Anael Régie Sécurité
Module Webservice - Fonctions

Version V9R5

Informations de publication

Version : Anael RS V9R5

Date de publication : 4 septembre 2025

Code du document : GW-V9R5-WebService-Fonction

Table des matières

À propos de ce manuel

Ce document décrit la liste des fonctions disponibles dans le module Webservice pour Anael TT.

Les structures représentant une entité (fiche candidat, client, chantier, etc..) reprennent les rubriques des tables correspondantes PERPEPP, INSTIAP. C'est donc vers le document de descriptions des tables de la base ANAEL TT que vous devez en priorité consulter pour connaitre le format attendu des données.

Le format de date utilisé dans Anael est du type AAAAMMJJ. Dans la mesure du possible, les dates qui seront échangées dans les transmissions du webservice sont de format AAAA-MM-JJ.

Exemples de mise en œuvre du webservice.

Des exemples sont donnés pour permettre une mise en œuvre du module plus aisées par des techniciens. Les valeurs sont à actualiser en fonction du contenu de la base de façon à obtenir des résultats.

Par défaut, les requêtes sont attendues en mode GET, donc si dans la description de la fonction la méthode n'est pas précisée, c'est qu'il s'agit de la méthode GET. Les quelques requêtes en mode POST ou DEL sont clairement identifiées.

Parmi les réponses, ou les paramètres certains membres auront toujours la même signification. Leur description ci-dessous permettra ensuite à ce document de na pas être trop répétitif :

Code	Signification
SOC	Code société
AGE	Code agence
CLI	Code client
MAT	Matricule de personnel
CPO	Code postal
IND	Indice
QUA	Qualification
NOM	Nom de famille
PRE	Prénom

Public concerné

Tout client ayant acquis la licence d'utilisation pour le module webservice d'Anael TT et devant le mettre en œuvre.

Ce document technique s'adresse en priorité, aux :

- Développeurs
- Administrateur des systèmes d'information

Documents liés

Le paramétrage facultatif à mettre en place pour la sécurisation du webservice est expliqué dans la documentation suivante : Anael TT et RS V9R5 -Module Webservice-Securisation (Infor_AnaelTTRS_V9R5_Webservice Securité.pdf)

Vous pouvez trouver les documents dans la section relative à la documentation produit sur le portail Infor Support Portal, tel que décrit dans la section « Contacter Infor », page 9.

Organisation

Ce tableau décrit les chapitres de ce manuel :

Chapitre	Description
Chapitre 1	Les fonctions par domaine et leur exploitation
Chapitre 2	Les fonctions

Historique du document

Version	Date	Auteur	Contenu
1.0	Juillet 2025	R&D	Création du document à partir du document V9R4

Contacter Infor

Pour toute question sur les produits Infor, rendez-vous sur https://concierge.infor.com et créez un cas.

Si nous modifions ce document après la sortie du produit, nous en publierons une nouvelle version dans *Documentation Central*. Pour accéder à la documentation, suivez l'URL https://docs.infor.com/fr-fr/products .

Nous vous conseillons de consulter régulièrement ce site afin de disposer des dernières versions de la documentation.

Pour tout commentaire sur la documentation Infor, vous pouvez envoyer un courrier à l'adresse documentation@infor.com.

Chapitre 1 Fonctions par domaine

Mise en garde

L'ouverture en modification d'une fiche (intérimaire, client, etc..) dans l'application ANAEL donne la possibilité d'enregistrer la totalité des valeurs de la fiche par l'utilisateur au moment où il clique sur le bouton "OK" de la fiche.

Dans le cas où un temps important existe entre l'ouverture en édition d'une fiche et sa validation par OK (qui peut être considéré par l'utilisateur comme le moyen de refermer la fiche, sans avoir nécessairement de modification à valider) va entrainer la recopie de tous les rubriques de l'enregistrement en base de données aux valeurs qu'elles avaient lors de l'ouverture de la fenêtre, et donc perdre des modifications que le webservice aurait pu réaliser sur l'enregistrement entre-temps.

Une façon de diminuer cette éventualité consiste à ce que les utilisateurs économisent le temps d'affichage des fiches en mode modification, et ne pas refermer une fiche ouverte depuis un certain temps par "OK" s'ils n'ont aucun changement à valider.

Présentation des fonctions par domaine

Voici une vue d'ensemble des fonctions (présentées en ligne) du webservice par domaine (présenté en colonne). Certaines fonctions sont présentes dans plusieurs domaines, elles ne seront décrites qu'une seule fois.

Fonction	Personnel	Client	Commande	Relevés d'heures	Autre	Contrats	
actionco		X					Actions commerciales
agelst					X		Liste des agences
adresseClient		X					Adresses client
adresseCVMSiren					X		Adresses SIREN, CVM, groupes
adresseETT					X		Adresses ETT

Fonction	Personnel	Client	Commande	Relevés d'heures	Autre	Contrats	
adresseOrganisme					X		Adresses organismes
adressePersonnel	X						Adresses personnel
canddbl	X						Recherche de doublons candidat
candidat	X						Fiche candidat
candplan	X						Planning candidat
candtel	X						Recherche candidat par n° tel
carfam	X	X				X	Familles de caractéristiques
carserie	X	X				X	Série de valeurs d'une famille
carval	X	X				X	Valeur d'une caractéristique
cc16n	X						Recherche de candidats
cc45n		X					Recherche de clients
chantier		X					Chantier d'un client
client		X					Fiche client
clients		X					Liste de clients
combo					X		Liste de valeurs possibles
contacts		X					Contacts d'un client
contratpdf	X	X				X	Edition PDF d'un contrat
contrats	X	X				X	Recherche de contrats
couv		X					Couverture assurance-crédit
couvmode		X					Méthode de gestion assurance-crédit
ctrdet						X	Détails d'un contrat
ctrfin						X	Financier d'un contrat
ctrhis						X	Historique d'un contrat
ctrlst		X				X	Liste rapide de contrats
doc	X	X					Document

Fonction	Personnel	Client	Commande	Relevés d'heures	Autre	Contrats	
doclst	X	X					Liste de documents
ferie					X		Jour férié
feries					X		Liste des jours fériés
groupe		X					Liste des groupes client
insee					X		Recherche code insee
pcs	X						Liste des professions et catégories
perlib	X						Liste de personnels
Perqua	X						Qualifications d'un personnel
prospect		X					Fiche client / prospect
qualification	X						Recherche de qualification
quaper	X						Recherche de personnel selon qualification
valpos					X		Valeurs possibles d'une option
verif_param					X		Teste l'existence d'une valeur
wsini					X		Teste la présence des fichiers INI et la connexion.
wsparam					X		Teste quelques paramètres
wstestpdf					X		Vérifie les droits pour la production de pdf
wsdirect					X		Accès à tout enregistrement de la base.

Exploitation des fonctions

Les fonctions en méthode GET s'utilisent sous forme d'url formées comme ceci :

http://serveurfonction?parametre1=valeur1&parametre2=valeur2

Le nom de la fonction est séparé des paramètres attendus dans l'url par le caractère "?", chaque paramètre et sa valeur sont séparés du paramètre suivant par le séparateur "&"

Pour chaque paramètre : son nom suivi du signe égal et de sa valeur. Les valeurs des paramètres sont encodées conformément à l'encodage des url. Un aperçu des conversions de caractères est disponible ici : https://www.w3schools.com/tags/ref_urlencode.asp

Si la valeur d'un paramètre est une chaine vide, alors le paramètre peut être omis dans l'appel de la fonction sans toutefois provoquer d'erreur. Néanmoins, la présence du paramètre et de sa valeur vide sont plus explicites. L'ordre des paramètres n'est pas pris en compte.

"response" et "message"

Les réponses comportent très souvent les membres "response" et "message". Le premier prend la valeur "OK" lorsque rien n'est venu contrarier le déroulement de la fonction, même si le résultat d'une requête est vide. Il peut parfois être complété par une information négligeable, comme le nombre d'éléments dans le tableau des réponses. Le second, en cas de "response" = "KO" comporte un message permettant de mieux cerner la raison de l'échec de la fonction.

Format de date

Le format de date attendu en paramètre est AAAA-MM-JJ

Emploi des minuscules et des majuscules dans les noms de paramètre et de rubrique

D'une façon générale, les membres des structures en provenance du webservice sont en majuscule, pour reprendre la règle utilisée dans la description des tables d'ANAEL, dont souvent les noms de rubrique sont utilisés.

Dans les appels, les paramètres prennent le plus souvent des noms en minuscule.

Visualisation des réponses

Les réponses aux url sont le plus souvent des structures JSON. Dans la phase de découverte et de mise au point de vos outils utilisant le module webservice ANAEL TT RS, les outils suivants vous permettront de mieux appréhender le résultat obtenu :

Extension à Chrome :

JSON Viewer awesome

https://chrome.google.com/webstore/detail/json-viewer-awesome/iemadiahhbebdklepanmkjenfdebfpfe?hl=fr

Ce premier outil vous permet de visualiser confortablement le contenu d'un résultat d'une requête en méthode GET :

En remplacement de :

{ "response":"OK", "DBL":[{ "PERSOC":"001", "PERIND":"PRA", "PERMAT":"nnnnnnnnn", "PERNOM":"XXXXXXXX", "PERPRE":"XXXXXXXX", "PERNSS":"nnnnnnnnnnnnn",…

Talend API Tester

https://chrome.google.com/webstore/detail/talend-api-tester-free-ed/aejoelaoggembcahagimdiliamlcdmfm?hl=fr

Ce dernier va plus loin en permettant de créer un véritable dossier de tests unitaires avec des variables, préparer des jeux d'appels aussi bien en mode GET que POST ou DELETE.

Chapitre 2 Fonctions personnel

Vue d'ensemble des fonctions spécifiques pour la gestion de personnel

	Personnel	Client	Commande	Relevés d'heures	Autre	Contrats	
adressePersonnel	X						Adresses personnel
canddbl	X						Recherche de doublons candidat
candidat	X						Fiche candidat
candplan	X						Planning candidat
candtel	X						Recherche de candidat par n° tel
carfam	X	X				X	Familles de caractéristiques
carserie	X	X				X	Série de valeurs d'une famille
carval	X	X				X	Valeur d'une caractéristique
cc16n	X						Recherche de candidats
contratpdf	X	X				X	Edition PDF d'un contrat
contrats	X	X				X	Recherche de contrats
ctrdet						X	Détail d'un contrat
ctrfin						X	Financier d'un contrat
ctrhis						X	Historique d'un contrat
doc	X	X					Document
doclst	X	X					Liste de documents
pcs	X						Liste des professions et catégories
perlib	X						Liste de personnels
perqua	X						Qualifications d'un personnel
qualification	X						Recherche de qualification
quaper	X						Recherche de personnel selon qualification

Fonction adressePersonnel : lecture en méthode GET

La fonction adressePersonnel en méthode GET, permet de récupérer les adresses d'un personnel ou candidat (données du fichier ADRPERP)

L'URL est de la forme :
http://server//V1/adressePersonnel?soc=001&mat=001100001&ind=CAA&cod=001

Les paramètres sont :

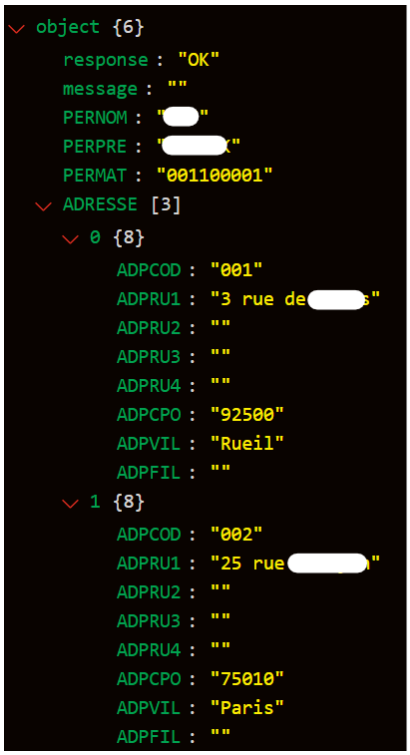
- soc pour le code société (obligatoire)
- ind pour l'indice (facultatif, par défaut PRA) : CAA pour un candidat.
- mat pour le matricule du personnel (obligatoire)
- cod : pour le code de l'adresse (facultatif)

Si le paramètre cod n'est pas précisé, la procédure récupère toutes les adresses du personnel.

Si le paramètre cod est précisé, seule l'adresse correspondant au code est récupérée.

Le résultat est une structure JSON comportant un tableau ADRESSE de structures. Chacune de ces structures comporte des membres correspondant aux rubriques de la table ADRPERP des enregistrements trouvés.

Structure de la réponse :



Fonction adressePersonnel : écriture en méthode POST

La fonction adressePersonnel en méthode POST permet d'enregistrer une nouvelle adresse pour un personnel ou de modifier une adresse existante.

Exemple d'appel :

http://host/V1/adressePersonnel?mode=A

Et présence de la structure de données dans le corps de la requête.

Paramètres :

- Le seul paramètre présent dans l'url est optionnel : "mode" qui peut prendre la valeur « A » (ajout) ou « M » (modification). Si le paramètre n'est pas précisé, il est par défaut à « A » (création d'une nouvelle adresse)

- En méthode POST, les données sont transmises sous forme d'une structure JSON présente dans le corps de la requête http. La structure attendue est identique à celle fournie par la fonction adressePersonnel en méthode GET.

