
>GetVtAdcTypeIdByTcdTypeAndClassResult
        );
        $tabKtpynr[] = array(
            "Ktpynr" => substr(($element->IDTYPVEH), 2, -3)
        );
        $tabClassId[] = substr(($element->IDTYPVEH), 0, 1);
    }
}
```



```

    }
} else {
    $tabTypeld = array(
        "Type Id" => appelwsTypeld(_COMPANY, _USER, _PASSWORD,
        substr(($tabTypVeh[0]->IDTYPVEH), 2, -3), substr(($tabTypVeh[0]->IDTYPVEH), 0, 1))-
        >GetVtAdcTypeldByTcdTypeAndClassResult
    );
    $tabKtypnr = array(
        "Ktypnr" => substr(($tabTypVeh[0]->IDTYPVEH), 2, -3)
    );
    $tabClassld[] = substr(($tabTypVeh[0]->IDTYPVEH), 0, 1);
}

```

On remarque que le typeld est récupéré par appel à la fonction appelwsTypeld. Notez également qu'il faut utiliser la fonction substr pour prélever le ktypnr et le classld dans la chaîne de résultat.

Le tableau de ktypnr, de typeld et de classld sont envoyés dans la fonction d'affichage comme plus haut :

```
afficheDonneesTypnrTab($tabTypeld, $tabClassld, $tabKtypnr, $conn, $pays, "CATDATA");
```

Le principe est le même pour les données AAA. Le ktypnr sera récupéré de la façon suivante :

```
// récupération du ktypnr AAA
```

```
$tabk_type = $resAAA['dataAAA']->return->k_type;
```

Contrairement à CATDATA, AAA ne fournit pas plusieurs ktypnrs, mais soit un seul, soit rien.

La dernière balise div contient le tableau de données issues du VinFilter. La requête n'est lancée qu'après le clic sur le bouton 'Afficher VinFilter' (script placé dans le header) :

```

$(document).ready(function() {
    var click = false;
    $("#afficheVinFilter").click(function() {
        if(!click) { // accès au résultat du VinFilter
            // afficher image loader
            $("#divImmatVIN").html("<img src='img/ajax-loader-large.gif'
width='30'>");
            $.ajax({
                type: "POST",
                url: "ajax/immat_vinfilter_fr.php",

```

```
data: {recupImmat : $("#txt_immat").val(),
      recupLogin : $("#txt_login").val(),
      recupMotDePasse : $("#txt_mdp").val()
    },
success: function(data){
    // affichage du tableau VinFilter
    $("#divImmatVIN").html(data);
}
});
$(location).attr('href', "#divImmatVIN");
$("#afficheVinFilter").html(" Masquer VinFilter");
click = true;
} else { // suppression du résultat du VinFilter
    $("#divImmatVIN").empty();
    $(location).attr('href', "#");
    $("#afficheVinFilter").html(" Afficher VinFilter");
    click = false;
}
});
});
```

Le clic sur le bouton amène l'utilisateur directement à l'emplacement de la div. Un nouveau clic supprime le tableau et renvoie l'utilisateur en haut de page.

Enfin un dernier test sur le booléen \$erreur permet l'affichage ou non d'un unique message d'erreur. Attention, des erreurs sur les tableaux ktypnr issu de CATDATA, de AAA et de DATANEO provoqueront également des messages d'erreur.

- [immat_vin_catdata_pt.php](#)

Même principe et organisation que dans le fichier précédent.

Toutefois, la langue et le pays sont fixés à pt.

Il n'y a pas d'appel vers AAA, uniquement vers le webservice CATDATA :

```
// Appel WS CATDATA
```

```
$resWS = wsImmatPortugal($login, $motDePasse, $immat);
```

Le retour est fractionné pour permettre un double affichage : à gauche un tableau avec des données générales, à droite un tableau par type de véhicule :

```
// tableau de gauche
```

```
afficheEnTeteImmat();
```

```
afficheDonneesImmat($resWS->oVehiclePlateNumberInfo);
```

```
// tableau de droite
```

```
foreach($resWS->aTecdocInfo as $element) {
```

```
    ?><table><?php
```

```
    afficheEnTeteImmat();
```

```
    afficheDonneesImmat($element);
```

```
    ?></table><?php
```

```
}
```

- [immat_vin_catdata_it.php](#)

Même principe et organisation que dans le fichier précédent.

Toutefois, la langue et le pays sont fixés à it.

Ici il faut récupérer en plus des trois paramètres le nom de la compagnie :

