



Anael Travail Temporaire Module Webservice - Fonctions

Version V9R5

Copyright © 2025 Infor

Tous droits réservés. Les termes et marques de conception mentionnés ci-après sont des marques et/ou des marques déposées d'Infor et/ou de ses partenaires et filiales. Tous droits réservés. Toutes les autres marques répertoriées ci-après sont la propriété de leurs propriétaires respectifs.

Avertissement important

Les informations contenues dans cette publication (y compris toute information supplémentaire) sont confidentielles et sont la propriété d'Infor.

En accédant à ces informations, vous reconnaissez et acceptez que ce document (y compris toute modification, traduction ou adaptation de celui-ci) ainsi que les copyrights, les secrets commerciaux et tout autre droit, titre et intérêt afférent, sont la propriété exclusive d'Infor. Vous acceptez également de ne pas vous octroyer les droits, les titres et les intérêts de ce document (y compris toute modification, traduction ou adaptation de celui-ci) en vertu de la présente, autres que le droit non-exclusif d'utilisation de ce document uniquement en relation avec et au titre de votre licence et de l'utilisation du logiciel mis à la disposition de votre société par Infor conformément à un contrat indépendant (« Objectif »).

De plus, en accédant aux informations jointes, vous reconnaissez et acceptez que vous devez respecter le caractère confidentiel de ce document et que l'utilisation que vous en faites se limite aux Objectifs décrits ci-dessus.

Infor s'est assuré que les informations contenues dans cette publication sont exactes et complètes.

Toutefois, Infor ne garantit pas que les informations contenues dans cette publication ne comportent aucune erreur typographique ou toute autre erreur, ou satisfont à vos besoins spécifiques. En conséquence, Infor ne peut être tenu directement ou indirectement responsable des pertes ou dommages susceptibles de naître d'une erreur ou d'une omission dans cette publication (y compris toute information supplémentaire), que ces erreurs ou omissions résultent d'une négligence, d'un accident ou de toute autre cause.

Reconnaissance des marques

Tous les autres noms de société, produit, commerce ou service référencé peuvent être des marques déposées ou des marques de leurs propriétaires respectifs.

Informations de publication

Version : Anael Travail Temporaire V9R5

Date de publication : 20 octobre 2025

Code du document : 1.0

Table des matières

À propos de ce manuel	7
Public concerné	8
Documents liés	8
Organisation.....	8
Historique du document.....	8
Contacter Infor	9
Chapitre 1 Fonctions par domaine.....	10
Mise en garde	10
Présentation des fonctions par domaine	10
Exploitation des fonctions	13
"response" et "message".....	13
Format de date	14
Emploi des minuscules et des majuscules dans les noms de paramètre et de rubrique	14
Visualisation des réponses.....	14
Talend API Tester.....	14
Chapitre 2 Fonctions personnel.....	16
Fonction adressePersonnel : lecture en méthode GET	17
Fonction adressePersonnel : écriture en méthode POST	18
Fonction canddbl.....	19
Fonction candidat	20
Particularités fonction candidat GET :	20
Particularités fonction candidat POST :	20
Les rubriques obligatoires	21
Fonction candplan	22
Fonction cc16n.....	23

Fonction candtel.....	24
Fonctions Profils et Caractéristiques	25
Fonction carfam	25
Fonction carserie	26
Fonction carval méthode GET.....	27
Fonction carval méthode POST	28
Fonction doclst.....	30
Fonction doc en méthode GET	31
Fonction doc en méthode POST	31
Fonction pcs.....	33
Fonction qualification	33
Fonction contratpdf : Edition du contrat en PDF	34
Fonction contrats : liste de contrats.....	35
Fonction CTRDET : détails d'un contrat.....	36
Fonction CTRFIN : financier d'un contrat.....	37
Fonction CTRHIS : historique d'un contrat.....	38
Fonction CTRLST : une autre liste de contrat.....	39
Fonction perlib : Le personnel par qualification.....	39
Fonction perqua GET : Qualifications d'un personnel	40
Fonction perqua POST : Inscrire une qualification	41
Fonction perqua DEL : Suppression de qualification	41
Fonction quaper : recherche candidat / qualification.....	42
Chapitre 3 Fonctions Client	43
Fonction actionco : les actions commerciales.....	43
Fonction adresseClient : lecture en méthode GET	44
Fonction adresseClient : écriture en méthode POST.....	45
Fonction de recherche de clients : cc45n.....	46
Fonction chantier	47
Fonction client : lecture en méthode GET	47
Fonction client : écriture en méthode POST	48
Fonction clients : liste des clients	49
Fonction contacts : lecture en méthode GET	50
Fonction contacts : écriture en méthode POST	51

Fonction couvmode : le mode de calcul de l'encours	53
Fonction couv GET : couverture assurance-crédit.....	53
Fonction couv POST : mettre à jour la couverture	53
Fonction ctrlst : liste des contrats	55
Fonction groupe : liste des groupes client.....	56
Chapitre 4 Fonctions de temps	57
Rappels.....	57
Fonction cycle : les cycle de semaines du client.....	58
Fonction semaine : les semaines d'un cycle client.....	58
Fonction horaires : les horaires journaliers	60
Fonction sem : la semaine de calendrier.....	62
Fonction rhsem : Squelette de la semaine de contrat.....	62
Fonction rhimport : les RH présents en RHIMPORT	64
Fonction rhimport en POST : pour importer	65
Fonction rh : lecture des relevés	67
Fonction rhfinancier : lecture du financier des relevés d'heures.....	71
Fonction planning : le planning du client	72
Chapitre 5 Les commandes	76
Fonction cde_lst pour la liste des commandes	76
Fonction cde_cde : lecture d'une commande.....	77
Fonction cde_fin GET : lecture du financier	78
Fonction cde_fin POST : Ajout dans le financier.....	79
Fonction cde_mdI : recherche de modèle de poste	80
Fonction cde_set : l'enregistrement de commande.....	82
Fonction cde_affectation : Affectation d'un personnel	86
Fonction cde_affectationraz	87
Chapitre 6 Fonctions diverses	88
Fonction adresseETT	88
Fonction adresseOrganisme	89
Fonction adresseCVMSiren : lecture en méthode GET	90
Fonction adresseCVMSiren : écriture en méthode POST	91

Fonction agelst : la liste des agences	92
Fonction fériés : liste des jours fériés d'une année	93
Fonction férie : un jour férié dans ce département ?	94
Fonction combo : valeurs possibles des paramètres	95
Fonction INSEE : rechercher un code insee	97
Fonction valpos : les valeurs possibles	97
Fonction verif_param : tester l'existence d'une valeur	98
Fonction wsini : le webservice est bien connecté ?	99
Fonction wstestpdf : Impression pdf possible ?	100
Fonction wsdirect : l'accès aux données de la base	101
Chapitre 7 Multi-bases.....	104

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 connaître 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 ne 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-Sécurisation (Infor_AnaelTTRS_V9R5_Webservice_Sécurité.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 entraîner 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
cde_affectation			X				Affectation d'une commande
cde_affectationraz			X				Suppression d'une affectation
cde_cde			X				Lecture commande
cde_fin			X				Financier d'une commande
cde_lst			X				Liste des lignes de commande
cde_mdl			X				Modèle de poste
cde_set			X				Enregistrement de commande
cde_test			X				Pour tests
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