  Exemple de structure JSON :

```
{
"ADPSOC": "001",
"ADPIND": "CAA",
"ADPMAT": "001101300",
"ADPCOD": "",
"ADPRU1": "56 RUE ALBERT EINSTEIN",
"ADPRU2": "",
"ADPRU3": "",
"ADPRU4": "",
"ADPCPO": "92500",
"ADPVIL": "RUEIL",
"ADPFIL": ""
}
```

Fonctionnement :

- En demande de création d'adresse, le code ADPCOD est laissé vide.
- En demande de modification, ADPCOD doit être renseigné

Réponse

La réponse est constituée d'une structure JSON comportant "response" avec une valeur KO en cas d'échec, et OK en cas de succès et un membre "message".

Fonction canddbl

La fonction canddbl permet, en amont de la création d'une nouvelle fiche candidat, de connaître la présence éventuelle de candidats dans la base ayant les mêmes nom ou numéro de sécurité sociale (recherche de doublons)

Méthode : GET

Les paramètres attendus sont :

Paramètre	Présence	Commentaire
soc	Obligatoire	Code société
nss	Nss est obligatoire si les nom et prénom sont absents, et réciproquement. Nom et prénom ont une taille minimum de 3 caractères, sinon la recherche ne se fait pas Nom (si renseigné) doit être saisi en entier (pour chercher un doublon, donc quelqu'un qui a le même nom exactement)	Numéro de sécurité sociale sur 13 caractères
nom		
pre		Peut prendre la valeur * pour indiquer que la recherche ne se fait pas sur le prénom

La recherche est réalisée sur le numéro de sécurité sociale s'il est transmis, et sur les noms et prénom. Le résultat est la somme des réponses trouvées dans les deux recherches.

La réponse est composée, outre "response" et éventuellement "message", d'un tableau "DBL" de structures. Chaque structure comporte quelques rubriques des fiches de PERPERP correspondantes :

Fonction candidat

La fonction candidat existe dans les méthodes GET et POST

Méthode	Effet	Transmission
GET	Récupère les valeurs d'une fiche personnel	Paramètre dans l'url, Structure JSON dans le corps de la réponse
POST	Enregistre une fiche personnel à partir des valeurs fournies	Structure JSON dans le corps de la requête

Particularités fonction candidat GET :

En méthode GET, l'url d'appel prend les paramètre mat (obligatoire) et ind, avec en valeur de mat le matricule du personnel recherché, comme dans cet exemple :

http://server/V1/candidat?mat=1300001

La paramètre ind est optionnel, peut prendre les valeurs "PRA", qui est sa valeur par défaut, pour spécifier que l'on recherche un personnel validé et non une fiche candidat que l'on rechercherait avec ind à la valeur "CAA".

En réponse, une structure JSON comportant un membre "response", éventuellement un membre "message", ainsi qu'une structure "PER", disposant d'une liste de membres correspondant aux rubriques de PERPEP, PERPERP2, une apparition, et 7 omissions :

Membre ajouté aux membres de PERPERP et PERPERP2 : PERPA2 correspondant au code ISO du pays de résidence.

Membres omis : DATCRE, DATMOD, DATSUP, AUTCRE, HEUCRE, AUTMOD, HEUMOD

Particularités fonction candidat POST :

La fonction candidat en méthode POST fonctionne pour créer une nouvelle fiche, et en modification de fiche existante.

Pour la création, il suffit de laisser le matricule PERMAT à blanc. Si la rubrique PERMAT est renseignée, alors il s'agit d'une demande de modification de la fiche du matricule.

La structure attendue dans le corps de la requête est identique en format au membre PER reçu en requête GET.

La réponse à un ajout est une structure JSON telle que :

```
{
response: 'OK',
message: 'Le candidat a été ajouté sous le matricule 130100430',
PERMAT: '130100430'
```

```
}
```

Ou :

```
{
response: 'OK',
message: 'Le candidat a été mis à jour'
PERMAT: '130100430'
}
```

Une détection de doublon peut fonctionner lors de l'import, par utilisation du paramètre homonym=true sur l'url de la requête POST. En cas de détection de doublon, la création n'a pas lieu. La détection se fait sur le n° de sécurité sociale, et la concaténation du nom et du prénom.

Si une homonymie est détectée, la réponse devient :

```
{
response: 'KO',
message: ' Homonyme NSS MAT=130100193'
PERMAT: ''
}
```

Les rubriques obligatoires

L'omission d'une rubrique obligatoire à la création de fiche ou à la mise à jour retourne KO et un message d'erreur.

Voici la liste des rubriques dont la présence est contrôlée :

PERAGE	Agence
PERTIP	Civilité
PERNOM	Nom
PERPRE	Prénom
PERRU1	Adresse ligne 1
PERPA2	Pays de résidence
PERCPO	Code Postal
PERPAY	Pays de naissance
PERNAT	Nationalité
PERVIL	Ville
PERLIE	Lieu de naissance
PERDEP	Département de naissance

PERDTN	Date de naissance
PERSIT	Situation de famille
PERNSS	Code NIR
PERNSC	Clé NIR
PERCOD	Piece d'identité + date de validité
PERQUA	Qualification
PERNOT	Niveau d'autonomie
PERCAT	Catégorie employé
PERMRE	Mode de règlement
PERFBU	Forme du bulletin
PERFDP	Fréquence du bulletin

Fonction candplan

La fonction candplan est en méthode GET, permet de disposer du planning des contrats d'un personnel sur une période. Si la période est omise, la recherche est réalisée depuis le 1er janvier à la date courante.

Les paramètres sont :

- soc pour le code société (obligatoire)
- mat pour le matricule du personnel (obligatoire)
- d1 pour la date de début d'intervalle (facultatif)
- d2 pour la date de fin d'intervalle (facultatif)

Particularité pour d1 et d2 : on peut recevoir des dates ou des semaines. En cas de transmission d'un numéro de semaine sous la forme AAAASS, on utilisera dans la recherche le premier jour de la semaine (lundi) pour d1 et le dernier jour de la semaine (dimanche).

Le résultat est une structure JSON comportant un tableau CTRLST de structures. Chacune de ces structures comporte des membres correspondant aux rubriques de la table ETTCTCP des enregistrements trouvés, plus deux libellés LIBQUA et LIBCLI pour présenter respectivement le libellé de la qualification, et le nom du client.

Structure de la réponse :

```
{
 CTRLST: [
  {
   CTCSOC: '001',
   CTCAGE: '130',
   CTCCTR: '103289',
   CTCAVE: '  ',
   CTCDEB: '2019-02-22',
   CTCFIN: '2019-02-22',
   CTCREL: '2019-02-22',
   CTCXFI: '2019-02-22',
   CTCXF2: '2019-02-22',
   CTCFMO: '32',
   CTCPER: '1301000061',
   CTCCHG: 'VRR',
   CTCCLI: '100014',
   CTCAFF: '100015',
   CTCQUA: 'CPLR01',
   CTCSAL: 10.03,
   CTCSNH: 1,
   CTCSTA: '',
   LIBQUA: 'CONDUCTEUR PL REGIONAL',
   LIBCLI: 'PHOENIX DISTRIBUTION'
  },
 ],
 response: 'OK',
}
```


Fonction cc16n

La fonction GET cc16n propose une recherche de personnel basée sur différents critères.

Liste des paramètres :

Paramètre	Présence	Description
soc	Obligatoire	Code société
age	Facultatif	Code agence
blo	Facultatif	B pour personnel bloqué, NB pour personnel non bloqué, absence du paramètre pour les deux catégories
nom	Facultatif	Les premières lettres du nom.
pre	Facultatif	Des lettres dans le prénom. Les performances sont maximale lorsqu'une partie du nom est renseignée.

mat	Facultatif	Matricule. Ce paramètre prend le pas sur les autres critères

La réponse est une structure JSON comportant "response" et un tableau de structure PER. Chaque structure est composée de rubriques de PERPERP, dont ils prennent les 3 derniers caractères du nom de la rubrique, et une rubrique JQUA portant le libellé de la qualification.


```
{
"response": "OK",
"PER": [
{
"SOC": "001",
"IND": "PRA",
"AGE": "350",
"PAG": "350",
"MAT": "350003651",
"NOM": "ATHIE",
"PRE": "ABOUBAKRY",
"CPO": "35200",
"TEL": "06 61 55 77 40",
"BLO": "",
"QUA": "MAN102",
"NSS": "1800799336033",
"LQUA": "MANUTENTIONNAIRE",
"NOT": "2",
"STX": "C"
},
{...}
]
}
```


Fonction candtel

La fonction de candtel permet une recherche de personnel à partir d'un numéro de téléphone. Le seul paramètre est un numéro de téléphone encodé URL.

Le résultat est une structure JSON comportant "response" et un tableau TEL de structures. Chacune comporte des membres correspondants des rubriques de PERPERP dont ils reprennent les 3 derniers caractères du nom de rubrique comme nom.


```
{
  TEL: [
    {
      soc: '001',
      age: '130',
      mat: '130100187',
      nom: 'BALMHI',
```

```
      pre: 'TAREK',
      tel: '07 83 23 35 23'
    }
  ],
  response: 'OK'
}
```


Fonctions Profils et Caractéristiques

Par "profils et caractéristiques" on entend la méthode d'Anael pour attribuer des paires clé+valeur à des entités parmi personnels, clients, contrats. Les clés sont formées d'un domaine, d'une famille et d'une caractéristique.

- Les domaines sont fixes : PER pour le personnel, CLI pour les clients et CTR pour les contrats.
- Les familles sont définies par paramétrage dans Anael
- Chaque famille connaît une liste de caractéristiques possibles, définis par paramétrage
- La valeur donnée à la caractéristique est libre, et comporte plusieurs rubriques.

Fonction carfam

Cette première fonction sur les profils et caractéristiques permet de disposer de la liste des familles dans un domaine.

Les paramètres sont :

- Le domaine "dom" avec une valeur parmi "PER", "CLI", "CTR", obligatoire,
- Le code société "soc", obligatoire
- Le code famille "fam", facultatif, permet de limiter la recherche à une seule famille

Exemple de requête : http://server/V1/carfam?soc=001&dom=PER&fam=

Exemple de réponse :


```
{
  response: 'OK',
  CARFAM: [
    { FAM: 'AA', LIB: 'AUTRES ADRESSES', VL: '' },
    { FAM: 'CO', LIB: 'COFFREO', VL: '' },
    { FAM: 'DE', LIB: 'DEMAT', VL: '' },
    { FAM: 'DI', LIB: 'DIPLOMES', VL: '' },
    { FAM: 'DS', LIB: 'DIVERS SOCIAL', VL: 'DO' },
    { FAM: 'DV', LIB: 'DIVERS', VL: '' },
```

```
        { FAM: 'EX', LIB: 'EXPERIENCE', VL: '' },
        { FAM: 'FO', LIB: 'FOURNITURES', VL: '' },
        { FAM: 'FP', LIB: 'FORMATION PROF.', VL: 'DO' },
        { FAM: 'KC', LIB: 'CACES', VL: 'DO' },
        { FAM: 'LG', LIB: 'LANGUES', VL: '' },
        { FAM: 'LO', LIB: 'LOGICIELS', VL: '' },
        { FAM: 'PA', LIB: 'PARTICULARITES', VL: '' },
        { FAM: 'PR', LIB: 'PERMIS', VL: 'DO' },
        { FAM: 'PX', LIB: 'PIXID', VL: '' },
        { FAM: 'VH', LIB: 'VEHICULE', VL: '' }
      ]
    }
```


Outre le membre "response", la réponse est constituée d'un tableau CARFAM de structures. Chacune comprend un membre "FAM" pour le code de la famille, "LIB" pour le libellé de la famille, et "VL" qui donne la valeur lue dans "PARVAL".

La présence de "DO" indique que des dates de début de validité et/ou de fin de validité (selon le contexte) sont attendues dans les valeurs d'une caractéristique, respectivement aux rubriques PERPRPP.PRPDT1 et PERPRPP.PRPDT2.

Fonction carserie

Cette fonction permet de recevoir la série de caractéristiques d'une famille d'un domaine.

Les paramètres :

- Le code société "soc" obligatoire
- Le domaine "dom" obligatoire parmi "PER", "CLI", "CTR",
- Le code famille "fam" facultatif. En cas d'omission, toutes les séries du domaine sont retournées
- "lib" pour une recherche d'une caractéristique sur un fragment de libellé.