```
// récupération de la compagnie et affichage dans la zone de saisie
```

```
if(empty($_POST['txt_mdp'])) {
```

```
    $company = _COMPANY_IT;
```

```
} else {
```

```
    $company = $_POST['txt_company'];
```

```
}
```

L'information sera aussi récupérée dans le script :

```
recupCompagnie : $("#txt_company").val()
```

Il n'y a pas d'appel vers AAA, uniquement vers le webservice CATDATA :

```
// Appel WS CATDATA
```

```
$resWS = GetVehicleDataIT($immat, $company, $motDePasse, _PRODUCT_IT, $login);
```

Là aussi le retour est fractionné pour obtenir à gauche un tableau de données générales et à droite un ou plusieurs tableaux spécifiques ; après un test sur l'existence d'un retour non nul et ensuite sur la nature de l'élément `AllestimentoCompletoTecDoc` :

// à gauche

```
afficheEnTeteImmat();
```

```
afficheDonneesImmatIt($resWS->GetRTCompletoDaTargaMinTecDocResult);
```

// à droite

```
if(is_array($resWS->GetRTCompletoDaTargaMinTecDocResult->Allestimenti-
>AllestimentoCompletoTecDoc)) {
```

```
    foreach($resWS->GetRTCompletoDaTargaMinTecDocResult->Allestimenti-
>AllestimentoCompletoTecDoc as $element) {
```

```
        ?><table><?php
```

```
        afficheEnTeteImmat();
```

```
        afficheDonneesImmatIt($element);
```

```
        ?></table><?php
```

```
    }
```

```
} else {
```

```
    foreach($resWS->GetRTCompletoDaTargaMinTecDocResult->Allestimenti as
$element) {
```

```
        ?><table><?php
```

```
        afficheEnTeteImmat();
```

```
        afficheDonneesImmatIt($element);
```

```
        ?></table><?php
```

```
    }
```

```
}
```

- **ktypnrFR**

Le pays est fixé à fr. Un script jquery permet la modification d'un label en fonction de l'option choisie (ktypnr pour véhicule léger ou ntypnr pour poids lourd) :

```
$(document).ready(function() {
```

```
    $("#classId1").click(function() {
```

```
        $("#lbl_saisie").html("Entrez un ktypnr : ");
```

```
    });
```

```
    $("#classId2").click(function() {
```

```
        $("#lbl_saisie").html("Entrez un ntypnr : ");
```

```
    });
```

```
});
```

Par défaut une option est déjà cochée, ce qui entraîne un label adapté à l'option par défaut, et l'option choisie par bouton radio reste après chargement de la page :

```
<label for="lbl_saisie" id="lbl_saisie"><?php
    // affichage du message au lancement de la page
    if(empty($_POST['classId'])) {
        if (_CLASSID_DEPART == 1) {
            ?>Entrez un ktypnr : <?php
        } else if (_CLASSID_DEPART == 2) {
            ?>Entrez un ntypnr : <?php
        }
    } else if ($_POST['classId'] == 1) {
        ?>Entrez un ktypnr : <?php
    } else if ($_POST['classId'] == 2) {
        ?>Entrez un ntypnr : <?php
    }
?></label>

<input type="radio" id="classId1" name="classId" value=1 <?php
    if (empty($_POST['classId'])) {
        if (_CLASSID_DEPART == 1) {
            echo 'checked="checked"';
        }
    } else if ($_POST['classId'] == 1) {
        echo 'checked="checked"';
    }
?>
```

Un formulaire permet de récupérer le ktypnr et le classId est saisi à partir de l'option choisie. Un contrôle est appliqué sur la saisie du ktypnr pour vérifier la présence de chiffres, de lettres et de séparateurs (, ; /) uniquement et la suppression des espaces :

```
<form action="ktypnrFR.php" method="post" name="form">
<p id="bas" name="bas">
...
<input type="text" id="txt_saisie" name="txt_saisie" size="50" pattern="[0-9A-Za-z,;\/s-]{1,}"
required />
```

```
<input type="submit" id="btn_saisie" name="btn_saisie" value="Envoyer"/>
</p></form>
```

```
<?php
```

```
// récupération du typnr saisi après vérification du contenu
```

```
if(empty($_POST['txt_saisie'])) {
    $typnr = _TYPNR_DEPART;
} else {
    $typnr = str_replace(" ", "", $_POST['txt_saisie']);
    if(empty($typnr)) {
        $typnr = _TYPNR_DEPART;
    }
}
```