Fonction	Personnel	Client	Commande	Relevés d'heures	Autre	Contrats	
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
cycle		x					Cycle de semaine client
doc	x	x					Document
doclst	x	x					Liste de documents
ferie					x		Jour férié
feries					x		Liste des jours fériés
groupe		x					Liste des groupes client
horaires		x					Horaires d'une semaine
insee					x		Recherche code insee
pcs	x						Liste des professions et catégories
perlib	x						Liste de personnels
Perqua	x						Qualifications d'un personnel
planning		x					Planning client (par journée)
prospect		x					Fiche client / prospect
qualification	x						Recherche de qualification
quaper	x						Recherche de personnel selon qualification
rh				x			Relevés d'heures
rh_semaine				x			Trouve la semaine du client
rhimport				x			Importation des relevés d'heures

Fonction	Personnel	Client	Commande	Relevés d'heures	Autre	Contrats	
rhsem				X			Schéma RH d'une semaine de contrat
sem		X					Jours d'une semaine
semaine		X					Semaines de cycle client
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¶metre2=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 chaîne 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/iemadiahbbedklepanmkjenfdebfpfe?hl=fr>

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

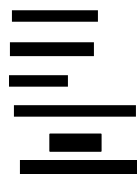
En remplacement de :

```

{ "response": "OK", "DBL": [ { "PERSOC": "001", "PERIND": "PRA", "PERMAT": "nnnnnnnn", "PERNOM": "XXXXXXXX",
"PERPRE": "XXXXXXXX", "PERNSS": "nnnnnnnnnnnn", ...

```

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 :

```

✓ object {6}
  response : "OK"
  message : ""
  PERNOM : " "
  PERPRE : " "
  PERMAT : "001100001"
  ✓ ADRESSE [3]
    ✓ 0 {8}
      ADPCOD : "001"
      ADPRU1 : "3 rue de "
      ADPRU2 : ""
      ADPRU3 : ""
      ADPRU4 : ""
      ADPCPO : "92500"
      ADPVIL : "Rueil"
      ADPFIL : ""
    ✓ 1 {8}
      ADPCOD : "002"
      ADPRU1 : "25 rue "
      ADPRU2 : ""
      ADPRU3 : ""
      ADPRU4 : ""
      ADPCPO : "75010"
      ADPVIL : "Paris"
      ADPFIL : ""

```

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 :

```
{
  "response": "OK",
  "DBL": [
    {
      "PERSOC": "001",
      "PERIND": "PRA",
      "PERMAT": " ",
      "PERNOM": " ",
      "PERPRE": " ",
      "PERNSS": " ",
      "PERDTN": "1979",
      "PERRU1": "4",
      "PERCPO": "85600",
      "PERVIL": "ST GEORGES DE MONTAIGU"
    }
  ]
}
```

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 paramètres mat (obligatoire) et ind, avec en valeur de mat le matricule du personnel recherché, comme dans cet exemple :

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

Le 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 PERPER, PERPERP2, une apparition, et 7 omissions :

Membre ajouté aux membres de PERPER 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: '130100061',
      CTCCHG: 'VRR',
      CTCCLI: '100014',
      CTCAFF: '100015',
      CTCQUA: 'CPLR01',
      CTCBAL: 10.03,
      CTCSTN: 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.PR PDT1
- DT1 pour celui de PERPRPP.PR PDT2

```
{
  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 :

<pre>{ verb: 'SX', SOC: '001', IND: 'PER', AGE: '130', MAT: '130100193', FAM: 'DS', SERIE: 'CVI', COM1: 'test 854', COM2: '', DT1: '', DT2: '', XCOM1: 'test 1', XCOM2: '' }</pre>	<p>Verb : indique l'action à réaliser</p> <p>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.</p> <p>Les valeurs possibles de IND sont "PER", "CLI", "CTR", "CT1", "CTD" respectivement pour personnel, client, contrat, modèle, commande</p>
--	---

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 chaîne 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 chaîne codée Base64 dans une valeur "file"

```
{ "file": 'JVBERi0xLjQKJeTjz9IKMyAwIG9iago8PC9UeXB1IC9QYWdl...' }
```

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 :

```
{
  "testtype": "test d'impression dans le répertoire temporaire",
  "rep": "C:\\WINDOWS\\TEMP\\",
  "attention": "Répertoire des documents ANAEL absent de C:\\Anaël\\Prog\\TT\\anaelTT.ini pour la société 9",
  "err": "NoError",
  "fichier": "C:\\WINDOWS\\TEMP\\test_impression.pdf",
  "existe": true,
  "taille": 1152,
  "test": "data:application/pdf;base64,JVBERi0xLjQKJeTjz9IKMyAwIG9iago8PC9UeXB1IC9QYWdlCi9QYXJlbnQgMSAwIFI1cyAyIDAgUGovTWkafC3ggWzAgMCA10TUuMjggODQxLjg5XQovQ3JvcEJveCBBMCAwIDU5NS4yOCA4NDEuODlkcj9Db250ZW50cyA0IDAgUGo+PgplbmRvYmoKNCAwIG9iago8PCAvRmlsdGVyIC9GbGF0ZURlY29kZSAvTG9uZ3RoIDE3Mz4+CnN0cmVhbQ0KeJyN0T00wjAMBc9p3hbYTF0moasIJVKCDHlCAXEwI8A"
```


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 chaîne à retrouver dans le libellé. Par défaut la recherche portera sur les libellés commençant par la chaîne fournie. Cette chaîne est codée URL. Si la chaîne commence par le caractère *, alors la recherche portera sur tous les libellés comprenant cette chaîne.
 - 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 on vérifie à son tour qu'il soit bien présent.

Une impression ou 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 chaîne base64 s'obtient en plaçant la chaîne 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=&xfi1=&xfi2=&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 ensemble, 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 "deb2" 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",
      "XEDLIQ": "",
      "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=&debl=&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 comportant 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¬=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/ctrlist		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 :

```

object {0}
  response : "OK"
  message : ""
  TIASOC : "001"
  TIAIND : "CL1"
  TIAAGE : "001"
  TIAAFF : ""
  TIARSO : 
  ✓ ADRESSE [2]
    ✓ 0 {0}
      ADACOD : "001"
      ADARU1 : "1 rue de la mairie"
      ADARU2 : ""
      ADARU3 : ""
      ADARU4 : ""
      ADACPO : "14000"
      ADAVIL : "Caen"
      ADAFIL : ""
    ✓ 1 {0}
      ADACOD : "002"
      ADARU1 : "14 rue de la mairie"
      ADARU2 : ""
      ADARU3 : ""
      ADARU4 : ""
      ADACPO : "92500"
      ADAVIL : "Rueil"
      ADAFIL : ""

```

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'indiquer 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 chaîne 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/V1/couv?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 1^{er} 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 terminés 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 de temps

	Personnel	Client	Commande	Relevés d'heures	Autre	Contrats	
/V1/cycle		X					Cycle de semaine client
/V1/semaine		X					Semaines de cycle client
/V1/horaires		X					Horaires d'une semaine
/V1/planning		X					Planning client (par journée)
/V1/sem		X					Jours d'une semaine
/V1/rhsem							Donne les horaires théoriques d'une semaine de contrat
/V1/rhimport							Lecture et écriture des relevés d'heures à importer
/V1/rh							Lecture des relevés d'heures

Rappels

Chaque client dispose d'au moins 1 cycle de semaine de travail, dont la numérotation commence par "001" et qui comporte au moins 1 semaine. La numérotation des semaines commence par "01".

Chaque semaine indique pour chacun des 7 jours de la semaine le code horaire à utiliser pour le jour. Le jour 1 correspond au lundi.

Le client dispose d'au moins 1 horaire journalier dont le premier est codifié "01". Chaque code horaire indique les heures de début et de fin, le nombre d'heures de travail de jour, de nuit, de jour férié.

Lors de la validation d'un contrat, seront enregistré le cycle utilisé, et la semaine de cycle à la date du début du contrat. Avec ces indications d'une part, et avec la codification des horaires du client d'autre part, on parvient à connaître pour une semaine calendaire, le code de semaine, et donc les horaires attendus.

Fonction cycle : les cycle de semaines du client

La fonction cycle permet de lister les cycles de semaine du client.

Exemple d'appel :

<http://host/V1/cycle?soc=001&age=130&cli=100003&aff=&cyc=>

Paramètres :

- Code société "soc" obligatoire
- Code agence "age" obligatoire
- Code client "cli" obligatoire
- Code chantier "aff" facultatif
- Code cycle "cyc" facultatif

Réponse

La réponse est une structure JSON comportant les membres "message" et "response" comportant CYC de structures JSON. Ces structures reprennent les principales rubriques des enregistrements de la table AFFAFP dont ils reprennent les noms.

Exemple de réponse :