  Exemple d'appel : http://host/V1/carserie?soc=001&dom=PER&fam=DI&lib=

  Exemple de retour :


```
{
  response: 'OK',
  CARSERIE: [
    { FAM: 'DI', CODE: 'BAC', LIB: 'BACCALAUREAT' },
    { FAM: 'DI', CODE: 'BAP', LIB: 'BAC PROFESSIONNEL' },
    { FAM: 'DI', CODE: 'BEP', LIB: 'BEP' },
    { FAM: 'DI', CODE: 'BEPC', LIB: 'B.E.P.C' },
    { FAM: 'DI', CODE: 'BRC', LIB: 'BREVET COMPAGNON' },
    { FAM: 'DI', CODE: 'BTN', LIB: 'BREVET TECHNICIEN' },
    { FAM: 'DI', CODE: 'BTS', LIB: 'B.T.S' },
    { FAM: 'DI', CODE: 'CAP', LIB: 'C.A.P' },
    { FAM: 'DI', CODE: 'CCA', LIB: 'CERT DE CAPACITE AMBULANCIER' },
    { FAM: 'DI', CODE: 'DEA', LIB: 'D.E.A.' },
```

```
        { FAM: 'DI', CODE: 'DES', LIB: 'D.E.S.S.' },
        { FAM: 'DI', CODE: 'DEU', LIB: 'D.E.U.G.' },
        { FAM: 'DI', CODE: 'DUT', LIB: 'D.U.T' },
        { FAM: 'DI', CODE: 'ETA', LIB: 'DIPLOME' },
        { FAM: 'DI', CODE: 'FOR', LIB: 'FORMATION' },
        { FAM: 'DI', CODE: 'ING', LIB: 'INGENIEUR' },
        { FAM: 'DI', CODE: 'LIC', LIB: 'LICENCE' },
        { FAM: 'DI', CODE: 'MAI', LIB: 'MAITRISE' },
        { FAM: 'DI', CODE: 'RQTH', LIB: 'RQTH' },
        { FAM: 'DI', CODE: 'SST', LIB: 'CERTIFICAT DE SAUVETEUR SECOUR' },
        { FAM: 'DI', CODE: 'XX', LIB: 'AUTRE' }
      ]
    }
```

Une structure JSON comportant "response" et un tableau "CARSERIE" de structures. Chacune comprend le code famille "FAM", le code de la caractéristiques "CODE", et "LIB" pour le libellé de cette caractéristique.

Fonction carval méthode GET

Les deux premières fonctions permettent de connaître le paramétrage d'Anael : les familles dans les domaines, les caractéristiques par famille. La fonction carval permet de lire la partie "valeur" du système clé-valeur des profils et caractéristiques.

La fonction connaît deux méthodes :

- GET pour la lecture d'une valeur,
- POST pour l'ajout, la modification, la suppression d'une caractéristique.

  Exemple d'appel GET : http://host/V1/carval?soc=001&age=130&dom=PER&mat=130100193

Les paramètres :

- Le code société "soc" obligatoire
- Le domaine "dom" parmi "PER", "CLI", "CTR","CT1","CTD" : obligatoire,
- Le code agence "age" obligatoire
- Le code "mat" attend un matricule lorsque l'entité est un personnel, un code client lorsque le domaine est "CLI", un numéro de contrat pour le domaine "CTR", un numéro de modèle pour le domaine "CT1", un numéro de commande + numéro ligne pour le domaine "CTD",
- Le code de famille "fam", facultatif
- Le code de caractéristique "serie" est facultatif

La réponse est une structure JSON comportant le membre "response" et un tableau CAR de structures qui reprennent SOC, AGE IND, MAT pouvant être en fonction du domaine un matricule, un code client ou un numéro de contrat, un numéro de modèle ou une ligne de commande, FAM et FAMLIB pour le code et le libellé de la famille, SERIE et SERIELIB pour respectivement le code et le libellé de la caractéristique.

Ensuite, viennent les valeurs proprement dites, c'est-à-dire les rubriques dans lesquelles on peut inscrire de l'information :

- COM1 qui représente le contenu de la rubrique PERPRPP.PRPCOM
- COM2 pour le contenu de PERPRPP.PRPCO2
- DT1 pour celui de PERPRPP.PRPDT1
- DT1 pour celui de PERPRPP.PRPDT2

```
{
  response: 'OK',
  CAR: [
    {
      SOC: '001',
      AGE: '130',
      IND: 'PER',
      MAT: '130100193',
      FAM: 'DE',
      FAMLIB: 'DEMAT',
      SERIE: 'CONT',
      SERIELIB: 'DEMAT CONTRAT',
      COM1: '',
      COM2: '',
      DT1: '',
      DT2: ''
    },
```

Fonction carval méthode POST

L'utilisation de la fonction carval en méthode post permet de manipuler une caractéristique. En méthode POST les paramètres ne sont plus transmis dans l'url, mais dans une structure JSON dans le corps de la requête.

Il n'y a pas de contrôle d'unicité : il est possible d'enregistrer plusieurs fois la même caractéristique avec les mêmes valeurs.

Pour être certain d'appliquer votre action de modification ou de suppression à une caractéristique bien précise, des instructions MX et SX utilisent les valeurs de COM1 et COM2 comme clé de recherche. Les valeurs recherchées sont alors transmises dans les membres XCOM1 et XCOM2.

En fonction de la valeur de la rubrique "verb", cette fonction va produire :

"verb"	Action
A	Ajout d'une caractéristique
M	Modification d'une caractéristique.

	On modifie la première occurrence trouvée
MX	Modification d'une caractéristique recherchée à l'identique des rubriques de commentaire : PERPRPP.PRPCOM doit être identique à XCOM1 PERPRPP.PRPCO2 doit être identique à XCOM2
S	Suppression d'une caractéristique. On supprime la première occurrence trouvée
SX	Suppression d'une caractéristique recherchée à l'identique des rubriques de commentaire : PERPRPP.PRPCOM doit être identique à XCOM1 PERPRPP.PRPCO2 doit être identique à XCOM2

Le format attendu dans de cette structure est le suivant :

```
{
    verb: 'SX',
    SOC: '001',
    IND: 'PER',
    AGE: '130',
    MAT: '130100193',
    FAM: 'DS',
    SERIE: 'CVI',
    COM1: 'test 854',
    COM2: '',
    DT1: '',
    DT2: '',
    XCOM1: 'test 1',
    XCOM2: ''
}
```

```
Verb : indique l'action à réaliser


Champs utilisés pour la recherche de
la caractéristique : SOC pour le code
société, AGE pour le code agence, IND
pour le domaine, MAT pour le code de
l'entité dans son domaine.


Les valeurs possibles de IND sont
"PER", "CLI", "CTR", "CT1", "CTD"
respectivement pour personnel,
client, contrat, modèle, commande
```

Les rubriques SOC, IND, AGE, MAT FAM, SERIE, COM1, COM2, DT1, DT2 ont la même signification que dans la structure reçue de la fonction carval en méthode GET.

Les membres XCOM1 et XCOM2 sont utilisés quand "verb" prend les valeurs MX ou SX, pour retrouver l'occurrence recherchée.

Le format de date attendu dans DT1 et DT2 est "AAAA-MM-JJ"

En retour, une structure JSON comprenant "response" et "message", le premier ayant la valeur "OK" en cas de succès, "KO" en cas d'échec, et dans ce cas "message" contient une indication des raisons de l'échec d'exécution de la fonction.

Fonction doclst

La fonction doclst présente la liste des documents disponibles pour une entité.

Exemple d'appel : http://host/V1/doclst?soc=001&ind=PRA&age=130&mat=130100003

Paramètres :

- Code société "soc" obligatoire
- Code agence "age" obligatoire

Pour un document lié à :

- Un personnel :
  - Code "ind" avec une valeur "PER" ou "PRA", obligatoire
  - Code "mat" avec le matricule du personnel, obligatoire
- Un client
  - Code "ind" à "CLI" ou "CL1", obligatoire
  - Code "mat" avec le code client
- Un chantier
  - Code "ind" à "CHA" ou "CL5", obligatoire
  - Code "mat" avec le code chantier
- Une commande
  - Code "ind" à "CMD" ou "CDE", obligatoire
  - Code "mat" avec le code de la commande

Le retour est une structure JSON comportant le membre "response" et un tableau DOC de structures. Chacune représente un document théoriquement disponible en redonnant les informations SOC, IND, AGE, MAT de la sélection, LIB comprenant le nom du document et LOC la localisation du document.


```
{
   "response" : "OK",
   "message" : "",
   "DOC" :
   [
      {
         "SOC" : "001",
         "AGE" : "001",
         "IND" : "PRA",
         "MAT" : "001100170",
         "LIB" : "pdftest3.pdf",
         "LOC" : "C:\\DATAS\\pdftest3.pdf"
      }
   ]
}
```

Fonction doc en méthode GET

La fonction en méthode GET utilise les informations d'un document connu pour fournir en retour ce document sous sa forme base64.

Exemple d'appel :

http://hostdoc?soc=001&ind=PRA&age=130&mat=130100003&lib=pl%20CI%20et%20C.VITALE%20verso.PDF

Paramètres :

- Le code société "SOC" obligatoire
- Le code agence "AGE", obligatoire
- Les codes "IND" obligatoire et "MAT" obligatoire, ainsi :
  - Pour le document d'un personnel :
    - Ind = "PER" ou "PRA", et un matricule pour "mat"
  - Pour le document d'un client :
    - Ind = "CLI" ou "CL1", et un code client pour " mat "
  - Pour le document d'un chantier :
    - Ind = "CHA" ou "CL5", et un code chantier pour " mat "
  - Pour le document d'une commande :
    - Ind = "CMD" ou "DE", et un code commande pour " mat "
- "LIB", obligatoire, encodé URL, contenant le nom du fichier tel que reçu dans la rubrique LIB de la fonction DOCLST.

En réponse une structure JSON comportant un membre "response" et un membre "base64". Ce dernier détient une chaine base64 du document PDF.

Lorsque le fichier demandé n'existe pas, un fichier générique est envoyé en remplacement, et la réponse est `'OK document pour test'.`

Fonction doc en méthode POST

La version méthode POST de la fonction "doc" permet d'enregistrer un document. Certains paramètres sont attendus dans l'url, et le document sous sa forme base64 est attendu dans le corps de la requête.

Les paramètres attendus dans l'url :

- Le code société "soc" obligatoire,
- Le code agence "age" obligatoire
- Le type de document "ind"
  - Pour le document d'un personnel :
    - Ind = "PER" ou "PRA", et un matricule pour "mat"
  - Pour le document d'un client :
    - Ind = "CLI" ou "CL1", et un code client pour "mat"

  - Pour le document d'un chantier :
    - Ind = "CHA" ou "CL5", et un code chantier pour "mat"
  - Pour le document d'une commande :
    - Ind = "CMD" ou "DE", et un code commande pour "mat"

- Le drapeau de remplacement "replace" mis à "true" ou "false" pour indiquer que l'action demandée consiste à remplacer un fichier existant
- Le nom du fichier "file" encodé URL
- Le répertoire de destination du document "repdoc" (facultatif).
  Si ce paramètre n'est pas renseigné, le document est créé dans l'emplacement désigné dans ANAELTT.INI pour la société de connexion, pour la valeur "DOCUMENTS"

Attendu dans le corps de la requête :

- Le contenu du fichier sous forme d'une chaine codée Base64 dans une valeur "file"

```
{"file":'JVBERi0xLjQKJeTjz9IKMyAwIG9iago8PC9UeXBlIC9QYWdl….
```

En réponse, une structure JSON comportant "response" et "message" pour savoir si la fonction a pu d'exécuter correctement ou non.

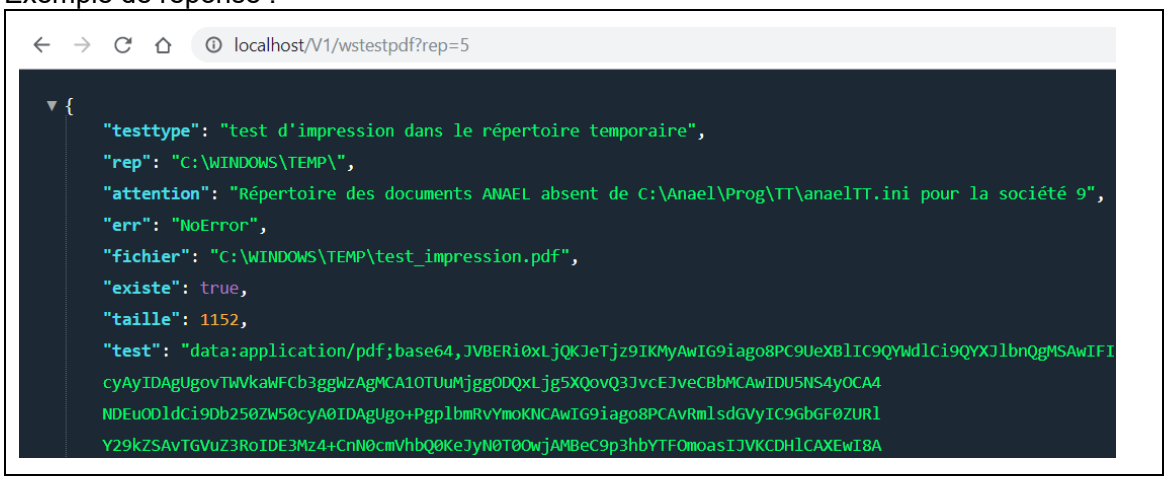
Une des raisons possibles d'un échec peut venir de l'impossibilité pour le compte utilisé par le webservice d'écrire dans le répertoire des documents.

Pour tester utilisez l'url suivante :

http://server/V1/wstestpdf?rep=5

Le paramètre rep=5 permet de réaliser un test de création d'un fichier pdf de test dans l'emplacement désigné dans ANAELTT.INI pour la société de connexion, pour la valeur "DOCUMENTS".

Exemple de réponse :



Fonction pcs

La fonction pcs en méthode GET doit permettre de retrouver des codes PCS (selon la nomenclature des professions et des catégories de l'INSEE)

Exemple d'appel : http://host/V1/pcs?soc=001&cod=&lib=

Paramètres :

- Code société "soc" obligatoire
- Début d'un code PCS ou code PCS "cod", facultatif. Cela permet de retrouver des codes connexes à un code déjà connu.
- Fragment du libellé 'lib' facultatif, encodé URL.