```
// récupération du classId
```

```
if (!empty($_POST['classId'])) {
    $classId = $_POST['classId']; // 1 pour vl, 2 pour pl
} else {
    $classId = _CLASSID_DEPART;
}
```

La donnée récupérée est découpée selon les séparateurs présents pour récupérer le ou les ktypnr dans un array s'il y a des séparateurs ou laissée dans une variable simple dans le cas contraire. Une fonction spécifique lettreOut(\$ktypnr) vérifie la présence ou non de lettres pour supprimer les valeurs avec des lettres. Un second filtre supprime les cases vides. A partir de là il est possible de récupérer le ou les typeld grâce à la fonction appelwsTypeld et de stocker les informations dans des arrays :

```
// récupération des ktypnr après découpage de la chaîne et du typeld
```

```
// test sur la présence de séparateurs
```

```
if(!preg_match('[\W]', $typnr)) {
    // test sur la présence de lettres dans le ktypnr
    if(lettreOut($typnr)) {
        $tabKtypnr = array(
            "Ktypnr" => $typnr
        );
        $tabTypeld = array(
```

```

        "Type Id" => appelwsTypeId(_COMPANY, _USER,
        _PASSWORD, $tabKtypnr, $classId)->GetVtAdcTypeIdByTcdTypeAndClassResult
    );
} else {
    $tabKtypnr = array(
        "Ktypnr" => "0"
    );
    $tabTypeId = array(
        "Type Id" => "0"
    );
}
} else {
    // filtrer en supprimant les cases avec des lettres
    $tab = array_filter(preg_split('[\W]', $typnr), "lettreOut");
    // refiltrer en supprimant les cases vides
    $tab = array_filter($tab);
    $tabKtypnr = array();
    $tabTypeId = array();
    if(!empty($tab)) {
        foreach($tab as $element) {
            if(!empty($element)) {
                // si la case n'est pas vide on prend la valeur
                $tabKtypnr[] = array(
                    "Ktypnr" => $element
                );
                $tabTypeId[] = array(
                    "Type Id" => appelwsTypeId(_COMPANY,
        _USER, _PASSWORD, $element, $classId)->GetVtAdcTypeIdByTcdTypeAndClassResult
                );
            }
        }
    }
} else {
    // cas où le tableau est vide
    $tabKtypnr = array(
        "Ktypnr" => "0"
    );

```

```
);
$tabTypeId = array(
    "Type Id" =>"0"
);
}
}
```

Les arrays sont ensuite envoyées en paramètres dans les fonctions d'affichage. Un test sur \$classId permet d'afficher l'en - tête adapté :

```
if($classId != 2) {
    // affichage titre et en - tête BDD France vl
    ?><caption>Ktypnr Véhicule léger (BDD France)</caption><?
    afficheEnTeteHorizPaysClassId($pays, $classId);
} else if($classId == 2) {
    // affichage titre et en - tête BDD France pl
    ?><caption>Ntypnr Poids lourd (BDD France)</caption><?
    afficheEnTeteHorizPaysClassId($pays, $classId);
}

// affichage des données de tous les véhicules avec tri
if($typnr != "") {
    afficheDonneesKtypnr($tabTypeId, $classId, $tabKtypnr, $conn, $pays);
}
```

- [ktypnrPT](#)

Même fonctionnement que dans le fichier précédent.

Le pays est fixé à pt et le classId à 1 car il n'y a que des véhicules légers. Du coup l'affichage se fait sans test sur le classId.

- [typeid_vers_typnr](#)

Le typeid est récupéré dans un formulaire après contrôle sur le contenu. Le typeid ne doit contenir que des chiffres, entre 1 et 9 chiffres maximum :

```
<form action="typeid_vers_typnr.php" method="post" name="form">
<p>
    <label for="lbl_saisie" id="lbl_saisie">Entrez un typeid : </label>
```



```
<input type="text" id="txt_saisie" name="txt_saisie" size="9" maxlength="9"
pattern="[0-9]{1,9}" required />
```

```
<input type="submit" id="btn_saisie" name="btn_saisie" value="Envoyer"/>
```

```
</p>
```

```
</form>
```

```
...
```

```
// récupération du typeld après vérification du contenu
```

```
if (empty($_POST['txt_saisie'])) {
```

```
    $typeld = _TYPEID_DEPART;
```

```
} else {
```

```
    $typeld = $_POST['txt_saisie'];
```

```
}
```

Le typeld permet ensuite la récupération du ktypnr et du classld par l'appel à deux fonctions spécifiques :

```
// récupération du classld et du typnr à partir du typeld si saisi
```

```
if($typeld != "") {
```

```
    $classld = appelwsKtypnrFromTypeld($typeld, _COMPANY, _PASSWORD, _USER)-
>GetVtTcdTypeAndClassByAdcTypeldResult->TcdClassld;
```

```
    $typnr = appelwsKtypnrFromTypeld($typeld, _COMPANY, _PASSWORD, _USER)-
>GetVtTcdTypeAndClassByAdcTypeldResult->TcdTypeld;
```