```
{
  response: 'OK',
  message: '',
  CYC: [
    {
      AFFSOC: '001',
      AFFIND: 'CL1',
      AFFAGE: '130',
      AFFCLI: '100003',
      AFFAFF: '',
      AFFCL1: '',
      AFFCL2: '',
      AFFCL3: '',
      AFFCL4: '',
      AFFCL5: '',
      AFFCYC: '001',
      AFFLIB: 'Lun/ Ven 8H/17H Variable',
      AFFDUR: 35,
      AFFNBS: 1,
      AFFFER: 'I',
      AFFTCY: '',
      AFFTYP: '',
      AFFDC: ''
    }
  ],
}
```

Fonction semaine : les semaines d'un cycle client

La fonction semaine est en méthode GET et permet de disposer du détail d'une semaine de cycle d'un cycle client.

Exemple d'appel : <http://host/V1/semaine?soc=001&age=130&cli=100003&aff=&cyc=001>

Dans cet exemple d'appel, on demande la composition du premier cycle client, le cycle de code 001.

Paramètres :

- Code société "soc" obligatoire
- Code agence "age" obligatoire
- Code client "cli" obligatoire
- Code chantier "aff" facultatif
- Code du cycle "cyc"

Retour de la fonction

La fonction retourne une structure JSON comprenant les membres "message" et "response", ainsi qu'un tableau SEM de structures. Chacune reprend les principales rubriques de AFFAFGP dont les membres ont le même nom.

Exemple de retour :

```
{
  SEM: [
    {
      AFGSOC: '001',
      AFGIND: 'CL1',
      AFGAGE: '130',
      AFGCLI: '100003',
      AFGAFF: '',
      AFGCL1: '',
      AFGCL2: '',
      AFGCL3: '',
      AFGCL4: '',
      AFGCL5: '',
      AFGCYC: '001',
      AFGSEM: '01',
      AFGH01: '01',
      AFGH02: '01',
      AFGH03: '01',
      AFGH04: '01',
      AFGH05: '01',
      AFGH06: '',
      AFGH07: ''
    }
  ],
  response: 'OK',
  message: ''
}
```

Dans cet exemple, le code horaire du lundi de la première semaine du cycle 001 est "01", on constate qu'il n'y a pas de travail le samedi ni le dimanche.

Détail des rubriques :

AFGSOC	Code société
AFGIND	Indice
AFGAGE	Code agence
AFGCLI	Code client
AFGAFF	Code chantier
AFGCL1	Div 1
AFGCL2	Div 2

AFGCL3	Div 3
AFGCL4	Div 4
AFGCL5	Div 5
AFGCYC	Cycle auquel cette semaine appartient
AFGSEM	Code semaine
AFGHO1	Code horaire lundi
AFGHO2	Code horaire mardi
AFGHO3	Code horaire mercredi
AFGHO4	Code horaire jeudi
AFGHO5	Code horaire vendredi
AFGHO6	Code horaire samedi
AFGHO7	Code horaire dimanche

Fonction horaires : les horaires journaliers

La fonction horaire en méthode GET permet de disposer d'un horaire journalier par son code. A chaque jour d'une semaine de cycle est associé un horaire, désigné par son code.

Exemple d'appel : <http://host/V1/horaires?soc=001&age=130&cli=100003&aff=&hor=01>

Paramètres :

- Code société "soc" obligatoire
- Code agence "age" obligatoire
- Code client "cli" obligatoire
- Code chantier "aff" facultatif
- Code horaire "hor" obligatoire

Réponse

La réponse est une structure JSON comportant les membres "message" et "response", un tableau HOR de structures JSON. Chaque structure reprend les rubriques de l'enregistrement de la table AFFAFDP dont chaque membre reprend le nom de la rubrique correspondante.

Exemple de réponse :

```

{
  "HOR": [
    {
      "AFDSOC": "001",
      "AFDIND": "CL1",
      "AFDAGE": "130",
      "AFDCLI": "100003",
      "AFDAFF": "",
      "AFDCL1": "",
      "AFDCL2": "",
      "AFDCL3": "",
      "AFDCL4": "",
      "AFDCL5": "",
      "AFDHOR": "01",
      "AFDHDE": 8,
      "AFDHFI": 17,
      "AFDNBR": 7,
      "AFDNBH": 0,
      "AFDFDE": 0,
      "AFDFFI": 0,
      "AFDFBR": 0,
      "AFDFBH": 0,
      "AFDPSD": 0,
      "AFDPSF": 0,
      "AFDLIB": ""
    }
  ],
  "response": "Ok",
  "message": ""
}

```

AFDCL4	Div 4
AFDCL5	Div 5
AFDHOR	Code horaire
AFDHDE	Heure Début
AFDHFI	Heure Fin
AFDNBR	Nbr Heures
AFDNBH	Heures Nuit
AFDFDE	Heure début (féerie)
AFDFFI	Heure fin (ferié)
AFDFBR	Nbr heures (ferié)
AFDFBH	Heure nuit (ferié)
AFDPSD	Nombre d'heures de pause
AFDPSF	Fin pause
AFDLIB	Libellé

AFDSOC	Code société
AFDIND	Indice
AFDAGE	Code agence
AFDCLI	Code client
AFDAFF	Code chantier
AFDCL1	Div 1
AFDCL2	Div 2
AFDCL3	Div 3

Fonction sem : la semaine de calendrier

La fonction sem donne la semaine de calendrier, avec les jours fériés en fonction du département.

Exemple d'appel : <http://host/V1/sem?sem=201952&dep=67>

Les paramètres :

- L'année et le numéro de semaine "sem" au format AAAASS obligatoire
- Le département "dep" facultatif. En cas d'absence, utilise le département de l'entreprise.

Réponse

En réponse, vous recevez une structure JSON avec les membres "response" et "message", ainsi que 7 structures pour chaque jour de la semaine, identifiées de "1" à "7". Le "1" représente le lundi de la semaine reçue en paramètre. Pour chaque jour, "date" vous donne la date du jour, "ferie" indique si le jour est férié, et "nom" vous donne le nom en clair du jour férié.

Exemple de réponse :