Réponse

La réponse est une structure JSON comportant les membres "response" et "message" et un tableau PCS de structures JSON.

Chaque structure comporte les membres "COD" pour le code de la PCS, et "LIB" pour le nom complet de cette PCS.

Fonction qualification

La fonction en méthode GET qualification permet de réaliser une recherche parmi les qualifications.

Exemple d'appel : http://host/V1/qualification?soc=001&cod=&lib=&fam=&pcs=551a

Paramètres

- Code société "soc" obligatoire
- Codes qualification, libellé, PCS : Il est obligatoire qu'au moins un de ces paramètres soit présent.
    - Code qualification : il peut être partiel. Dans ce cas, la recherche portera sur tous les codes de qualification commençant par la portion de code fournie.
    - Libellé : fragment de chaine à retrouver dans le libellé. Par défaut la recherche portera sur les libellés commençant par la chaine fournie. Cette chaine est codée URL. Si la chaine commence par le caractère *, alors la recherche portera sur tous les libellés comprenant cette chaine.
    - Code PCS : Comme pour le code de qualification, on peut transmettre un code complet ou le début d'un code.
- Le code famille FAM, facultatif. S'il est présent, il permettra de filtrer à l'identique le code de famille de qualification présent dans la rubrique QA1FAM.

Le résultat est une structure JSON comprenant les membres "response" et "message" ainsi qu'un tableau QUALIF de structures. Chacune reprend certaines rubriques de la table PPYQA1P. les

noms des membres correspondent au 3 derniers caractères des rubriques de cette tables pour les enregistrements transmis. Les rubriques QA1ACT et QA1ACS sont concaténées dans le membre "ACT".

Fonction contratpdf : Edition du contrat en PDF

La fonction contratpdf permet de disposer de la version PDF de l'édition d'un contrat, dans l'une de ses versions, dont l'exemplaire personnel.

Exemple d'url : http://host/V1/contratpdf?soc=001&age=130&ctr=100012&ave=&type=1

Les paramètres :

- La société "soc" obligatoire
- L'agence "age" obligatoire
- Le numéro de contrat "ctr" obligatoire
- Le numéro d'avenant "ave" facultatif
- La version de l'édition contrat : (facultative, prend la valeur "2" en cas d'omission)
    - "1" pour l'exemplaire agence
    - "2" pour l'exemplaire personnel
    - "3" pour l'exemplaire client
    - "4" pour l'exemplaire personnel avec le logo
    - "5" pour l'exemplaire client avec logo

Pour que les exemplaires avec logo donnent une représentation satisfaisante, vous devez vous assurer que le répertoire "client" d'ANAEL, emplacement attendu en " …\prog\TT\client " soit accessible. Vous disposez également d'une fonction de test des conditions d'exécution, la fonction de diagnostic wscontratpdf, invoquée comme dans cet exemple :

http://server/V1/wscontratpdf?soc=001&age=130&demat=1

La paire des paramètres "soc" et "age", compte-tenu du paramétrage permettent de composer le nom du fichier attendu, et savoir s'il contient par exemple le code de société de gestion.

Cette fonction prend en paramètre obligatoires "soc" pour la société, "age" pour l'agence cible, le numéro de contrat "ctr" et "demat" à 1 pour indiquer que l'on souhaite un fichier .def particulier, ce qui est le cas lorsque la version de contrat est de type 4 ou 5.

Dans ces modes type = 4 ou 5, on ne prend pas le fichier .def par défaut pour la combinaison (soc + age), mais d'un fichier .def spécifique, dont le nom est contient "_DEM" . Ce fichier peut par exemple comprendre un logo spécifique pour la production de PDF.

La réponse est de ce type :

On identifie le fichier .def désigné par le contexte société et agence, on vérifie qu'il soit bien présent. S'il est présent, on recherche le fichier de logo qui y est fait référence, et ton vérifie à son tour qu'il soit bien présent.

Une impression une production PDF correcte sera obtenue lorsque le fichier .def recherché sera trouvé, ainsi que le fichier de logo qui y est fait référence.

Pour rappel : La visualisation dans un navigateur d'un document PDF sous forme de chaine base64 s'obtient en plaçant la chaine base64 dans l'url précédé de " data:application/pdf;base64"

Fonction contrats : liste de contrats.

La fonction contrats s'utilise en méthode GET pour disposer d'une liste de contrats correspondant aux critères de recherche transmis.

Exemple d'appel : http://host/V1/contrats?soc=001&age=130&ctr=&cli=&aff=&per=&xfil1=&xfil2=&deb1=2019-04-01&dfin=&encours=true

Liste des paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- Les paramètres facultatifs :
  - Le code client "cli"
  - Si le code client est présent, le code chantier "aff" peut être présent
  - Le matricule du personnel "per"
  - La date minimum de fin de contrat "xfi1" et la date maximum de fin de contrat "xfi2 pour un contrat en cours. Si "xfi1" et "xfi2" sont fournis ensembles, alors ils représentent les bornes dans lesquels doit se situer ETTCTCP.CTCXFI

- La date minimale de début de contrat "deb1" et la date maximale de début de contrat "deb2". "deb1" et "deb"2" transmis ensemble forme l'intervalle dans lequel doit se situer ETTCTCP.CTCDEB.
- Si "encours" est présent à la valeur "true" alors le résultat est limité aux contrats en cours.

La réponse est une structure JSON comportant un membre "response", et un tableau de structures. Chacune des structures représente un enregistrement de ETTCTCP dont vous retrouvez les noms des rubriques.

Fonction CTRDET : détails d'un contrat

La fonction CTRDET s'utilise en méthode GET pour disposer des détails (commentaires) d'un contrat (données de la table ETTCTDP)

Exemple d'appel : http://host/V1/ctrdet?soc=001&age=001&ctr=100001

Liste des paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- Le code contrat "ctr" obligatoire

La réponse est une structure JSON comportant un membre "response", et les rubriques de l'enregistrement ETTCTDP correspondant au contrat passé en paramètre

Extrait d'un résultat :


```
{
  "response": "OK",
  "message": "",
  "CTDSOC": "001",
  "CTDIND": "CTR",
  "CTDAGE": "001",
  "CTDCTR": "100001",
  "CTDAVE": "",
  "CTDCOD": "",
  "CTDAVN": "01",
  "CTDAVT": "P",
  "CTDAVD": "20141220",
  "CTDARC": "N",
  "CTDARP": "N",
  "CTDTX1": "COMMANDE A FINIR DANS LES DELAIS",
  "CTDTX2": "",
```

Fonction CTRFIN : financier d'un contrat

La fonction CTRFIN s'utilise en méthode GET pour disposer des lignes de financier d'un contrat (données de la table ETTCTEP)

Exemple d'appel : http://host/V1/ctrfin?soc=001&age=001&ctr=100001

Liste des paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- Le code contrat "ctr" obligatoire
- Le code rubrique "rub" facultatif (si on ne veut récupérer qu'une rubrique)

La réponse est une structure JSON comportant un membre "response", et un tableau de structures. Chacune des structures représente un enregistrement de ETTCTEP dont vous retrouvez les noms des rubriques pour le contrat passé en paramètre

Extrait d'un résultat :


```
{
  "FINANCIER": [
    {
      "CTESOC": "001",
      "CTEIND": "CTR",
      "CTEAGE": "001",
      "CTECTR": "100001",
      "CTEAVE": "0",
      "CTEDEB": "",
      "CTEFIN": "99991231",
      "CTEJOD": "0",
      "CTEJOF": "7",
      "CTERUB": "001",
      "CTEBAS": "0",
      "CTETXP": "12",
      "CTETXF": "23.64",
      "CTEMTP": "0",
      "CTEACT": "",
      "CTEDCR": "20141217",
      "CTEDMO": "20141217",
      "CTEDSU": "",
      "CTEEXL": "999"
    },
```

Fonction CTRHIS : historique d’un contrat

La fonction CTRHIS s'utilise en méthode GET pour disposer des lignes d’historique d’un contrat (données de la table XEDCTCP)

Exemple d'appel : http://host/V1/ctrhis?soc=001&age=001&ctr=100001

Liste des paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- Le code contrat "ctr" obligatoire
- Le code avenant "ave" facultatif (si on ne veut récupérer qu’un historique précis)

La réponse est une structure JSON comportant un membre "response", et un tableau de structures. Chacune des structures représente un enregistrement de XEDCTCP (historique) dont vous retrouvez les noms des rubriques pour le contrat passé en paramètre

Extrait d’un résultat :


```
{
  "HISTO": [
    {
      "XEDSOC": "001",
      "XEDIND": "CTR",
      "XEDAGE": "001",
      "XEDCLI": "100002",
      "XEDAFF": "100002",
      "XEDMAN": ".78513",
      "XEDSUC": "",
      "XEDCHG": "MTI",
      "XEDCH2": "",
      "XEDCPC": "0",
      "XEDCDE": "0",
      "XEDLIG": "",
      "XEDSLI": "",
      "XEDPAG": "001",
      "XEDPER": "001100001",
      "XEDCTR": "100001",
      "XEDAVE": "01",
      "XEDQUA": "SOU001",
      "XEDPON": "",
```

Fonction CTRLST : une autre liste de contrat

Cette fonction est plutôt dédiée au besoin d'afficher rapidement une liste de contrats, comme remplir une liste dans une page web.

Exemple d'appel :

http://host/V1/ctrlst?soc=001&age=130&ctr=&cli=&aff=&mat=&fmo=&deb1=&dfin=2019-06-17

Liste des paramètres :

- Le code société "soc", obligatoire
- Le code agence "age" obligatoire
- Le code client "cli", optionnel
- Le code chantier "aff", optionnel
- Le matricule du salarié "mat", optionnel
- Le code du contrat "ctr" optionnel

Cette fonction retourne une structure JSON compotant le membre "response" et un tableau CTR de structures. Chacune dispose de quelques rubriques issues de CTTCTCP dont elles reprennent certains membres, deux libellés :

- CTCRSO pour la raison sociale du client
- CTCNOM pour la concaténation du nom et du prénom du salarié.

Et le calcul du nombre de journée de la mission :

- NBJ pour le nombre de jours du contrat

Fonction perlib : Le personnel par qualification

La fonction perlib est une fonction de recherche de personnel par sa qualification principale

Exemple d'appel : http://host/V1/perlib?soc=001&age=130&blo=N&qua=CSPLR1&ind=

Les paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- Le type de blocage "blo" facultatif :
  - Les personnels non bloqués avec la valeur N
  - Les personnels bloqués avec la valeur B
  - Tous les personnels en omettant le paramètre
- Le code de qualification "qua", obligatoire, il a une taille minimale de 3 et maximale de 6 caractères. Ceci permet de réaliser des recherches sur des qualifications voisines.
- L'indice ind qui peut être omis pour que la requête s'applique aussi bien aux candidats qu'aux personnels ayant déjà eu un contrat, ou mis à la valeur "PRA" pour des personnels ou "CAA" pour des candidats

Le retour :

Le retour de la fonction est une structure JSON comportant un membre "response" et un tableau "PERLIB" de structures. Chacune reprend quelques rubriques de PERPERP, dont leur nom respectif est réduit aux 3 derniers caractères :

- SOC pour le code société
- AGE pour le code agence
- MAT pour le matricule
- NOM pour le nom
- PRE pour le prénom
- QUA pour le code de qualification
- TEL pour le numéro de téléphone

Fonction perqua GET : Qualifications d'un personnel

La fonction perqua en méthode GET permet de disposer de la liste des qualifications d'un personnel.

Exemple d'appel : http://host/V1/perqua?soc=001&mat=130100194

Les paramètres :

- Code société "soc" obligatoire
- Matricule "mat" obligatoire

Le retour

Le retour de la fonction est une structure JSON comportant un membre "response", le message éventuel de défaut "message", les nom et prénom de l'intérimaire "NON", "PRE", et un tableau de structures PERQUA qui reprend chacune des qualifications principales et secondaires :

- PRQLIB pour le libellé de la qualification
- PRQSOC pour le code société
- PRQIND pour l'indice avec la valeur "PRA" ou "CAA"
- PRQAGE pour le code agence
- PRQPER pour le matricule
- PRQQUA pour le code de la qualification
- PRQCOM indique avec "QUALIFICATION PRINCIPALE" s'il s'agit de la qualification trouvée dans sa fiche ou non. Dans ce cas, les membres PRQCO2, PRQDT1, PRQDT2 sont vides, et PRQPRI est à "true". Dans le cas où il ne s'agit pas de la qualification principale, alors c'est le contenu de PERPRQP.PRQCOM
- PRQCO2 : contenu de PERPRQP.PRQCO2 pour une qualification secondaire
- PRQDT1 : contenu de PERPRQP. PRQDT1pour une qualification secondaire
- PRQDT2 : contenu de PERPRQP. PRQDT2pour une qualification secondaire,
- PRQPRI : indique "true" pour la qualification principale, et "false" pour les autres
- PRQNOT : contenu de PERPERP.PERNOT pour la qualification principale, et de PERPRQP.PRQNOT pour les autres. (Degré d'autonomie dans le poste)

Fonction perqua POST : Inscrire une qualification

La fonction perqua en méthode POST permet d'associer une qualification à un personnel. Cette qualification sera ajoutée en qualification secondaire.