```
}
```

Il est désormais possible de mettre en forme l'affichage des résultats.

Le tableau de gauche contient les données issues de la fonction wsVehiculeName. L'appel n'est fait que si un typeld existe (d'où le test sur son existence), sinon on n'affiche que l'en-tête :

```
// --- mise en forme de la fonction wsVehiculeName --- tableau gauche ---
```

```
if($typeld != "") {
```

```
    $retour1 = appelwsVehiculeName(_COMPANY, _PASSWORD, _USER, $typeld);
```

```
    // vérification du contenu de $retour1
```

```
    if (!empty($retour1->GetVtTypeDetailsResult->VtTypeDetail)) {
```

```
        $tab = $retour1->GetVtTypeDetailsResult->VtTypeDetail->TypeDetails-
>VtTypeAdditionalDetail;
```

```
    } else {
```

```
        echo '<p style="color:red;"><strong>Informations manquantes sur ce véhicule
(typeld = <u>'.$typeld.'</u>) dans la BDD.</strong></p>';
```

```

}
?><section id="tabG">
    <table>
        <caption>Type Id : <?php print_r($typeld) ?></caption><?php
        afficheEnTete();
        if (isset($stab)) {
            afficheDonneesTab($stab);
        }
    ?></table>
</section><?php
} else {
    ?><section id="tabG">
        <table>
            <caption>Type Id : <?php print_r($typeld) ?></caption><?php
            afficheEnTete();
        ?></table>
    </section><?
}

```

Le tableau de droite contient les données issues d'une requête à la base de données fr, uniquement s'il existe un typeld, et en fonction du classld (appel de la base de données des véhicules légers ou des poids lourds). Si la requête ne renvoie pas de données, un message d'erreur s'affiche, de même si le classld est nul (le typeld ne renvoie pas à un véhicule) :

// --- mise en forme de la requête KtypnrFromTypId --- tableau droite ---

```

if($typeld != "") {
    if ($classld != 2 && $classld != 0) { // cas des véhicules avec classld = 4 ou autre
    (éventuellement)

        // tables des véhicules
        $req = $conn->query('SELECT * FROM w_veh_vl WHERE ktypnr = '.$typnr.'');
        $ligne = $req->fetch(PDO::FETCH_ASSOC);
        if (empty($ligne)) { // vérification du remplissage de la requête
            echo '<p style="color:red;"><strong>Informations manquantes sur ce
            véhicule (ktypnr = <u>'.$typnr.'</u>) dans la BDD.</strong></p>';
        }
        ?><section id="tabD1"><table>

```

```

<caption>Ktypnr Véhicule léger : <?php print_r($typnr) ?></caption><?php

afficheEnTete();
if (!empty($ligne)) {
    AfficheDonnees($ligne);
}
?></table></section><?php
} else if ($classId == 2) {
    // tables des poids lourds
    $req = $conn->query('SELECT * FROM w_veh_pl WHERE ntypnr = '.$typnr.'');
    $ligne = $req->fetch(PDO::FETCH_ASSOC);
    if (empty($ligne)) { // vérification du remplissage de la requête
        echo '<p style="color:red;"><strong>Informations manquantes sur ce
véhicule (ntypnr = <u>'.$typnr.'</u>) dans la BDD.</strong></p>';
    }
    ?><section id="tabD2"><table>
<caption>Ntypnr Poids lourds : <?php print_r($typnr) ?></caption><?php

afficheEnTete();
if (!empty($ligne)) {
    AfficheDonnees($ligne);
}
?></table></section><?php
} else if ($classId == 0) {
    // absence du véhicule dans la BDD
    echo '<p style="color:red;"><strong>Ce véhicule ne figure pas dans la
BDD.</strong></p>';
}
} else {
    ?><section id="tabD1"><table>
<caption>Ktypnr Véhicule léger : </caption><?php
afficheEnTete();
?></table></section><?
}

```

- typnr_vers_typeid

Le fonctionnement est l'inverse du fichier précédent. Les tableaux d'affichage sont également inversés.

Le classId doit être saisi par bouton radio, ce qui modifie le label. Par défaut une option est cochée :

```
<input type="radio" id="classId1" name="classId" value=1 <?php
if (empty($_POST['classId'])) {
    if (_CLASSID_DEPART == 1) {
        echo 'checked="checked"';
    }
} else if ($_POST['classId'] == 1) {
    echo 'checked="checked"';
}
?> /><label for="classId1">ktypnr (véhicule léger)</label>
<input type="radio" id="classId2" name="classId" value=2 <?php
if (empty($_POST['classId'])) {
    if (_CLASSID_DEPART == 2) {
        echo 'checked="checked"';
    }
} else if ($_POST['classId'] == 2) {
    echo 'checked="checked"';
}
?> /><label for="classId2">ntypnr (poids lourd)</label>
```

Le label peut ainsi être modifié au chargement de la page :