```
{
  '1': { date: '2019-12-23', ferie: false, nom: '' },
  '2': { date: '2019-12-24', ferie: false, nom: '' },
  '3': { date: '2019-12-25', ferie: true, nom: 'Noël' },
  '4': { date: '2019-12-26', ferie: true, nom: 'Saint Etienne' },
  '5': { date: '2019-12-27', ferie: false, nom: '' },
  '6': { date: '2019-12-28', ferie: false, nom: '' },
  '7': { date: '2019-12-29', ferie: false, nom: '' },
  response: 'OK',
  message: ''
}
```

Fonction rhsem : Squelette de la semaine de contrat

La fonction rhsem s'utilise en méthode GET et permet de disposer de la composition théorique d'une semaine d'un contrat. Elle prend en compte le cycle client pour déterminer la semaine.

Exemple d'appel : <http://host/V1/rhsem?soc=001&age=130&smt=201914&ctr=103538>

Paramètres :

- Code société "soc" obligatoire
- Code agence "age" obligatoire
- Numéro de contrat "ctr" obligatoire
- Numéro de semaine "smt" obligatoire, au format AAAASS

Retour de la fonction

En retour, on obtient une structure JSON avec les membres "response" et "message", DEB et FIN pour les dates de début et de fin du contrat, et JOUR un tableau de structures JSON qui donne pour chacun des jours de la semaine :

- MOI pour le mois au format AAAAMM
- SMT pour le n° de semaine au format AAAASS
- DATE pour la date du jour
- Ferie à true ou false
- Nom qui prend comme valeur le nom du jour s'il était férié
- NJO pour le numéro du jour dans la semaine, 1 = lundi
- HOR pour le code de l'horaire du client appliqué pour ce jour
- HDE pour l'heure de début dans la journée
- HDI pour l'heure de fin dans la journée
- NBR pour le nombre d'heures en journée
- NBH pour le nombre d'heures de nuit
- FBR pour le nombre d'heure en journée si le jour était férié
- FBH pour le nombre d'heure de nuit si le jour était férié
- Statut :
 - H = Hors contrat
 - T = Théorique

```

{
  "response": "OK",
  "message": "Cycle 001, semaine de cycle 01",
  "DEB": "2019-04-02",
  "FIN": "2019-04-02",
  "JOUR": [
    {
      "MOI": "201904",
      "SMT": "201914",
      "DATE": "2019-04-01",
      "Ferie": false,
      "nom": "",
      "NJO": 1,
      "HOR": "",
      "HDE": "",
      "HDI": "",
      "NBR": "",
      "NBH": "",
      "FBR": "",
      "FBH": "",
      "Statut": "H"
    },
  ],
}
```

Ce qui donne pour le tableau JOUR dans cet exemple :

MOI	SMT	DATE	Ferie	nom	NJO	HOR	HDE	HDI	NBR	NBH	FBR	FBH	Statut
'201904'	'201914'	'2019-04-01'	false	''	1	''	''	''	''	''	''	''	'H'
'201904'	'201914'	'2019-04-02'	false	''	2	'01'	8	17	7	0	0	0	'T'
'201904'	'201914'	'2019-04-03'	false	''	3	''	''	''	''	''	''	''	'H'
'201904'	'201914'	'2019-04-04'	false	''	4	''	''	''	''	''	''	''	'H'
'201904'	'201914'	'2019-04-05'	false	''	5	''	''	''	''	''	''	''	'H'
'201904'	'201914'	'2019-04-06'	false	''	6	''	''	''	''	''	''	''	'H'
'201904'	'201914'	'2019-04-07'	false	''	7	''	''	''	''	''	''	''	'H'

Fonction rhimport : les RH présents en RHIMPORT

En méthode GET, la fonction rhimport permet de consulter le contenu de la table RHIMPORT

Exemple d'appel : <http://host/V1/rhimport?soc=001&age=130&smt=201838&ctr=102228>

Les paramètres sont :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- Le numéro de semaine "smt" au format "AAAASS" obligatoire
- Le numéro de contrat "ctr" facultatif

En retour, on reçoit une structure JSON comportant les membres "response" et "message", ainsi qu'un tableau RHImport de structures JSON. Chacune reprend comme membre les rubriques de RHIMPORT avec le même nom.

Le membre RHIIND à la valeur "RHR" représente les enregistrements qui sont refusés à l'enregistrement dans ANAEL, les explications du refus sont lisibles dans RHIMSG.

Exemple de retour :


```
{
  response: 'OK',
  message: '',
  RHImport: [
    {
      RHISOC: '001',
      RHIIND: 'RHI',
      RHIAGE: '130',
      RHICLI: '100152',
      RHIAFF: '100171',
      RHIPER: '130100195',
      RHICTR: '102228',
      RHISMT: '201838',
      RHIDAT: '2018-09-17',
      RHINBJ: 1,
      RHINHJ: 11.43,
      RHINHN: 0,
      RHIMSG: '',
      RHIDCR: '2018-09-25',
      RHIHCR: '1438',
      RHIRE1: '',
      RHIRE2: '',
      RHIRE3: '',
      RHIRE4: '',
      RHIRE5: '',
      RHIRE6: '',
      RHIRE7: 0,
      RHIRE8: 0,
      RHIORG: 'P'
    }
  ],
}
```

RHISOC	Code cociété
RHIIND	Indice RHI ou RHR (rejet)
RHIAGE	Code agence
RHICLI	Code client
RHIAFF	Code chantier
RHIPER	Matricule personnel
RHICTR	Numéro de contrat
RHISMT	Semaine format AAAASS

RHIDAT	Date
RHINHJ	Nombre d'heures de jour
RHINHN	Nombre d'heures de nuit
RHIMSG	Message En cas de rejet, le processus peut indiquer une information sur la raison du rejet
RHIDCR	Date de création
RHIHCR	Heure de création
RHIRE1	Demande d'acompte : "O" "N"
RHIRE2	Autres heures que des heures de jour ou de nuit
RHIRE3	Non validé = "O"
RHIRE4	Numéro complémentaire
RHIRE5	
RHIRE6	
RHIRE7	
RHIRE8	
RHIORG	Origine : "E" pour extranet
RHINBJ	Numéro de jour dans la semaine 1 = lundi

Fonction rhimport en POST : pour importer

En méthode POST, la fonction rhimport permet d'importer des relevés d'heures.

Exemple d'appel : <http://host/V1/rhimport>

En méthode POST, l'url ne prend pas de paramètre : les données sont attendues sous forme d'une structure JSON dans le corps de la requête.

Paramètre.

Le paramètre attendu est une structure JSON disposant des membres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- Le numéro de contrat "ctr" obligatoire
- Le numéro de semaine "smt" obligatoire
- Le type d'heure "ahe" s'il s'agit d'autre chose que des heures de jour ou de nuit. Dans ce cas, la valeur "A" y est attendue. Peut être transmis sans valeur.
- L'indicateur d'heures non validées "nva" attendu à "true" ou "false" obligatoire
- Le numéro complémentaire "ncomp". Peut être transmis sans valeur.
- L'indicateur de demande d'acompte "acompte" à la valeur "true" ou "false" obligatoire
- Un tableau "RH" de 7 structures, obligatoire. Chaque structure est composée des membres :
 - "nbj", pour le numéro du jour de la semaine, obligatoire, de 1 à 7
 - "jh1" pour le nombre d'heures de jour, avec une valeur numérique
 - "nh1" pour le nombre d'heures de nuit avec une valeur numérique
 - "date" pour la date du jour au format AAAA-MM-JJ

Exemple de structure attendue :