C'est aussi le moyen de modifier les informations associées à une qualification : les rubriques com, co2, dt1 et dt2 viendront remplacer les valeurs d'une qualification qui serait déjà présente pour le personnel.

Exemple d'appel :

`http://host/V1/perqua?soc=001&mat=130100194&qua=COR002&com=com&co2=co2&dt1=dt1&dt2=dt2&not=2`

Liste des paramètres :

- Le code société "soc" obligatoire
- Le matricule du personnel "mat" obligatoire
- Le code de qualification "qua" obligatoire
- Les zones de commentaires "com" et "co2" facultatives, stockées respectivement dans les rubriques PERPRQP.PRQCOM et PERPRQP.PRQCO2
- Les zone de date dt1 et dt2 facultatives, stockées respectivement dans PERPRQP.PRQDT1 et PERPRQP.PRQDT1
- Le niveau d'autonomie pour le poste "not", facultatif, stocké dans PERPRQP.PRQNOT

La réponse est une structure JSON comportant les membres "response" avec une valeur "KO" en cas d'échec ou "OK" en cas de succès, et "messages" indiquant des informations sur le traitement.

En fonction de l'action, la réponse indiquant les conditions d'exécution peut prendre la forme :

Action	Réponse
Ajout d'une qualification (succès)	`{ response: 'OK', message: 'Insert OK' }`
Modification d'une qualification (succès)	`{ response: 'OK', message: 'Update OK' }`
Exemple de réponse d'une requête en échec	`{ response: 'KO', message: 'Personnel inconnu' } ou` `{ response: 'KO', message: 'Qualification inconnue' }, etc`

Fonction perqua DEL : Suppression de qualification

La fonction perqua en méthode DELETE permet de supprimer une qualification secondaire.

Exemple d'appel : http://host/V1/perqua?soc=001&mat=130100194&qua=COR002

Les paramètres :

- Le code société "soc" obligatoire
- Le matricule du personnel "mat" obligatoire
- Le code de qualification "qua" obligatoire

La réponse est une structure JSON comportant les membres "response" et "message". En cas de succès, elle prend cette forme :

```
{ response: 'OK', message='Deleted' }
```

En cas d'échec, "message" contient une indication sur l'origine de cet état.

Fonction quaper : recherche candidat / qualification

La fonction quaper permet de rechercher des candidats sur une qualification particulière. La recherche porte sur la qualification principale et les qualifications secondaires. La fonction s'utilise en méthode GET.

Exemple d'appel :

http://host/V1/quaper?soc=001&age=130&ind=&qua=CSPL01%2CAR002&pcs=&dispo=

Paramètres :

- Code société "soc" obligatoire
- Code agence "age" obligatoire
- Liste des qualifications séparées par des virgules, et encodée URL (les virgules sont remplacées par %2) facultatif si présence de PCS
- Liste des PCS, séparés par des virgules et encodés URL facultatif si présence de qualifications
- La date de disponibilité "dispo" facultative
- L'indice "ind" facultatif, avec la valeur "PRA" ou "CAA"

Description de la réponse :

Outre le compte-rendu d'exécution (les membres "response" et "message"), la réponse comprend un tableau QUAPER de personnels, chacun ayant un tableau QUALIFS de ses différentes qualifications.

Pour chaque personnel on retrouve les membres

- MAT pour le matricule,
- NOM pour le nom de famille
- PRE pour le prénom,
- TEL pour le téléphone.
- Un tableau QUALIFS de structures comprenant :
    - Un code de qualification PRQQUA,
    - Le libellé de la qualification PRQLIB,
    - Le commentaire PRQCOM, qui prend la valeur "QUALIFICATION PRINCIPALE" lorsque c'est le cas.
    - Le degré d'autonomie dans le poste PRQNOT.

Chapitre 3 Fonctions Client

	Personnel	Client	Commande	Relevés d'heures	Autre	Contrats	
/V1/actionco		X					Actions commerciales
/V1/adresseClient		X					Adresses client
/V1/carfam	X	X				X	Familles de caractéristiques
/V1/carserie	X	X				X	Série de valeurs d'une famille
/V1/carval	X	X				X	Valeur d'une caractéristique
/V1/cc45n		X					Recherche de clients
/V1/chantier		X					Chantier d'un client
/V1/client		X					Fiche client
/V1/clients		X					Liste de clients
/V1/contacts		X					Contacts client
/V1/contratpdf	X	X				X	Edition PDF d'un contrat
/V1/contrats	X	X				X	Recherche de contrats
/V1/couv		X					Couverture assurance-crédit
/V1/couvmode		X					Méthode de gestion assurance-crédit
/V1/ctrlst		X				X	Liste rapide de contrats
/V1/doc	X	X					Document
/V1/doclst	X	X					Liste de documents
/V1/groupe		X					Liste des groupes client

Voici un tableau récapitulatif des fonctions qui concernent le domaine client. Les fonctions en grisées ont déjà été décrites précédemment : elles ne seront pas décrites dans ce chapitre.

Fonction actionco : les actions commerciales

La fonction actionco est la fonction de requête des actions commerciales. Cette fonction s'utilise en méthode GET.

Exemple d'appel :

http://host/V1/actionco?soc=001&age=130&cli=100056&chg=FMI&d1=&d2=

Les paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- Le code client "cli" facultatif si le paramètre "chg" est fourni, obligatoire sinon
- Le code chargé d'affaire "chg" facultatif si le paramètre "cli" est fourni, obligatoire sinon
- Les l'intervalle de recherche formé par les dates "d1" et "d2", d1 et d2 sont facultatifs

Le résultat est une structure JSON comportant outre les membres "response" et "message", un tableau ACTCO de structures reprenant comme membres les rubriques de la table INSVISP pour les enregistrements correspondant à la recherche.

Fonction adresseClient : lecture en méthode GET

La fonction adresseClient en méthode GET, permet de récupérer les adresses d'un client ou chantier (données du fichier ADRTIAP)

L'URL est de la forme : http://server//V1/adresseClient?soc=001&cli=100004&age=001

Les paramètres sont :

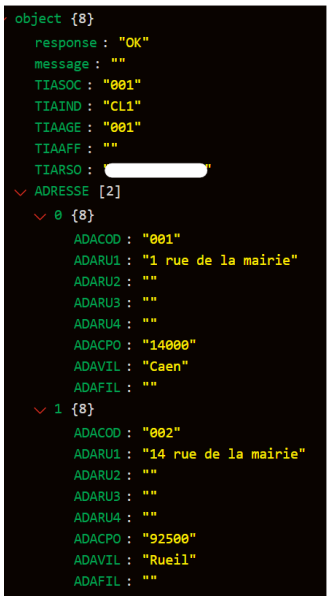
- soc pour le code société (obligatoire)
- ind pour l'indice (facultatif, par défaut CL1)
- age pour l'agence (obligatoire)
- cli : code client (obligatoire)
- aff : chantier (facultatif)
- cod : pour le code de l'adresse (facultatif)

Si le paramètre cod n'est pas précisé, la procédure récupère toutes les adresses du client.

Si le paramètre cod est précisé, seule l'adresse correspondant au code est récupérée.

Le résultat est une structure JSON comportant un tableau ADRESSE de structures. Chacune de ces structures comporte des membres correspondant aux rubriques de la table ADRTIAP des enregistrements trouvés.

Structure de la réponse :



Fonction adresseClient : écriture en méthode POST

La fonction adresseClient en méthode POST permet d'enregistrer une nouvelle adresse pour un client ou de modifier une adresse existante.

Exemple d'appel :

http://host/V1/adresseClient?mode=A

Et présence de la structure de données dans le corps de la requête.

Paramètres :

- Le seul paramètre présent dans l'url est optionnel : "mode" qui peut prendre la valeur « A » (ajout) ou « M » (modification). Si le paramètre n'est pas précisé, il est par défaut à « A » (création d'une nouvelle adresse)

- En méthode POST, les données sont transmises sous forme d'une structure JSON présente dans le corps de la requête http.

  Exemple de structure JSON :

  {

  "ADASOC": "001",

  "ADAIND": "CL1",

  "ADAAGE": "001",

  "ADACOD": "",

"ADACLI": "100004",
"ADAAFF": "",
"ADASIR": "11111111111111",
"ADATIT": "SARL",
"ADARSO": "TEST",
"ADARU1": "1 rue de la mairie",
"ADACPO": "14000",
"ADAVIL": "Caen"
}

Fonctionnement :

- En demande de création d'adresse, le code ADPCOD est laissé vide.
- En demande de modification, ADPCOD doit être renseigné

Réponse

La réponse est constituée d'une structure JSON comportant "response" avec une valeur KO en cas d'échec, et OK en cas de succès et un membre "message".

Fonction de recherche de clients : cc45n

La fonction de recherche de client cc45n propose une recherche un peu à la façon de la fenêtre cc45n d'ANAEL TT RS.

Exemple d'appel :

http://host/V1/cc45n?soc=001&age=350&ind=CL1&rso=&blo=&sir=&aff=&cli=100056

Les paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire si le code client "cli" est fourni
- Le code "ind", obligatoire, avec comme valeur "CL1" pour client, "CL5" pour chantier, "PR1" pour prospect, "PR5" pour chantier de prospect.
- L'indication de blocage "blo" facultatif, qui peut prendre les valeurs "true" ou "false" : "true" pour les clients bloqués, "false" pour les clients non bloqués, paramètre absent pour tous les clients.
- Le siren ou siret "sir", facultatif
- La raison sociale "rso", ou du moins le commencement de la raison sociale

Il existe une préséance de la méthode de recherche en fonction des paramètres fournis :

1. Code client ou code client + code chantier, respectivement "cli" et "aff". Dans ce cas, le paramètre "age" est requis.
2. Raison sociale avec le paramètre "rso"
3. Siren / Siret avec le paramètre "sir"

La réponse est une structure JSON comportant les membres "response" et "message", et un tableau CLI de structures.

La structure de chaque élément du tableau CLI reprend comme membres les rubriques de INSTIAP pour les clients retrouvés.

Fonction chantier

La fonction chantier permet de récupérer la liste des chantiers d'un client.

Exemple d'appel : http://host/V1/chantier?soc=001&age=130&ind=&cli=100056&blo=false

Paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- Le code client "cli" obligatoire
- L'indice "ind" facultatif, qui peut prendre les valeurs "CL5" (valeur par défaut) ou "PR5 »
- L'indication de blocage "blo" facultatif, qui peut prendre les valeurs "true" ou "false" : "true" pour les chantiers bloqués, "false" pour les chantiers non bloqués, paramètre absent pour tous les chantiers

Valeur de Retour :

Le retour est constitué d'une structure JSON comportant outre les membres "response" et "message" un tableau CHA de structures JSON.

Chaque structure reprend quelques membres de INSTIAP, en prenant comme nom de membre les 3 dernières lettres de la rubrique de INSTIAP correspondante. Exemple : le membre RSO de la structure JSON correspond à la rubrique INSTIAP.TIARSO du chantier.

Fonction client : lecture en méthode GET

La fonction client en mode GET permet de disposer des principales rubriques de la fiche client de INSTAP et INSTIAP2

Exemple d'appel : http://host/V1/client?soc=001&age=130&ind=CL1&cli=100056&aff=

Les paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- L'indice "ind" facultatif, dont la valeur par défaut est "CL1"
- Le code client "cli" obligatoire
- Le code chantier "aff" facultatif, uniquement en cas de recherche de la fiche chantier.

  Si le paramètre "aff" prend la valeur "*", alors sont renvoyés à la fois la fiche client et tous les chantiers.

La réponse est une structure JSON comportant les membres "response" et "message", et un tableau CLIENT de structures JSON. Chaque structure reprend comme membres les principales rubriques de INSTAP et INSTIAP2 dont ils reprennent les noms.

Fonction client : écriture en méthode POST

La fonction client en méthode POST permet d'enregistrer un nouveau client ou un nouveau chantier ou de modifier un client ou un chantier existant.

Exemple d'appel :

http://host/V1/client?aff=true

Et présence de la structure de données dans le corps de la requête.

Paramètres :

- En méthode POST, les paramètres les données sont transmises sous forme d'une structure JSON présente dans le corps de la requête http. La structure attendue est identique à celle fournie par la fonction client en méthode GET.

- Le seul paramètre présent dans l'url est optionnel, "aff" qui peut prendre la valeur "true" permet d'indique que l'on souhaite créer un chantier à l'identique de la fiche client.

Fonctionnement :

- En demande de création client, le code client "TIACLI" est laissé vide.
- La présence d'un code dans TIACLI indique que l'on souhaite procéder à la mise à jour du client

  La création n'est exécutée qu'en cas de respect de toutes les règles en vigueur pour la construction de la fiche.

Réponse

La réponse est constituée d'une structure JSON comportant "response" avec une valeur KO en cas d'échec, et OK en cas de succès, un membre "message" pour recevoir les messages d'erreur rencontrés pendant l'exécution, et le membre CLIENT contenant le code client si la création a fonctionné, et une chaine vide sinon.

Fonction clients : liste des clients

La fonction clients est une fonction en méthode GET pour la présentation de données.