```
// affichage du message au lancement de la page
if (empty($_POST['classId'])) {
    if (_CLASSID_DEPART == 1) {
        ?>Entrez un ktypnr : <?php
    } else if (_CLASSID_DEPART == 2) {
        ?>Entrez un ntypnr : <?php
    }
} else if ($_POST['classId'] == 1) {
    ?>Entrez un ktypnr : <?php
```

```
} else if ($_POST['classId'] == 2) {  
    ?>Entrez un ntypnr : <?php  
}
```

Le classId est le ktypnr sont récupérés grâce au formulaire :

// récupération du typnr saisi après vérification du contenu

```
if (empty($_POST['txt_saisie'])) {  
    $typnr = _TYPNR_DEPART;  
} else {  
    $typnr = $_POST['txt_saisie'];  
}
```

// récupération du classId

```
if (!empty($_POST['classId'])) {  
    $classId = $_POST['classId']; // 1 pour vl, 2 pour pl  
} else {  
    $classId = _CLASSID_DEPART;  
}
```

Le typeld pourra alors est récupéré par appel à la fonction appelwsTypeld :

// récupération du vt_typeid à partir de la fonction wsTypeld

```
$typeld = appelwsTypeld(_COMPANY, _USER, _PASSWORD, $typnr, $classId)-  
>GetVtAdcTypeldByTcdTypeAndClassResult;
```

Le tableau de droite contient les données issues de la fonction wsVehiculeName. L'appel puis l'affichage sont identiques au fichier précédent.

De même, le tableau de gauche contient les données issues de la base de données à partir du ktypnr récupéré. Là aussi l'appel à la base de données et l'affichage des résultats sont identiques au fichier précédent.

- wsVIN

Un bouton radio permet la sélection du pays. Le choix effectué va modifier le label grâce à un script :

```
<p id="haut" name="haut">  
<input type="radio" id="paysFr" name="pays" value="fr" <?php
```

```

if (empty($_POST['pays'])) {
    if (_PAYS_DEPART == "fr") {
        echo 'checked="checked"';
    }
} else if ($_POST['pays'] == "fr") {
    echo 'checked="checked"';
}

?> /><label for="paysFr">BDD France</label>
<input type="radio" id="paysPt" name="pays" value="pt" <?php
if (empty($_POST['pays'])) {
    if (_PAYS_DEPART == "pt") {
        echo 'checked="checked"';
    }
} else if ($_POST['pays'] == "pt") {
    echo 'checked="checked"';
}
?> /><label for="paysPt">BDD Portugal</label>
</p>
<p id="bas" name="bas">
<label for="lbl_saisie" id="lbl_saisie"><?php
// affichage du message au lancement de la page
if (empty($_POST['pays'])) {
    if (_PAYS_DEPART == "fr") {
        ?>Entrez un VIN (BDD France) : <?php
    } else if (_PAYS_DEPART == "pt") {
        ?>Entrez un VIN (BDD Portugal) : <?php
    }
} else if ($_POST['pays'] == "fr") {
    ?>Entrez un VIN (BDD France) : <?php
} else if ($_POST['pays'] == "pt") {
    ?>Entrez un VIN (BDD Portugal) : <?php
}
?></label>

```

Le script :

```
$(document).ready(function() {
    $("#paysFr").click(function() {
        $("#lbl_saisie").html("Entrez un VIN (BDD France) : ");
    });
    $("#paysPt").click(function() {
        $("#lbl_saisie").html("Entrez un VIN (BDD Portugal) : ");
    });
    ...
});
```

Le VIN est ensuite récupéré avec contrôle sur le contenu des caractères (ne sont autorisés que les chiffres et les lettres) et un autre script transforme les minuscules en majuscules et les caractères interdits (les 'o' et les 'q' en 0, les 'i' en 1) :

```
<input type="text" id="txt_saisie" name="txt_saisie" size="32" maxlength="17" pattern="[0-9A-Z]{17}" value="<?
// récupération du VIN après vérification du contenu et affichage dans la zone de saisie
if(empty($_POST['txt_saisie'])) {
    $vin = _VIN_DEPART;
} else {
    $vin = $_POST['txt_saisie'];
}

echo $vin; ?>" required />
```

Le script :