```
{
  soc: '001',
  age: '130',
  ctr: '103798',
  smt: '201920',
  ahe: '',
  nva: true,
  ncomp: '',
  acompte: false,
  RH: [
    { NBJ: 1, JH1: 7, NH1: 0, DAT: '2019-06-17' },
    { NBJ: 2, JH1: 7, NH1: 0, DAT: '2019-06-18' },
    { NBJ: 3, JH1: 7, NH1: 0, DAT: '2019-06-19' },
    { NBJ: 4, JH1: 7, NH1: 0, DAT: '2019-06-20' },
    { NBJ: 5, JH1: 7, NH1: 0, DAT: '2019-06-21' },
    { NBJ: 6, JH1: 0, NH1: 0, DAT: '2019-06-22' },
    { NBJ: 7, JH1: 0, NH1: 0, DAT: '2019-06-23' }
  ]
}
```

Réponse à la fonction

La réponse à la fonction est une structure JSON comportant les membres "response" et "message"

Fonction rh : lecture des relevés

La fonction rh s'utilise en méthode GET et permet de disposer des relevés d'une semaine de contrat

Exemple d'appel <http://host/V1/rh?soc=001&age=130&smt=201920&ctr=103798>

Paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- Le numéro de contrat "ctr" obligatoire si le numéro de client est absent. La recherche par numéro de contrat est prioritaire sur la recherche par client.
- Le numéro de semaine "smt" obligatoire
- Le numéro de semaine "smt2" facultatif. Permet de créer un intervalle de recherche des contrats concernés allant du lundi de la semaine smt au dimanche de la semaine de smt2
- Le code client "cli" obligatoire en cas d'absence du code contrat
- Le code chantier "aff" facultatif, pris en compte en cas de présence de "cli" et absence de "ctr", pour affiner la recherche sur un chantier.

Retour de la fonction

Le retour de la fonction est une structure JSON comprenant les membres "response" et "message", ainsi qu'un tableau CTR de structures.

Chaque élément du tableau reprend des informations issues du contrat, de l'intérimaire et du client, et un tableau RH de structures JSON.

Informations issues du contrat :

Membre	Rubrique d'origine	Domaine
SOC	ETTCTCP.CTCSOC	Contrat
AGE	ETTCTCP.CTCAGE	
CTR	ETTCTCP.CTCCTR	
CLI	ETTCTCP.CTCCLI	
AFF	ETTCTCP.CTCAFF	
MAT	ETTCTCP.CTCPER	

Membre	Rubrique d'origine	Domaine
FMO	ETTCTCP.CTCFMO	
MOT	ETTCTCP.CTCMOT	
QUA	ETTCTCP.CTCQUA	
DEB	ETTCTCP.CTCDEB	
FIN	ETTCTCP.CTCFIN	
REL	ETTCTCP.CTCREL	
XFI	ETTCTCP.CTCXFI	
XF2	ETTCTCP.CTCXF2	
TIP	PERPERP.PERTIP	Personnel
PRE	PERPERP.PERPRE	
NOM	PERPERP.PERNOM	
RSO	INSTIAP.TIARSO	Chantier
RU1	INSTIAP.TIARU1	
RU2	INSTIAP.TIARU2	
CPO	INSTIAP.TIACPO	
VIL	INSTIAP.TIAVIL	

Chaque élément du tableau RH est une structure JSON comprenant comme membres des rubriques de la table HREHR1 qui représente l'enregistrement d'entête des relevés d'heure, et un tableau "H" qui reprend en membre des enregistrements de HREJOU concernés.

Membre	Rubrique d'origine	Complément d'information
HR1SO	HREHR1.HR1SOC	Code société
HR1AG	HREHR1.HR1AGE	Code agence
HR1CT	HREHR1.HR1CTR	Numéro de contrat
HR1SM	HREHR1.HR1SMT	Semaine au format AAAASS
HR1JT	HREHR1.HR1JTQ	Partie de semaine
HR1PA	HREHR1.HR1PAG	Agence du personnel
HR1PE	HREHR1.HR1PER	Matricule du personnel
HR1AG	HREHR1.HR1AGT	Agence de traitement
HR1FM	HREHR1.HR1FMO	Motif de fin de mission
HR1PR	HREHR1.HR1PR1	Paie des IFM ^\$ [O] [N]
HR1PR	HREHR1.HR1PR2	Paie des IFM 2 ^\$ [Z]
HR1BH	HREHR1.HR1BHT	Type de RH

Membre	Rubrique d'origine	Complément d'information
HR1RE	HREHR1.HR1REL	Fin réelle
HR1MO	HREHR1.HR1MOE	Motif écart
HR1TP	HREHR1.HR1TP1	Demande d'IFM
HR1TP	HREHR1.HR1TP2	Demande d'ICP
HR1TP	HREHR1.HR1TP3	Statut du paiement des IFM et CP
HR1PL	HREHR1.HR1PLO	Lot de paie
HR1FL	HREHR1.HR1FLO	Lot de facturation

Chaque élément du tableau "H" est une structure qui reprends les enregistrements de la table HREJOU.

Membre	Rubrique d'origine	Complément d'information
HRENB	HREJOU.HRENB	Numéro jour dans la semaine
HREDAT	HREJOU.HREDAT	Date
HREHDD	HREJOU.HREHDD	Heure de début
HREHFD	HREJOU.HREHFD	Heure de fin
HREJH1	HREJOU.HREJH1	Heures de jour validées
HRENH1	HREJOU.HRENH1	Heure de nuit validées
HREJH2	HREJOU.HREJH2	Heures de jour prévues
HRENH2	HREJOU.HRENH2	Heures de nuit prévues
HREJHF	HREJOU.HREJHF	Heures de jour facturées
HRENHF	HREJOU.HRENHF	Heures de nuit facturées
HREQUA	HREJOU.HREQUA	Code qualification
HRETXP	HREJOU.HRETXP	Taux de paie
HRETXF	HREJOU.HRETXF	Taux de facturation
HRETYP	HREJOU.HRETYP	Type d'enregistrement
HREAOB	HREJOU.HREAOB	A=Maitre,B=Contremaitre
HREEQ2	HREJOU.HREEQ2	Equipe précédente
HRELI2	HREJOU.HRELI2	Ligne précédente
HRESL2	HREJOU.HRESL2	Sous ligne précédente
HRECP2	HREJOU.HRECP2	Compteur précédent

Exemple de réponse :