Exemple d'appel : http://host/V1/clients?soc=001&age=130&ind=CL1

Paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- L'indice "ind" facultatif, à la valeur "CL1" pour les clients, "PR1" pour les prospects. En cas d'absence, prend la valeur "CL1"
- Le code du chargé d'affaire "chg" optionnel
- Le drapeau de blocage "blo" : facultatif, qui peut être à "true" ou "false".
    - Si blo = true : affichage des clients bloqués
    - Si blo = false : affichage des clients non bloqués
    - Si blo non renseigné : affichage de tous les clients

Résultat

Le résultat renvoyé par la fonction est une structure JSON comportant les deux membres du compte-rendu d'exécution "response" et "message", et un tableau CLIENTS comprenant des structures JSON. Chaque structure reprend des membres de INSTIAP, les noms des membres reprennent les 3 dernières lettres des rubriques correspondantes de INSTIAP.

Détail des rubriques de la structure :

RSO	Raison sociale
CLI	Code client
SIR	SIRET
RU1	Adresse ligne 1
RU2	Adresse ligne 2
CPO	Code postal
VIL	Ville
TLP	Téléphone

TTI	INSEE commune
CV1 CV2 CV3	Civilité contact 1/2/3
CO1 CO2 CO3	Contact 1/2/3
COE	Coef
AGE	Agence
CHG	Code chargé d'affaire

Fonction contacts : lecture en méthode GET

La fonction contacts en mode GET permet de lister les contacts d'un client ou d'un chantier (données de la table INSTICP)

Exemple d'appel : http://localhost/V1/contacts?soc=001&age=001&ind=CL1&mat=100003&cpt=001

Les paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- L'indice "ind" obligatoire
- Le code client "mat" obligatoire
- Le compteur « cpt » facultatif (si on veut récupérer un contact en particulier au lieu de la liste complète)

La réponse est une structure JSON comportant les membres "response" et "message", et un tableau CONTACTS de structures JSON. Chaque structure reprend comme membres les rubriques de INSTICP (3 dernières lettres de la rubrique).

Exemple :

```
{
  "response": "OK",
  "message": "",
  "CONTACTS": [
    {
      "SOC": "001",
      "IND": "CL1",
      "AGE": "001",
      "MAT": "100003",
      "CPT": "003",
      "FON": "REPI",
      "TIT": "MR",
      "NOM": "TEST ANNIE",
      "TLP": "06 01 02 02 04",
      "TLC": "",
      "PST": "",
      "MAI": "annie.test4@toto.fr",
      "TC1": "",
      "TC2": "",
      "TC3": "",
      "TC4": "",
      "TC5": ""
    }
```


Fonction contacts : écriture en méthode POST

La fonction contacts en méthode POST permet de créer ou de modifier un contact sur un client existant (ou un chantier)

Exemple d'appel :

http://host/V1/contacts

Et présence de la structure de données dans le corps de la requête.

Paramètres :

- En méthode POST, les données sont transmises sous forme d'une structure JSON présente dans le corps de la requête http. La structure attendue est identique à celle fournie par la fonction contacts en méthode GET.
- Paramètre « VERB » obligatoire : C pour création, M pour modification

Fonctionnement :

- En création, ne pas mettre le compteur CPT dans la structure. Le compteur est généré automatiquement
- En modification, CPT obligatoire pour mettre à jour le bon contact. Seules les rubriques présentes dans le JSON sont modifiées

Exemple de JSON à transmettre pour une création de contact :

```
{
    "verb":"C",
    "SOC": "001",
    "IND": "CL1",
    "AGE": "001",
    "MAT": "100003",
    "FON": "REPI",
    "TIT": "MR",
    "NOM": "Martin Test",
    "TLP": "06 01 02 03 04",
    "TLC": "",
    "PST": "",
    "MAI": "martin.test@toto.fr",
    "TC1": "",
    "TC2": "",
    "TC3": "",
    "TC4": "",
    "TC5": ""
}
```

Exemple de JSON à transmettre pour une modification de contact :

Les rubriques en jaune sont obligatoires pour déterminer le contact à modifier. Les rubriques suivantes sont celles à modifier. Dans cet exemple, on ne modifie que le téléphone et le mail. Les autres données restent inchangées.

```
{
    "verb":"M",
    "SOC": "001",
    "IND": "CL1",
    "AGE": "001",
    "MAT": "100003",
    "CPT": "005",
    "TLP": "06 99 00 99 00",
    "MAI": "martin.test@titi.fr"
}
```

Réponse

La réponse est constituée d'une structure JSON comportant "response" avec une valeur KO en cas d'échec, et OK en cas de succès, un membre "message" pour recevoir les messages d'erreur rencontrés pendant l'exécution.

Fonction couvmode : le mode de calcul de l'encours

Le mode de calcul de l'encours est paramétré dans ANAET en étant soit par client, soit par siren. La fonction couvmode permet de connaître ce paramétrage.

Exemple d'appel : http://host/V1/couvmode?soc=001

Paramètre : le code société obligatoire

Réponse : une structure JSON comportant les membres "response" et "message". Si "response = "KO", le membre "niveausiren" prend la valeur "true" si l'encours est évalué au siren, et "false" si l'encours est évalué au niveau client.

Fonction couv GET : couverture assurance-crédit

En méthode GET, la fonction réalise Lecture de la couverture. Il y a deux méthodes de gestion de la couverture : niveau siren ou niveau client. La méthode en cours dans l'entreprise est paramétrée dans ANAELTT.

Exemple d'appel : http://host/V1couv?soc=001&age=130&cli=100001&siren=

Paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire si vous êtes en mode "client"
- Le code client "cli" obligatoire si vous êtes e mode "client"
- Le code siren "siren" obligatoire si vous n'êtes pas en mode "client", dont en mode "siren"

Résultat

Le résultat obtenu est une structure JSON comportant les membres "response", "message", "niveausiren" comme dans l'appel de couvmode, ainsi que le tableau "COUV" de structures.

Chacune d'entre elle reprend les enregistrements de INSTIEP trouvés, dont les noms des membres reprennent à l'identique les principales rubriques de INSTIEP, ainsi que 3 libellés lus dans le paramétrage : TIEROPLIB, TIECOPLIB et TIETOPLIB.

Ils sont respectivement les libellés associés aux codes des type de garantie présents dans INSTIEP.TIEROP, INSTIEP.TIECOP et INSTIEP.TIETOP.

Fonction couv POST : mettre à jour la couverture

La forme POST de la fonction couv prend comme paramètre une structure de donnée présente dans le corps de la requête. Il n'y a pas de paramètres dans l'url.

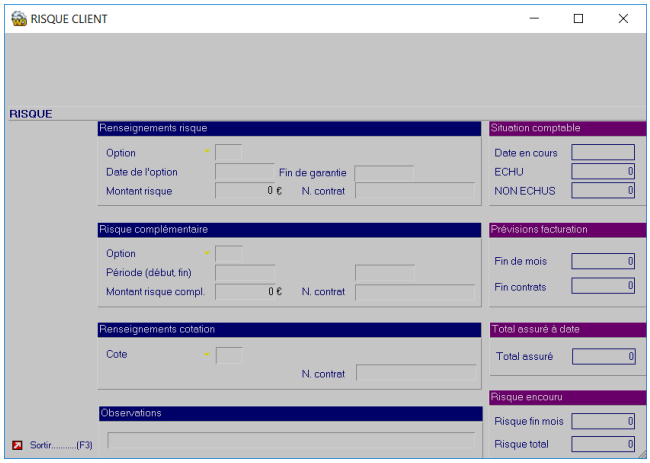
Exemple d'appel : http://host/V1/couv

Paramètre :

Structure JSON comportant les membres suivants :

Membre	Description	Contrainte	CC55N
TIESOC	Code société	[0-9A-Z]{3}	
TIEAGE	Code agence	[0-9A-Z]{3}	
TIECLI	Code client	[0-9]{6}	
TIEAFF	Code chantier	^$|[0-9]{6}	
Renseignement Risque			
TIEROP	Option de couverture	3 car.	Option
TIERDA	Date de l'option	Date	Date de l'option
TIEDFG	Date de fin de garantie	Date	Fin de garantie
TIERMT	Montant de couverture	Numérique	Montant du risque
TIERCO	N° de Contrat	14	N. contrat
Risque complémentaire			
TIECOP	Option	3 car.	Option
TIECDB	Date de début	Date	Période début
TIECFI	Fin	Date	
TIECMT	Montant	Numérique	Montant risque compl.
TIECCO	Complément contrat	14 car.	N. contrat
Renseignement cotation			
TIETOP	Option de cotation	3 car.	
TIETCO	Contrat	14 car.	
Observations			
TIEOBS	Observations	60 car.	

Affichage de la fenêtre de risque client cc55n :



Exemple :


```
{
  TIESOC: '001',
  TIEAGE: '130',
  TIECLI: '100001',
  TIEAFF: '',
  TIEROP: 'GA ',
  TIERDA: '2018-08-14',
  TIERMT: 45000,
  TIERCO: '',
  TIECOP: '',
  TIECDB: '',
  TIECFI: '',
  TIECMT: 0,
  TIECCO: '',
  TIETOP: '',
  TIEOBS: 'SFAC / GA',
  TIETCO: '',
  TIEDEC: '2019-06-17',
  TIEDFG: ''
}
```


Fonction ctrlst : liste des contrats

La fonction ctrlst permet de disposer de la liste des contrats.

Exemple d'appel :

http://host/V1/ctrlst?soc=001&age=350&ctr=&cli=&aff=&mat=&fmo=&d1=&d2=2019-06-17

Paramètres :

- Code société "soc" obligatoire
- Code agence "age" obligatoire
- Matricule salarié "mat" facultatif
- Code client "cli" facultatif
- Code chantier "aff" facultatif, utilisé uniquement si présence de "cli"
- Date de début d'intervalle d1, facultative. En cas d'absence, le 1er janvier de l'année en cours est utilisé
- Date de fin d'intervalle facultative : en cas d'absence on prend la date du jour
- Motif de fin de mission "fmo" facultatif
- Code chargé d'affaire "chg" facultatif
- Code contrat "crt" facultatif

Réponse :

La réponse est une structure JSON comportant les membres "response" et "message", ainsi qu'un tableau CTR de structures json. Chacune comprend comme membre des rubriques de ETTCTCP pour le contrat trouvé, ainsi que les rubriques NBJ exprimant le nombre de jours calendaires du contrat, STSRSO pour la raison sociale du client, CTCNOM pour le nom de l'intérimaire et CTCPRE pour son prénom.

La recherche retourne tous les contrats qui ont débuté ou qui se sont terminé dans l'intervalle.

Fonction groupe : liste des groupes client

La fonction groupe permet de disposer de la liste des groupes de clients.

Exemple d'appel :

http://host/V1/groupe?soc=001

Paramètres :

- Code société "soc" obligatoire
- Code groupe "cli" facultatif : permet d'afficher les données d'un seul groupe

Réponse :

La réponse est une structure JSON comportant les membres "response" et "message", ainsi qu'un tableau GRO de structures json. Chaque structure reprend comme membres les rubriques de INSTIUP

Chapitre 4 Fonctions diverses

Quelques fonctions qui peuvent se révéler pratiques bien qu'inclassables :

	Personnel	Client	Commande	Relevés d'heures	Autre	Contrats	
/V1/adresseETT					X		Adresses ETT
/V1/adresseOrganisme					X		Adresses organismes
/V1/adresseCVMSiren					X		Adresses Siren, centre VM, groupe
/V1/agelst					X		Liste des agences
/V1/combo					X		Liste de valeurs possibles
/V1/ferie					X		Jour férié
/V1/feries					X		Liste des jours fériés
/V1/insee					X		Recherche code insee
/V1/valpos					X		Valeurs possibles d'une option
/V1/verif_param					X		Teste l'existence d'une valeur
/V1/wsini					X		Teste la présence des fichiers INI et la connexion.
/V1/wsparam					X		Teste quelques paramètres
/V1/wstestpdf					X		Vérifie les droits pour la production de pdf

Fonction adresseETT

La fonction adresseETT en méthode GET, permet de récupérer les adresses d'une agence… (données du fichier ADRET1P)

L'URL est de la forme : http://server//V1/adresseETT?soc=001&ind=AGE&age=002

Les paramètres sont :

- soc pour le code société (obligatoire)
- ind pour l'indice (obligatoire)
- age pour l'agence (obligatoire)

- cod : pour le code de l'adresse (facultatif)

Si le paramètre cod n'est pas précisé, la procédure récupère toutes les adresses de l'entité.
Si le paramètre cod est précisé, seule l'adresse correspondant au code est récupérée.

Le résultat est une structure JSON comportant un tableau ADRESSE de structures. Chacune de ces structures comporte des membres correspondant aux rubriques de la table ADRET1P des enregistrements trouvés.

Structure de la réponse :

Fonction adresseOrganisme

La fonction adresseOrganisme en méthode GET, permet de récupérer les adresses d'un organisme (données du fichier ADRDIAP)

L'URL est de la forme : http://server//V1/adresseOrganisme?soc=001&ind=ORS&age=AGI

Les paramètres sont :

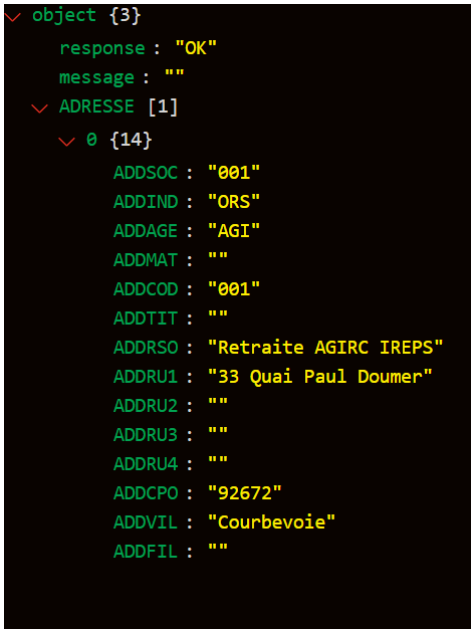
- soc pour le code société (obligatoire)
- ind pour l'indice (obligatoire)
- age pour l'agence (obligatoire)
- mat (facultatif)
- cod : pour le code de l'adresse (facultatif)

Si le paramètre cod n'est pas précisé, la procédure récupère toutes les adresses de l'organisme

Si le paramètre cod est précisé, seule l'adresse correspondant au code est récupérée.