```
$(document).ready(function() {
    ...
    // modification à la volée des frappes incorrectes
    $("#txt_saisie").keyup(function(event) {
        // transformation des minuscules en majuscules
        var texte = $("#txt_saisie").val();
        $("#txt_saisie").val(texte.toUpperCase());
        // transformation des O en 0, des Q en 0, des I en 1
        if(event.key == 'o' || event.key == 'O' || event.key == 'q' || event.key == 'Q') {
            var texte = ($("#txt_saisie").val()).slice(0, -1);
```

```

    texte += "0";
    $("#txt_saisie").val(texte);
}
if(event.key == 'i' || event.key == 'l') {
    var texte = ($("#txt_saisie").val()).slice(0, -1);
    texte += "1";
    $("#txt_saisie").val(texte);
}
});
});

```

Ensuite le choix du pays est stocké dans une variable, ce qui permet également la récupération de la langue :

// récupération du pays et de la langue après vérification du contenu

```

if(empty($_POST['pays'])) {
    $pays = _PAYS_DEPART;
    $langue = _LANGUE_DEPART;
} else {
    $pays = $_POST['pays'];
    $langue = $_POST['pays'];
}

```

On cherche ensuite à récupérer tous les éléments nécessaires : marque, modèle, données du véhicule, nombre de versions, kypnr, typeld, classld. Cette récupération se fera par appel à des fonctions spécifiques :

// pour accéder au nom de la marque et son id

```

$retourGetVtMakeList = appelwsGetVtMakeList($vin, _COMPANY, $pays, $langue,
_PASSWORD, _USER);

```

```

if (!empty($retourGetVtMakeList->GetVtMakeListResult->VtMake)) {
    if(sizeof($retourGetVtMakeList->GetVtMakeListResult->VtMake) == 1) {
        $tabMarque = array(
            "Marque"           =>          $retourGetVtMakeList-
>GetVtMakeListResult->VtMake->MakeName
        );
        $makeId = $retourGetVtMakeList->GetVtMakeListResult->VtMake->MakeId;
    }
}

```



```

} else if(sizeof($retourGetVtMakeList->GetVtMakeListResult->VtMake) > 1) {
    $tabMarque = array();
    $tabMake = $retourGetVtMakeList->GetVtMakeListResult->VtMake;
    for($i = 0; $i < count($tabMake); $i++) {
        $var = $tabMake[$i];
        $tabMarque[$i] = array(
            "Marque" => $var->MakeName
        );
    }
    $makeld = "";
    $nbMarque = sizeof($retourGetVtMakeList->GetVtMakeListResult->VtMake); //
pour l'affichage
    echo '<p><strong>Il y a '.$nbMarque.' marques différentes.</strong></p>';
}
} else {
    $tabMarque = array(
        "Marque" => ""
    );
    $makeld = "";
    if($vin != "") { // pas d'affichage d'erreur si lancement de l'outil
        echo '<p style="color:red"><strong>Ce véhicule ne figure pas dans la
BDD.</strong></p>';
    }
}
}

```

Les marques sont stockées dans des arrays. Un compteur permet de stocker le nombre de marques. Par défaut, en cas d'erreur ou de contenu vide, le tableau de résultat est vide mais contient les clés.

Le makeld est récupéré pour être envoyé en paramètre dans la fonction d'appel des modèles, puis le rangeld pour les versions.

Le fonctionnement est identique pour les modèles et les versions (ainsi que les données), à l'exception de la fonction appelée :

```

// pour accéder au nom du modèle et son id
$retourGetVtRangeList = appelwsGetVtRangeList($vin, _COMPANY, $pays, $langue,
$makeld, _PASSWORD, _USER);

// pour récupérer le modèle :
$tabModele = array(

```

```

"Modèle" => $retourGetVtRangeList->GetVtRangeListResult-
>VtRange->RangeName
);

```

```

// pour accéder au nom de la version et son id, et aux détails du véhicule ou des véhicules
$retourGetVtTypeList = appelwsGetVtTypeList($vin, _COMPANY, $pays, $langue, $rangelid,
_PASSWORD, _USER);
// pour récupérer la version :
$tabVersion = array(
    "Version" => $retourGetVtTypeList->GetVtTypeListResult-
>VtType->TypeName
);

```

Une fonction spécifique permet de stocker les détails des véhicules après leur récupération :

```

$tab = $retourGetVtTypeList->GetVtTypeListResult->VtType->TypeDetails-
>VtTypeAdditionalDetail;
// remplissage du tableau tabDetail avec tous les éléments
$tabDetail = ajoutDetail($tab);

```

Les modèles, les versions et les détails sont stockées dans des arrays. Un compteur permet de stocker le nombre de modèles et un autre le nombre de versions. Par défaut, en cas d'erreur ou de contenu vide, le tableau de résultat est vide mais contient les clés.

A partir du tableau des détails, il est possible de récupérer le ou les typeld. Puis un appel vers la fonction `appelwsKtypnrFromTypeld` permet d'obtenir le ou les ktypnrs et le classld. Toutes ces informations sont stockées dans des arrays, sauf le classld :