```

▼ {
  "response": "OK",
  "message": "",
  "CTR": [
    ▼ {
      "RH": [
        ▼ {
          "H": [
            ▶ { 19 items },
            ▶ { 19 items },
            ▶ { 19 items },
            ▶ { 19 items },
            ▼ {
              "HRENB": 5,
              "HREDAT": "2019-04-05",
              "HREHDD": "",
              "HREHFD": "",
              "HREJH1": 6.5,
              "HRENH1": 0,
              "HREJH2": 0,
              "HRENH2": 0,
              "HREJHF": 0,
              "HREHNF": "0",
              "HREQUA": "",
              "HRETXP": 0,
              "HRETXF": 0,
              "HRETFP": "1",
              "HREA08": "",
              "HREEQ2": "",
              "HRELI2": "",
              "HRESL2": "",
              "HRECP2": ""
            }
          ],
          "HRISOC": "001",
          "HRIAGE": "130",
          "HR1CTR": "103532",
          "HR1SMT": "201914",
          "HR1JTQ": "",
          "HR1PAG": "130",
          "HR1PER": "130100374",
          "HR1AGT": "130",
          "HR1FMO": "32",
          "HR1PR1": "0",
          "HR1PR2": "",
          "HR1BHT": "4",
          "HR1REL": "2019-04-05",
          "HR1MOE": "",
          "HR1TP1": "",
          "HR1TP2": "",
          "HR1TP3": "",
          "HR1PLO": "0890029",
          "HR1FLO": "0890032"
        }
      ],
    }
  ],
}

```

Fonction rhfinancier : lecture du financier des relevés d'heures

La fonction rhfinancier s'utilise en méthode GET et permet de disposer du financier des relevés d'une semaine de contrat

Exemple d'appel <http://host/V1/rhfinancier?soc=001&age=001&ctr=103789&sem=202420>

Paramètres :

- Le code société "soc" obligatoire
- Le code agence "age" obligatoire
- Le numéro de contrat "ctr" obligatoire.
- Le numéro de semaine "smt" obligatoire (format AAAAMM)
- La partie de semaine "jtg" facultatif (pour les semaines tronquées, à cheval sur deux mois = 1 ou 2) (par défaut = 1)
- Le numéro de rubrique "rub" facultatif : si le numéro de rubrique est renseigné, il faut renseigner aussi la partie de semaine, notamment pour les rubriques sur la 2eme partie de semaine

Retour de la fonction

Le retour de la fonction est une structure JSON comprenant les membres "response" et "message", ainsi qu'un tableau « lignefinancier » de structures.

Chaque élément du tableau reprend des informations issues du financier des relevés d'heures du contrat pour la semaine sélectionnée. Les données proviennent du fichier RELPRXP

Exemple de réponse :

```

{
  "response": "OK",
  "message": "",
  "ligneFinancier": [
    {
      "ET2SOC": "001",
      "ET2AGE": "001",
      "ET2PER": "001100043",
      "ET2CTR": "100069",
      "ET2AGT": "001",
      "ET2SMT": "201514",
      "ET2JTQ": "1",
      "ET2RUB": "001",
      "ET2CPT": "101",
      "ET2PBS": "16",
      "ET2PTX": "12.5",
      "ET2PMT": "200",
      "ET2FBS": "16",
      "ET2FTX": "26",
      "ET2FMT": "416",
      "ET2CLI": "",
      "ET2CL5": "",
      "ET2AFF": "",
      "ET2AF1": "",
      "ET2AF2": "",
      "ET2TYD": "",
      "ET2MOE": "",
      "ET2PLO": "0150002",
      "ET2FLO": "0150001",
      "ET2TF1": "",
      "ET2CC1": "",
      "ET2CD1": "",
      "ET2LB1": ""
    }
  ],
},

```

Fonction planning : le planning du client

Cette fonction permet de retrouver le planning des contrats d'un client entre deux dates. Le résultat est sonné en jour de contrat, accompagné du nombre d'heures pour chacun.

Exemple d'appel :

<http://host/V1/planning?soc=001&age=130&cli=100152&deb=2019-04-01&fin=2019-05-01&hsai=true>

Paramètres :

- Code société "soc" obligatoire

- Code agence "age" obligatoire
- Code client "cli" obligatoire
- Code chantier "aff" optionnel.
- Date de début "d1" facultative. En cas d'absence, prend le 1^{er} janvier de l'année en cours.
- Date de fin "d2" facultative. En cas d'absence prend la date du jour.
- Flag des heures saisie "hsai" optionnel. Sa présence avec la valeur "true" indique que l'on demande que la réponse comporte les informations d'heures présentes dans RHIMPORT.

Retour de la fonction

En retour, la fonction renvoie une structure JSON comportant les membres "message" et "response", ainsi qu'un tableau JPLAN de structures JSON. Chacune de ces structures représente une journée de contrat.

Rubrique	Origine	Remarque
SOC	ETTCTCP. CTCSOC	Code société
AGE	ETTCTCP. CTCAGE	Code agence
CLI	ETTCTCP. CTCCLI	Code client
AFF	ETTCTCP. CTCAFF	Code chantier
AFFLIB	INSTIAP.TIARSO	Raison sociale
AFFADR	INSTIAP.TIARU1 ... INSTIAP.TIAVIL	Adresse complète
CTR	ETTCTCP. CTCCTR	Numéro de contrat
CHG	ETTCTCP. CTCCHG	Code du chargé d'affaire
AVE	ETTCTCP. CTCAVE	N° d'avenant
DEB	ETTCTCP. CTCDEB	Date de début de contrat
FIN	ETTCTCP. CTCFIN	Date de fin de contrat
FIR	ETTCTCP. CTCREL	Date réelle de fin de contrat
SP1	ETTCTCP. CTCXFI	
SP2	ETTCTCP. CTCXF2	
MOT	ETTCTCP. CTCMOT	Code motif de fin de mission
MOTLIB		Libellé du motif de fin de mission
MAT	ETTCTCP. CTCPER	Matricule
NOM	PERPERP.PERNOM	Nom
PRE	PERPERP.PERPRE	Prénom
Date		Date du jour de contrat
NBJ		Numéro du jour dans la semaine
SMT		Numéro de semaine
SHJ	RHImport.RHINHJ	Nombre d'heures de jour saisies
SHN	RHImport.RHINHJ	Nombre d'heures de nuit saisies
VHJ	HREJOU.HREJH1	Nombre d'heures de jour validées
VHN	HREJOU.HRENH1	Nombre d'heures de nuit validées
PHJ	AFFAFDP.AFDNBR	Nombre d'heures de jour prévues
PHN	AFFAFDP.AFDNBR	Nombre d'heures de nuit prévues
SBLQ		Indicateur RH non modifiable

Exemple de retour :