Le résultat est une structure JSON comportant un tableau ADRESSE de structures. Chacune de ces structures comporte des membres correspondant aux rubriques de la table ADRDIAP des enregistrements trouvés.

Structure de la réponse :



Fonction adresseCVMSiren : lecture en méthode GET

La fonction adresseCVMSiren en méthode GET, permet de récupérer les adresses d'un SIREN, centre de visite médicale ou groupes…(données du fichier ADRTIUP)

L'URL est de la forme : http://server//V1/adresseCVMSiren?soc=001&ind=CAA&cli=RRR

Les paramètres sont : (correspondent aux champs de INSTIUP)

- soc :code société (obligatoire)
- ind : l'indice (obligatoire)
- age (facultatif)
- cli (obligatoire)
- aff (facultatif)

- cod : pour le code de l'adresse (facultatif)

Si le paramètre cod n'est pas précisé, la procédure récupère toutes les adresses du siren, cvm….
Si le paramètre cod est précisé, seule l'adresse correspondant au code est récupérée.

Le résultat est une structure JSON comportant un tableau ADRESSE de structures. Chacune de ces structures comporte des membres correspondant aux rubriques de la table ADRTIUP des enregistrements trouvés.

Structure de la réponse :


```
object {3}
  response : "OK"
  message : ""
▾ ADRESSE [2]
  ▾ 0 {17}
      ADUSOC : "001"
      ADUIND : "CVM"
      ADUAGE : ""
      ADUCLI : "CVM009"
      ADUAFF : ""
      ADUCOD : "001"
      ADUSIR : ""
      ADUTIT : "01"
      ADURSO : "2 rue du cheval"
      ADURU1 : ""
      ADURU2 : ""
      ADURU3 : ""
      ADURU4 : ""
      ADUCPO : "92150"
      ADUVIL : "SURESNES"
      ADURTG : ""
      ADUFIL : ""
```


Fonction adresseCVMSiren : écriture en méthode POST

La fonction adresseCVMSiren en méthode POST permet d'enregistrer une nouvelle adresse pour un siren, cvm ou groupe ou de modifier une adresse existante dans le fichier ADRTIUP

Exemple d'appel :

http://host/V1/adresseCVMSiren?mode=A

Et présence de la structure de données dans le corps de la requête.

Paramètres :

- Le seul paramètre présent dans l'url est optionnel : "mode" qui peut prendre la valeur « A » (ajout) ou « M » (modification). Si le paramètre n'est pas précisé, il est par défaut à « A » (création d'une nouvelle adresse)

- En méthode POST, les données sont transmises sous forme d'une structure JSON présente dans le corps de la requête http.

  Exemple de structure JSON :


```
{
"ADUSOC":"001",
"ADUIND":"CVM",
"ADUAGE":"",
"ADUCLI":"CVM009",
 "ADUCOD":"",
"ADUAFF":"",
"ADUSIR":"",
"ADUTIT":"01",
"ADURSO":"2 rue du cheval",
"ADURU1":"",
"ADURU2":"",
"ADURU3":"",
"ADURU4":"",
"ADUCPO":"92150",
"ADUVIL":"SURESNES",
"ADURTG":"",
"ADUFIL":""
}
```


Fonctionnement :

- En demande de création d'adresse, le code ADUCOD est laissé vide.
- En demande de modification, ADUCOD doit être renseigné

Réponse

La réponse est constituée d'une structure JSON comportant "response" avec une valeur KO en cas d'échec, et OK en cas de succès et un membre "message".

Fonction agelst : la liste des agences

Cette fonction permet de récupérer la liste des agences, et de faire une recherche par son nom ou son département.

Exemple d'appel : http://frprnjdcarre/V1/agelst?soc=001&lib=&dep=13&open=true

Paramètres :

- Code société "soc" obligatoire
- Fragment de la raison sociale de l'agence "lib" facultatif
- Département de l'agence "dep", facultatif
- Indicateur d'agence ouverte "open" facultatif, peut prendre les valeurs "true" pour ne sélectionner que des agences ouvertes, "false" pour des agences fermées, et sans valeur renvoie aussi bien les agences ouvertes que fermées.

Retour de la fonction

En retour, la fonction renvoie une structure JSON avec les membres "message" et "response", aindi qu'un tableau "AGE" de structures. Chacune représente les informations de ENTE1P concernant l'agence, dont chaque membre conserve en nom les 3 derniers caractères du nom de la rubrique d'ENTET1P correspondante.

Exemple de retour :


```
{
  AGE: [
    {
      SOC: '001',
      AGE: '130',
      TIT: 'SARL',
      RSO: 'GS MARSEILLE',
      RU1: '140  AVENUE DU 12 JUILLET 1998',
      RU2: 'Batiment C',
      CPO: '13290',
      VIL: 'AIX LES MILLES',
      TLP: '04 84 49 24 10',
      TLC: '',
      CO1: 'FENDHY MERHOUNI',
      CO2: '',
      SIR: '82861620100046',
      APE: '7820Z',
      FAG: 'EXP',
      URS: '937000002063372607',
      AGR: 'CVM371',
      UR3: '192192192',
      UR4: '01745BD 200',
      CC2: '',
      DFI: ''
    }
  ],
  message: '',
  response: 'OK'
}
```


Fonction fériés : liste des jours fériés d'une année

Donne la liste des jours fériés d'une année civile :

Exemple d'appel : http://host/V1/feries?an=2020&dep=

Paramètres :

- L'année civile "an" obligatoire, entre 2000 et 2099
- Le département, facultatif. EN cas d'absence, prend le département de la société de connexion.

Retour de la fonction

En retour la fonction donne un tableau "dates" des jours fériés. Chaque structure d'un élément de tableau comprend dans le membre "date", la date du jour férié, dans "nom" le nom du jour férié en français, et dans le membre "n" le numéro du jour dans la semaine, soit un chiffre entre 1 et 7, 1 étant le lundi.

Exemple de retour (fragment) :

Fonction férie : un jour férié dans ce département ?

La fonction "férié" indique en retour di un jour est férié dans un département.

Exemple d'appel : http://host/V1/ferie?date=2020-07-14&dep=21

Paramètres :

- Date au format AAAA-MM-JJ obligatoire
- Département facultatif. Si absent, prend le département de l'entreprise

Réponse

En réponse, la fonction renvoie une structure JSON comprenant membres "response", "message", date (la date fournie en argument), ferie à "true" ou "false", nom pour le nom en clair du jour férié.

Exemple de réponse :

Fonction combo : valeurs possibles des paramètres.

Dans ANAELTT, beaucoup de valeurs à renseigner sont des codes issus du paramétrage. Qui souhaite disposer des valeurs possibles d'un paramètre pour par exemple alimenter une interface, pourra par cette fonction disposer des valeurs possibles et de leur libellé. La fonction combo, en méthode GET permet de disposer de ces valeurs.

Exemple d'appel :

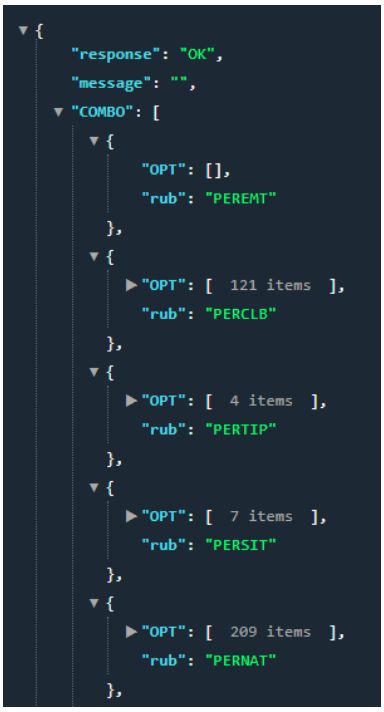
Paramètres :

- Code société "soc" obligatoire
- Code agence "age" obligatoire
- Code fenêtre "fen" qui peut prendre les valeurs "INSTIAP" ou "PERPERP". La première valeur renverra toutes les informations des rubriques de la fiche client, la seconde fera de même pour la fiche employée.

Retour de la fonction

En retour, la fonction renvoie une structure JSON comportant les membres "response" et message", ainsi que "COMBO" tableau de structures. Chacune d'entre elle comporte les membres "rub" pour le nom de la rubrique pour laquelle les options sont fournies, et "OPT" un tableau des valeurs possibles. Dans ce tableau, des structures JSON comportent chacune un membre "cod" pour la valeur du code à utiliser dans la rubrique, et un membre "lib" pour le libellé associé au code.

Exemple de retour :



```
{
  "response": "OK",
  "message": "",
  "COMBO": [
    {
      "OPT": [],
      "rub": "PEREMT"
    },
    {
      "OPT": [ 121 items ],
      "rub": "PERCLB"
    },
    {
      "OPT": [ 4 items ],
      "rub": "PERTIP"
    },
    {
      "OPT": [ 7 items ],
      "rub": "PERSIT"
    },
    {
      "OPT": [ 209 items ],
      "rub": "PERNAT"
    },
```


```
{
  "OPT": [
    {
      "cod": "...",
      "lib": ""
    },
    {
      "cod": "MME",
      "lib": "Madame"
    },
    {
      "cod": "MLE",
      "lib": "Mademoiselle"
    },
    {
      "cod": "MR",
      "lib": "Monsieur"
    }
  ],
  "rub": "PERTIP"
},
```

Fonction INSEE : rechercher un code insee

La qualité de l'information des codes insee est importante dans ANAEL pour la taxe transport, et la DSN.

Exemple d'appel : http://host/V1/insee?ville=apo&cp=21

Paramètres :

- "ville" pour un fragment du nom de la ville, facultatif
- "cp" pour le département ou le code postal, obligatoire si "ville" ne correspond pas au début du nom de la ville

Retour de la fonction

La fonction donne en retour une structure JSON comportant les membres "response" et "message", ainsi qu'un tableau VIL de structures JSON. Chacune comprend les membres "CP" pour le code postal, INSEE pour le code INSEE de la commune, et "ville" pour le nom de la commune.

Exemple de retour

Fonction valpos : les valeurs possibles

La fonction valpos permet de disposer des valeurs possibles pour une rubrique de la fiche client ou de la fiche personnel.

Exemple d'appel : http://host/V1/valpos?soc=001&age=350&ind=TIATIT&cli=

Les paramètres :

- Le code société "soc" obligatoire
- La rubrique "ind"
- Le code agence "age" facultatif. N'intervient que pour la recherche des valeurs par défaut, et els valeurs liées au cycle du client
- Le code client "cli" est facultatif : il n'intervient que pour la recherche suc les cycles, lors d'une requête avec ind=CYC

Retour

En retour, la fonction renvoie une structure JSON reprenant les membres "response" et "message", ainsi qu'un tableau PARAM de structures JSON. Chacune comprend 3 membres : "COD" pour le code qu'il est possible de proposer pour cette valeur, "LIB" pour le libellé associé à cette valeur, "IND" pour l'indice par lequel cette valeur est enregistrée dans PARDIVP, et enfin "DEF" qui prend la valeur "true" si cette valeur a été définie comme valeur par défaut dans le paramétrage d'ANAEL.

Exemple de retour :


```
{
  response: 'OK',
  message: '',
  PARAM: [
    { COD: 'ADD', LIB: 'ADM DECONCENTREE', IND: 'CVE', DEF: false },
    { COD: 'APE', LIB: 'ADM PUBLIQUE ETAT', IND: 'CVE', DEF: false },
    { COD: 'ADC', LIB: 'ADM. CENTRALE', IND: 'CVE', DEF: false },
```


Fonction verif_param : tester l'existence d'une valeur

Lorsque vous désirez vérifier la validité d'une valeur pour une rubrique, pour pouvez demander à la fonction verif_param de cette façon :

Exemple d'appel : http://host/V1/verif_param?soc=001&rub=CTCMOT&val=13

Paramètres

- Code société "soc", obligatoire
- Nom de la rubrique "rub", obligatoire
- Valeur à tester "val"

Retour

Le retour est une structure JSON comportant les membres "response" et "message", ainsi que le libellé s'il a été trouvé, la rubrique et sa valeur. Si aucune correspondance n'a été trouvée, "response" a la valeur "KO"

Exemple de retour :

```
{
  response: 'OK',
  message: '',
  lib: "Rempl. d'un salarié en cas de départ définitif précédant la suppression de son poste de travail",
  val: '13',
  COD: 'CTCMOT'
}
```


Fonction wsini : le webservice est bien connecté ?

La première fonction qui permette de tester la connexion du webservice à la base de données de bout en bout est celle-ci : wsini. C'est une méthode GET sans paramètre.