```

// accès au typnr et au classld à partir du typeld récupéré
// tester si un seul typeld ou plusieurs et son existence
if(!empty($retourGetVtTypeList->GetVtTypeListResult->VtType)) {
    if(sizeof($retourGetVtTypeList->GetVtTypeListResult->VtType) > 1) {
        // cas de plusieurs versions
        $tabTypeld = array(); // pour le tableau VINfilter
        $tabKtypnr = array();
        $tabDet = $retourGetVtTypeList->GetVtTypeListResult->VtType;
        for($i = 0; $i < sizeof($tabVersion); $i++) {
            $var = $tabDet[$i];
            $tabTypeld[$i] = array(

```

```

                                "Type Id" => $var->TypeId
                                );
                                $tabKtypnr[$i] = array(
                                "Ktypnr"
                                =>
appelwsKtypnrFromTypeId($var->TypeId, _COMPANY, _PASSWORD, _USER)-
>GetVtTcdTypeAndClassByAdcTypeIdResult->TcdTypeId
                                );
                                }
                                } else {
                                // cas de version unique
                                $tabTypeId = array(
                                "Type Id" => $retourGetVtTypeList-
>GetVtTypeListResult->VtType->TypeId
                                );
                                $classId = appelwsKtypnrFromTypeId($tabTypeId['Type Id'],
                                _COMPANY, _PASSWORD, _USER)->GetVtTcdTypeAndClassByAdcTypeIdResult-
                                >TcdClassId;
                                $tabKtypnr = array(
                                "Ktypnr" => appelwsKtypnrFromTypeId($tabTypeId['Type Id'], _COMPANY,
                                _PASSWORD, _USER)->GetVtTcdTypeAndClassByAdcTypeIdResult->TcdTypeId
                                );
                                }
                                } else {
                                // cas sans version
                                $classId = 0;
                                $tabTypeId = array(
                                "Type Id" => ""
                                );
                                $tabKtypnr = array(
                                "Ktypnr" => ""
                                );
                                }

```

On peut désormais gérer les affichages en envoyant les arrays en paramètres.

Pour le tableau des ktypnr, s'il y a plus que un typeId, le classId sera stocké dans un array car la fonction d'affichage associée ne gère que des arrays.

Un test s'exécute donc sur la taille de l'array \$tabTypepd (un ou plusieurs véhicules) puis sur le pays pour un affichage adéquat de l'en - tête et du titre :

```
if(sizeof($tabTypepd) > 1) {
    // saisie du 1er classId pour affichage du titre et de l'en - tête
    $classId1 = appelwsKtypnrFromTypepd($tabTypepd[0]['Type Id'], _COMPANY,
    _PASSWORD, _USER)->GetVtTcdTypeAndClassByAdcTypepdResult->TcdClassId;
    if($pays == "pt") {
        // affichage titre et en - tête BDD Portugal
        ?><caption>Ktypnr Véhicule léger (BDD Portugal)</caption><?
        afficheEnTeteHorizPaysClassId($pays, $classId1);
    } else if($pays == "fr" && $classId1 != 2) {
        // affichage titre et en - tête BDD France vl
        ?><caption>Ktypnr Véhicule léger (BDD France)</caption><?
        afficheEnTeteHorizPaysClassId($pays, $classId1);
    } else if($pays == "fr" && $classId1 == 2) {
        // affichage titre et en - tête BDD France pl
        ?><caption>Ntypnr Poids lourd (BDD France)</caption><?
        afficheEnTeteHorizPaysClassId($pays, $classId1);
    }
    $classId = array();
    for($i = 0; $i < sizeof($tabTypepd); $i++) {
        $classId[$i] = appelwsKtypnrFromTypepd($tabTypepd[$i]['Type Id'],
        _COMPANY, _PASSWORD, _USER)->GetVtTcdTypeAndClassByAdcTypepdResult-
        >TcdClassId;
    }
    // affichage des données de tous les véhicules avec tri
    if($vin != "") {
        afficheDonneesTypnrTab($tabTypepd, $classId, $tabKtypnr, $conn, $pays);
    }
} else {
    if($pays == "pt") {
        // affichage titre et en - tête BDD Portugal
        ?><caption>Ktypnr Véhicule léger (BDD Portugal)</caption><?
        afficheEnTeteHorizPaysClassId($pays, $classId);
    } else if($pays == "fr" && $classId != 2) {
```