```
▼ {
  "response": "OK",
  "message": "19 contrat(s)",
  ▼ "JPLAN": [
    ▼ {
      "SOC": "001",
      "AGE": "130",
      "CLI": "100152",
      "AFF": "100171",
      "AFFLIB": "",
      "AFFADR": "",
      "CTR": "103544",
      "AVE": "",
      "CHG": "FMI",
      "DEB": "2019-04-04",
      "FIN": "2019-04-06",
      "FIR": "2019-04-05",
      "SP1": "2019-04-05",
      "SP2": "2019-04-05",
      "MOT": "32",
      "MOTLIB": "FIN DE MISSION T.T",
      "MAT": "130100187",
      "NOM": "",
      "PRE": "",
      "Date": "2019-04-04",
      "NBJ": 4,
      "SMT": "201914",
      "SHJ": "",
      "SHN": "",
      "VHJ": 4,
      "VHN": 0,
      "PHJ": 7,
      "PHN": 0,
      "SBLQ": "1"
    },
  ],
}
```

Chapitre 5 Les commandes

	Personnel	Client	Commande	Relevés d'heures	Autre	Contrats	
/V1/cde_affectation			X				Affectation d'une commande
/V1/cde_affectationraz			X				Suppression d'une affectation
/V1/cde_cde			X				Lecture commande
/V1/cde_fin			X				Financier d'une commande
/V1/cde_lst			X				Liste des commandes
/V1/cde_md1			X				Modèle de poste
/V1/cde_set			X				Enregistrement de commande
/V1/cde_test			X				Pour tests

Fonction cde_lst pour la liste des commandes

La fonction cde_lst permet de disposer de la liste des lignes de commandes d'un chantier ou d'un client entre deux dates.

Exemple d'appel :

http://host/V1/cde_lst?soc=001&age=350&cli=000539&aff=&d1=20190401&d2=20190430

Paramètres :

- Code société "soc" obligatoire
- Code agence "age" obligatoire
- Code client "cli" obligatoire
- Code chantier "aff" optionnel.
- Début de l'intervalle "d1" optionnel. En cas d'absence, d1 prend la valeur du premier janvier de l'année civile. Il peut être exprimé en numéro de semaine dans le format AAAASS ou en

date dans le format AAAA-MM-JJ. Si un numéro de semaine est transmis, d1 prend la valeur du lundi de cette semaine

- Début de l'intervalle "d2" optionnel. En cas d'absence, d2 prend la valeur du 31 décembre de l'année civile. Il peut être exprimé en numéro de semaine dans le format AAAASS ou en date dans le format AAAA-MM-JJ. Si un numéro de semaine est transmis, d2 prend la valeur du dimanche de cette semaine
- Le code du chargé d'affaire "chg" optionnel.

Retour

En retour, la fonction renvoie une structure JSON comprenant les membres "response" et "message", ainsi qu'un tableau CDE de structures JSON d'informations sur les lignes de commande.

Description des membres :

Membre	Origine	Remarque
CTCSOC	MDLCTCP.CTCSOC	Code société
CTCAGE	MDLCTCP.CTCAGE	Code agence
CTCDEB	MDLCTCP.CTCDEB	Date début
CTCFIN	MDLCTCP.CTCFIN	Date fin
CTCCHG	MDLCTCP.CTCCHG	Code chargé d'affaire
CTCCLI	MDLCTCP.CTCCLI	Code client
CTCAFF	MDLCTCP.CTCAFF	Code chantier
CTCPER	MDLCTCP.CTCPER	Matricule
CTCSTA	MDLCTCP.CTCSTA	Statut de la ligne de commande
CTCCTR	MDLCTCP.CTCCTR	Numéro de commande
CTCAVE	MDLCTCP.CTCAVE	Numéro de ligne de commande
CTCMEX	MDLCTCP.CTCMEX	Numéro de contrat

Fonction cde_cde : lecture d'une commande

La fonction cde_cde permet de disposer des informations d'une commande : son enregistrement d'entête, ses lignes, son financier.

Exemple d'appel : http://host/V1/cde_cde?soc=001&age=350&cde=101097

Paramètres :

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

Retour

Retour envoyé par la fonction est une structure JSON comportant les membres "response", "message", ENT pour l'enregistrement d'entête de la commande, un tableau LIG de lignes de commande, un tableau FIN pour les lignes du financier de la commande

```
{
  "response": "OK",
  "message": "",
  "FIN": [ 3 items ],
  "LIG": [ 3 items ],
  "ENT": { 51 items }
}
```

Pour une commande identifiée par les paramètres donnés dans l'appel, ENT correspond à l'enregistrement de la table ETTCTAP trouvé pour cette commande, le tableau LIG comprend les lignes trouvées dans MDLCTCP et MDLCTDP, et dans FIN les lignes de MDLCTEP.

Dans un élément du tableau LIG, chaque ligne de MDLCTCP de la commande est enregistrée avec sa ligne correspondante de la table MDLCTDP. Les rubriques de ces deux tables y sont présentes dans la même structure.

Particularité du financier est de réunir dans le même tableau les enregistrements de MDLCTEP pouvant correspondre à différentes lignes de commande. Chaque ligne de commande est référencée par leur numéro de ligne, matérialisée par MDLCTCP.CTCAVE, MDLCTDP.CTDAVE et MDLCTEP.CTEAVE. La première ligne porte le code "01", "02" pour la seconde, etc.

Fonction cde_fin GET : lecture du financier

En méthode GET La fonction cde_fin permet de disposer du financier d'une commande

Exemple d'appel : http://frprnjdcarre/V1/cde_fin?soc=001&age=350&cde=100025&ave=

Paramètres :

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

- Code commande "cde" obligatoire
- Numéro de ligne de commande "ave" facultatif.

Retour

Le retour de la fonction est une structure JSON comportant les membres "message" et "response" ainsi qu'un tableau FIN de structures JSON. Chaque élément du tableau reprend les membres de MDLCTEP, plus le libellé CTELIB associé à la rubrique CTERUB.

Exemple de réponse :

```
{
  "response": "OK",
  "message": "",
  "FIN": [
    {
      "CTESOC": "001",
      "CTEIND": "CDE",
      "CTEAGE": "350",
      "CTECTR": "100025",
      "CTEAVE": "01",
      "CTELIG": "",
      "CTESLI": "",
      "CTEDEB": "",
      "CTEFIN": "9999-12-31",
      "CTEJOD": 0,
      "CTEJOF": 7,
      "CTERUB": "001",
      "CTELIB": "HEURES NORMALES",
      "CTEBAS": 0,
      "CTETXP": 9.53,
      "CTETXF": 1.98,
      "CTEMTP": 0,
      "CTEACT": "",
      "CTEDCR": "2014-12-03",
      "CTEDMO": "2014-12-03",
      "CTEDSU": "",
      "CTEEXL": "999"
    }
  ]
}
```

Fonction cde_fin POST : Ajout dans le financier

En méthode POST, la fonction cde_fin permet d'ajouter une ligne dans le financier d'une ligne de commande.

Exemple d'appel :

http://host/V1/cde_fin?soc=001&age=350&cde=100025&ave=&rub=002&bas=0&tp=1.25&tf=2

Paramètres :

- Code société "soc" obligatoire
- Code agence "age" obligatoire
- Code commande "cde" obligatoire
- Numéro de ligne de commande "ave" facultatif. En cas d'absence, prend la valeur "01"
- Code rubrique "rub" obligatoire
- Base "bas" facultative. Vaut 0 en cas d'absence
- Taux payé, numérique, facultatif. En cas d'absence, vaut 0.
- Taux facturé, numérique, facultatif. En cas d'absence, vaut 0.
- Montant permanent, "mp" facultatif. En cas d'absence, vaut 0.

Réponse

En réponse, la fonction donne une structure JSON comportant les membres "response" et "message".

Fonction cde_md1 : recherche de modèle de poste

La fonction cde_md1 permet la recherche de modèle de poste. Cette étape est un préalable à l'enregistrement d'une commande utilisant un modèle de poste.

Exemple d'appel :

http://host/V1/cde_md1?soc=001&age=350&cli=100056&aff=&qua=CAR001&mot=32&mdlqua=3&mdlmot=false

Paramètres :

- Code société "soc" obligatoire
- Code agence "age" obligatoire
- Code client "cli" obligatoire
- Code chantier "aff" facultatif. En cas de présence, la recherche ne porte que sur les profils de poste spécifiques au chantier du client.
- Code qualification "qua"
- Code motif de contrat "mot"

- Code "mdlqua", facultatif, pour le mode de recherche en fonction de la qualification. En cas d'absence, prend la valeur "1"
 - "1" pour une recherche des modèles à même qualification
 - "2" pour une recherche des modèles sans qualification ou avec même qualification
 - "3" pour tous les modèles de postes du client
- Code "mot", facultatif, pour la correspondance du motif de recours. Prend la valeur "false" en cas d'absence.
 - "false" : tous les motifs
 - "true" : le motif doit correspondre

Retour

En retour, la fonction renvoie outre les membres "response" et "message", un tableau MLD des différents modèles de poste pouvant correspondre à la recherche. Les membres sont ceux de MDLCTCP dont ils gardent comme nom de rubrique des 3 derniers caractères des rubriques de MDLCTCP, et quelques libellés.