Exemple d'appel : http://host/V1/wsini

Retour de la fonction

Le retour comporte les différentes parties suivantes :

- Version : permet de connaitre la date de la version utilisée
- La partie "ini" qui renseigne si le fichier de configuration est bien défini, et s'il a bien été trouvé dans l'emplacement prévu. Si c'est le cas, "found" à la valeur "true" Si ce n'est pas le cas, travaillez déjà sur ce point.
- La partie "anaelini" indique le fichier anaeltt.ini à utiliser pour reprendre les identifiants de connexion. Cet emplacement dans le fichier aws.ini est désigné comme suit. Si la valeur

  ```
  [ANAELTT]
  INI=C:\Anael\Prog\TT\anaelTT.ini
  ```

  "found" de ce paragraphe est à "true", c'est que non seulement le fichier anaeltt.ini ou anaelgw.ini a bien été défini dans aws.ini, mais qu'il a été trouvé.
- La partie "target" reprend les informations de la base sur laquelle se connecter, telle qu'indiquée dans le fichier ini d'ANAEL.
- La partie "connection", qui si elle est à la valeur "true" indique que la connexion est réalisée sur la base cible.
- La partie « transactions » qui permet d'afficher
  - Le nombre de transactions (POST) autorisées sur un trimestre (selon la licence)
  - Le nombre de transactions (POST) réalisées sur le dernier trimestre

Exemple de retour de la fonction wsini :

```
{
  "Version": "Version du 07/12/2023",
  "ini": {
    "defined": true,
    "file": "C:\\dataWS\\ANAELWS\\aws.ini",
    "found": true
  },
  "anaelini": {
    "defined": true,
    "file": "C:\\ANAELTTV9R4b\\Prog\\TT\\anaelTT.ini",
    "found": true
  },
  "target": {
    "numsoc": "1",
    "name": "
    "server": "
    "type": "WinDevClientServeurHF",
    "database": "
    "user": "admin"
  },
  "connection": true,
  "nbTransactionsAutorisees": 45000,
  "nbTransactions": 12
```


Fonction wstestpdf : Impression pdf possible ?

L'impression PDF nécessite de disposer de droits d'écriture dans un emplacement du serveur. Si les droits attribués à l'utilisateur que représente le serveur web ne permettent pas d'utiliser un emplacement pour l'écriture de fichier, alors la production de fichier pdf est compromise. Plusieurs emplacements peuvent être testé.

Exemple d'appel : http://host/V1/wstestpdf?rep=1

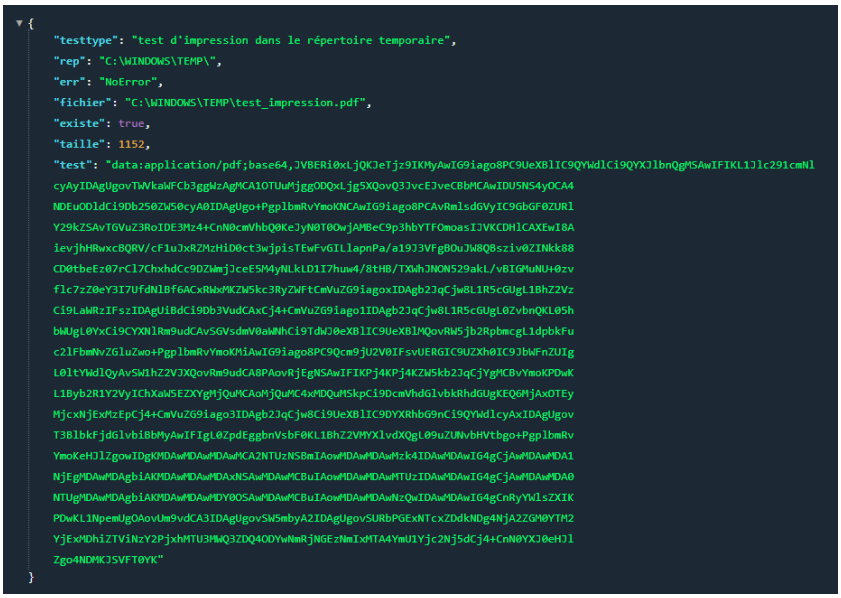
Paramètre :

- Le numéro de répertoire à tester "rep" :
  - 1 : Le répertoire d'exécution
  - 2 : Le répertoire des données
  - 3 : le répertoire de données commun
  - 4 : le répertoire temporaire

- 5 et le répertoire "DOCUMENTS" n'est pas défini dans ANAELTT.INI : On teste le répertoire temporaire
- 5 et le répertoire "DOCUMENTS" est défini dans ANAELTT.INI : On teste le répertoire des documents ANAEL

Exemple de réponse :

La réponse permet de savoir si les droits étaient suffisants pour réaliser l'opération dans le répertoire désigné par le paramètre "rep" :



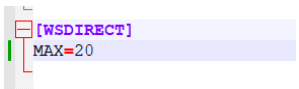
```
{
    "testtype": "test d'impression dans le répertoire temporaire",
    "rep": "C:\WINDOWS\TEMP\",
    "err": "NoError",
    "fichier": "C:\WINDOWS\TEMP\test_impression.pdf",
    "existe": true,
    "taille": 1152,
    "test": "data:application/pdf;base64,JVBERi0xLjQKJeTjz9IKMyAwIG9iago8PC9UeXBlIC9QYWdlCi9QYXJlbnQgMSAwIFIKL1Jlc291cmNl
    cyAyIDAgUgovTWVkaWFCb3ggWzAgMCA1OTUuMjggODQxLjg5XQovQ3JvcEJveCBbMCAwIDU5NS4yOCA4
    NDEuODldCi9Db250ZW50cyA0IDAgUgo+PgplbmRvYmoKNCAwIG9iago8PCAvRmlsdGVyIC9GbGF0ZURl
    Y29kZSAvTGVuZ3RoIDE3Mz4+CnN0cmVhbQ0KeJyN0T0OwjAMBeC9p3hbYTFOmoasIJVKCDHlCAXEwI8A
    ievjhHRwxcBQRV/cF1uJxRZMzHiD0ct3wjpisTEwFvGILlapnPa/a19J3VFgBOuJW8QBsziv0ZINkk88
    CD0tbeEz07rCl7ChxhdCc9DZWmjJceE5M4yNLkLD1I7huw4/8tHB/TXWhJNON529akL/vBIGMuNU+0zv
    flc7zZ0eY3I7UfdNlBf6ACxRWxMKZW5kc3RyZWFtCmVuZG9iagoxIDAgb2JqCjw8L1R5cGUgL1BhZ2Vz
    Ci9LaWRzIFszIDAgUiBdCi9Db3VudCAxCj4+CmVuZG9iago1IDAgb2JqCjw8L1R5cGUgL0ZvbnQKL05h
    bWUgL0YxCi9CYXNlRm9udCAvSGVsdmV0aWNhCi9TdWJ0eXBlIC9UeXBlMQovRW5jb2RpbmcgL1dpbkFu
    c2lFbmNvZGluZwo+PgplbmRvYmoKMiAwIG9iago8PC9Cm9jU2V0IFsvUERGIC9UZXh0IC9JbWFnZUIg
    L0ltYWdlQyAvSW1hZ2VJXQovRm9udCA8PAovRjEgNSAwIFIKPj4KZW5kb2JqCjYgMCBvYmoKPDwK
    L1Byb2R1Y2VyIChXaW5EZXYgMjQuMCAoMjQuMC4xMDQuMSkpCi9DcmVhdGlvbkRhdGUgKEQ6MjAxOTEy
    MjcxNjExMzEpCj4+CmVuZG9iago3IDAgb2JqCjw8Ci9UeXBlIC9DYXRhbG9nCi9QYWdlcyAxIDAgUgov
    T3BlbkFjdGlvbiBbMyAwIFIgL0ZpdEggbnVsbF0KL1BhZ2VMYXlvdXQgL09uZUNvbHVtbgo+PgplbmRv
    YmoKeHJlZgowIDgKMDAwMDAwMDAwMCA2NTUzNSBmIAowMDAwMDAwMzk4IDAwMDAwMDA1
    NjEgMDAwMDAgbiAKMDAwMDAxNSAwMDAwMCBuIAowMDAwMDAwMTUzIDAwMDAwIG4gCjAwMDAwMDA0
    NTUgMDAwMDAgbiAKMDAwMDAwMDY0OSAwMDAwMCBuIAowMDAwMDAwNzQwIDAwMDAwIG4gCnRyYWlsZXIK
    PDwKL1NpemUgOAovUm9vdCA3IDAgUgovSW5mbyA2IDAgUgovSURbPGExNTcxZDdkNDg4NjA2ZGM0YTM2
    YjExMDhiZTViNzY2PjxhMTU3MWQ3ZDQ4ODYwNmRjNGEzNmIxMTA4YmU1Yjc2Nj5dCj4+CnN0YXJ0eHJl
    Zgo4NDMKJSVFT0YK"
}
```


Fonction wsdirect : l'accès aux données de la base

La fonction wsdirect permet l'accès aux enregistrements des tables de la base, pourvu que vous puissiez fournir les paramètres table, clef et val. Toutes les tables de l'analyse de base sont concernées, sauf les tables de sécurité PARPS1P, PARPS1P2, PARPS2P.

Le résultat est un jeu de résultats pouvant atteindre un nombre d'enregistrements définis en paramètre depuis le fichier de configuration, et dont la valeur par défaut est 10.

Le paramétrage pour modifier ce nombre maximum d'enregistrements dans le résultat se fait dans le fichier aws.ini : (exemple pour afficher 20 enregistrements maximum)



Exemples d'appel :
http://host/V1/wsdirect?table=PARDIVP&clef=PARDIVL1&val=001TYK

Dans cet exemple, on récupère les valeurs des différents types de contrats tels que présents dans le paramétrage.

http://host/V1/wsdirect?table=PPXPPHP&clef=PPYPPHL3&val=001130100009

Dans cet exemple, on récupère quelques lignes de bulletin de salaire du matricule 130100009

http://host/V1/wsdirect?table=PERPERP&clef=PERMAT&val=130100001

Dans cet exemple, on récupère la fiche du personnel de matricule 130100001

Paramètres :

- Le nom de la table "table", issue de l'analyse d'ANAEL TT RS
- Le nom de la clef "clef", qui peut être le nom d'une clef composée ou plus rarement une seule colonne clef comme dans le cas PERPERP.PERMAT. Les informations concernant les tables d'ANAEL TT RS sont disponibles auprès d'INFOR.
- La valeur de la clef. On bénéficie du fait que beaucoup de clef utilisée dans la base peuvent se construire par concaténation de leurs constituants.

Retour de la fonction

En retour, la fonction renvoie une structure JSON comportant les membres "response", "message", "table" pour reprendre le nom de la table, et "value" un tableau de structures JSON représentant les enregistrements trouvés

Par définition, les structures trouvées dans le tableau valeurs de "value" ne sont jamais de même structure, puisque celle-ci dépend de la table passée en paramètre

Exemple de retour :


```
{
    "response": "OK",
    "message": "",
    "table": "PARDIVP",
    "value": [
        {
            "PARSOC": "001",
            "PARIND": "TYK",
            "PARCOD": "BCD",
            "PARLIB": "CDD (pour module borloo)",
            "PARVAL": "D",
            "PARLB2": "000111100022220000011000015  00000010000012",
            "PAREXL": "999",
            "PARDCR": "",
            "PARDMO": "",
            "PARDSU": ""
        },
        {
            "PARSOC": "001",
            "PARIND": "TYK",
            "PARCOD": "BCI",
            "PARLIB": "CDI (pour module borloo)",
            "PARVAL": "I",
            "PARLB2": "",
            "PAREXL": "999",
            "PARDCR": "",
            "PARDMO": "",
            "PARDSU": ""
        },
```

Chapitre 5 Multi-bases

Si la configuration multi-bases est mise en place (voir documentation d'installation du webservice), un paramètre supplémentaire doit être ajouté à chaque appel de fonction.

Ce paramètre supplémentaire est le suivant : **ParamEnvir=XXXX** où XXXX est le suffixe du fichier awsXXXX.ini configuré précédemment.

Exemple :

La fonction candidat, qui permet de récupérer les données d'un personnel, a pour url d'appel :

http://server/V1/candidat?mat=1300001

Cet appel va se connecter sur la base définie dans le fichier anaelTT.ini, lui-même défini dans le fichier aws.ini.

Dans le cas d'une configuration multi-bases, l'appel se fera de cette façon :

- http://server/V1/candidat?mat=1300001&ParamEnvir=TEST
  Dans ce cas, on pointe sur le fichier awsTEST.ini dans lequel on a défini un fichier anaelTT.ini permettant la connexion à une première base

- http://server/V1/candidat?mat=1300001&ParamEnvir=PROD
  Dans ce cas, on pointe sur le fichier awsPROD.ini dans lequel on a défini un fichier anaelTT.ini différent permettant la connexion à une deuxième base.

Remarque : il est tout à fait possible, dans une configuration multi-base, de garder un premier fichier aws.ini (comme pour une configuration simple) et d'ajouter un (ou plusieurs) autre awsXXXX.ini

Dans ce cas l'appel simple (sans paramètre supplémentaire) pointera sur le fichier aws.ini et les appels avec le paramètre **ParamEnvir=XXXX** pointeront sur les fichiers awsXXXX.ini