```
// affichage titre et en - tête BDD France vl
?><caption>Ktypnr Véhicule léger (BDD France)</caption><?
afficheEnTeteHorizPaysClassId($pays, $classId);
} else if($pays == "fr" && $classId == 2) {
    // affichage titre et en - tête BDD France pl
    ?><caption>Ntypnr Poids lourd (BDD France)</caption><?
    afficheEnTeteHorizPaysClassId($pays, $classId);
}
if($vin != "") {
    afficheDonneesTypnr(1, $tabTypeld['Type Id'], $classId, $tabKtypnr['Ktypnr'],
$conn, $pays);
}
}
```

Pour le tableau du VIN Filter, il faut gérer plusieurs cas :

- Soit on n'a que des marques : on affiche toutes les marques et les autres cases sont vides (d'où la nécessité d'avoir des arrays vides).
- Soit on a une marque et des modèles : on affiche la marque et tous les modèles. Les autres cases sont laissées vides.
- Soit on a plusieurs versions : on affiche toutes les données.
- Soit on a un seul véhicule : on affiche toute ses données. Dans cette situation il n'y a pas de boucle.

S'il n'y a pas de résultat, le tableau reste vide et un message d'erreur s'affiche :

```
// tableau VINfilter
```

```
// si plusieurs marques : un tableau par marque
```

```
if(isset($tabMarque[0])) {
    ?><table class="tableVIN">
    <caption>Type id : <?php print_r($tabTypeld['Type Id']) ?></caption><?
    afficheEnTeteHoriz();
    if($vin != "") {
        $i = 1; // pour l'affichage du compteur
        foreach($tabMarque as $element) {
            ?><tr><td><? print_r($i); ?></td><? // pour l'affichage du compteur
            $i++;
            afficheDonneesHoriz($tabKtypnr);
            afficheDonneesHoriz($tabTypeld);
            afficheDonneesHoriz($element);
        }
    }
}
```

```

        afficheDonneesHoriz($tabModele);
        afficheDonneesHoriz($tabVersion);
        afficheDonneesHoriz($tabDetail);
        ?></tr><?

    }
}
?></table><?

} else if(isset($tabModele[0])) { // si plusieurs modèles : un tableau par modèle
    ?><table class="tableVIN">
    <caption>Type id : <?php print_r($tabTypeId['Type Id']) ?></caption><?
    afficheEnTeteHoriz();
    if($vin != "") {
        $i = 1; // pour l'affichage du compteur
        foreach($tabModele as $element) {
            ?><tr><td><? print_r($i); ?></td><? // pour l'affichage du compteur
            $i++;
            afficheDonneesHoriz($tabKtypnr);
            afficheDonneesHoriz($tabTypeId);
            afficheDonneesHoriz($tabMarque);
            afficheDonneesHoriz($element);
            afficheDonneesHoriz($tabVersion);
            afficheDonneesHoriz($tabDetail);
            ?></tr><?

        }
    }
    ?></table><?

} else if(isset($tabVersion[0])) { // si plusieurs versions : un tableau par version
    ?><table class="tableVIN">
    <caption>Type id : <?php print_r(count($tabTypeId)) ?> versions.</caption><?
    afficheEnTeteHoriz();
    if($vin != "") {
        $tabLigne = array();
        for($i = 0; $i < sizeof($tabVersion); $i++) {

```

```

        $tabLigne[] = array_merge($tabKtypnr[$i], $tabTypeId[$i], $tabMarque,
        $tabModele, $tabVersion[$i], $tabDetail[$i]);
    }
    array_multisort($tabLigne, SORT_DESC);
    $i = 1; // pour l'affichage du compteur
    foreach($tabLigne as $element) {
        ?><tr><td><? print_r($i); ?></td><?
        afficheDonneesHoriz($element);
        $i++;
        ?></tr><?
    }
}
?></table><?
} else { // un seul véhicule
    ?><table class="tableVIN">
    <caption>Type id : <?php print_r($tabTypeId['Type Id']) ?></caption><?
    afficheEnTeteHoriz();
    if($vin != "") {
        ?><tr><td><? print_r(1); ?></td><? // pour l'affichage du compteur
        afficheDonneesHoriz($tabKtypnr);
        afficheDonneesHoriz($tabTypeId);
        afficheDonneesHoriz($tabMarque);
        afficheDonneesHoriz($tabModele);
        afficheDonneesHoriz($tabVersion);
        afficheDonneesHoriz($tabDetail);
        ?></tr><?
    }
    ?></table><?
}

```