```

{
  "MDL": [
    { 15 items },
    {
      "QUA": "CHA411",
      "QUALIB": "CHAUFFEUR VL",
      "CTR": "100194",
      "LIB": "REMPLACEMENT",
      "MOT": "11",
      "TER": "P",
      "CYC": "001",
      "SCD": "01",
      "COE": 1.97,
      "SAL": 10.03,
      "IND": "CT1",
      "INDLIB": "CLIENT",
      "AGE": "130",
      "CLI": "100001",
      "AFF": ""
    },
    { 15 items },
    { 15 items },
    { 15 items },
    { 15 items },
    { 15 items }
  ],
  "response": "OK",
  "message": ""
}

```

QUA	Qualification
QUALIB	Libellé de qualification
CTR	Code du modèle de poste
LIB	Libellé du modèle de poste
MOT	Motif de recours
TER	Type de terme : P=précis
CYC	Cycle
SCD	Semaine de cycle a départ du contrat
COE	Coefficient
SAL	Salaire
IND	Indice de rattachement du modèle : au client ou au chantier
INDLIB	Libellé de l'indice de rattachement
CLI	Code client
AGE	Code agence

AFF	Code chantier
-----	---------------

Si un modèle a été retenu pour la création d'une ligne de commande, vous reporterez le membre CTR dans le membre MDL_CTR de la ligne de commande, et IND dans MDL_IND.

Fonction cde_set : l'enregistrement de commande

La fonction cde_set s'utilise en méthode POST pour enregistrer une commande client. Une structure JSON contenant les paramètres à fournir est attendue dans le corps de la requête.

L'enregistrement d'une commande consiste à transmettre en méthode post une structure de données contenant les informations de la commande à créer, modifier ou supprimer.

La commande peut soit être enregistrée en renseignant toutes les rubriques attendues, soit minimiser les informations transmises en indiquant un modèle de poste dont on va reprendre les informations, soit un mix des deux : un modèle de poste est indiqué, mais certaines informations sont fournies qui viendront se substituer aux informations du modèle de poste.

Schématiquement, chaque commande est composée d'un entête, et d'un tableau de lignes de commandes, dont la quantité de ligne est au moins égale à 1. Chaque ligne comporte un membre QTE qui indique la quantité de contrat unitaire que cette seule ligne de commande doit engendrer.

Structure à fournir

Une structure JSON est à fournir qui doit comporter les membres suivants. La mention "facultatif" désigne un membre dont la valeur serait laissée à blanc. La mention "obligatoire" désigne un membre dont la valeur doit être consistante.

- Code "verb", obligatoire, désigne l'action demandée, à choisir parmi C, M, A :
 - C pour une demande de création de commande
 - M pour une demande de modification de commande
 - A pour une demande d'annulation de commande
- Code société "soc" obligatoire
- Code agence "age" obligatoire
- Code client "cli" obligatoire
- Code chantier
- Code type de commande, "tyf" obligatoire. Correspond à l'un des types de commande présent dans votre paramétrage, et dont vous pouvez retrouver la liste par l'appel suivant : <http://host/V1/wsdirect?table=PARDIVP&clef=PARDIVL1&val=001TYK>
- Le numéro de commande "cde" : obligatoire pour une modification ou une annulation, laissé vide pour une création.
- Commentaire "mem", facultatif, encodé url
- L'interlocuteur chez le client "ref" (30 caractères maxi), facultatif
- Le rédacteur "user", facultatif
- La date de commande "DAT", facultatif
- Le code du chargé d'affaire "chg" obligatoire
- Motif annulation : MOTANN : obligatoire pour une annulation de commande (si verb = A)

- Un tableau LINE contenant des structures JSON pour chaque ligne de commande

Chaque ligne du tableau LINE est attendu au format suivant :

Si un modèle est transmis, alors les deux champs MDL_CTR et MDL_IND sont attendus avec respectivement le membre CTR et le membre IND du modèle renvoyé par la fonction CDE_MDL

Membre	Signification	Mention	Impacté par	Destination	Taille
CTCQUA	Code qualification	Obligatoire			6
CTCMOT	Code motif de recours	Obligatoire			2
CTCTER	Code du type de terme	Obligatoire			1
CTCDEB	Date de début	Obligatoire			8
CTCFIN	Date de fin prévue	Obligatoire			8
QTE					
CTCAVE	Numero de ligne de commande	Uniquement en modification pour déterminer quelle ligne doit être modifiée			
CTCHRE	Heure au format HHMM	Uniquement si heure différente des infos. de cycle et de semaine	MDL		4
CTCCHG	Code Chargé d'affaire	Obligatoire			3
CTCCH2	Code Chargé d'affaire 2	Facultatif			3
MDL_CTR					6
MDL_IND					3
CTCPER	Matricule, en cas de désignation d'un personnel à ce stade. En cas de QTE > 1, affecte uniquement la première ligne de commande en cas de quantité > 1	Facultatif			9
CTCTTI	Code insee taxe transport				7

Membre	Signification	Mention	Impacté par	Destination	Taille
CTCCDE	Numéro de commande donné par le client	Facultatif			4
CTCAFF	Code chantier	Obligatoire			6
CTCACT	Code activité	Obligatoire	MDL	MDLCTCP.CTCACT	1
CTCACS	Code sous-activité	Facultatif	MDL	MDLCTCP.CTCACS	1
CTCPON	Niveau	Facultatif	MDL	MDLCTCP.CTCPON	2
CTCPOE	Echelon	Facultatif	MDL	MDLCTCP.CTCPOE	2
CTCPOC	Coef	Facultatif	MDL	MDLCTCP.CTCPOC	3
CTCDUR	Durée Heb.		MDL	MDLCTCP.CTCDUR	4
CTCSAL	Salaire	Obligatoire, numérique	MDL	MDLCTCP.CTCSAL	8
CTCSNH	Nombre d'heure pour le salaire indiqué	Obligatoire, numérique	MDL	MDLCTCP.CTCSNH	4
CTCCYC	Code cycle du client	Obligatoire	MDL	MDLCTCP.CTCCYC	3
CTCSCD	Code semaine de cycle au départ du contrat	Obligatoire	MDL	MDLCTCP.CTCSCD	2
CTCCOE	Coefficient	Obligatoire, numérique	MDL	MDLCTCP.CTCCOE	4
CTCFVM	Fréquence de visite médicale	0	MDL	MDLCTCP.CTCFVM	2
CTCCAT	Code catégorie	Obligatoire	MDL	MDLCTCP.CTCCAT	3
CTCREF	Référence de la commande donnée par le client	0	MDL	MDLCTCP.CTCREF	30
CTCBHT	Type de relevé d'heures	0	MDL	MDLCTCP.CTCBHT	1
CTCTCT	Famille de contrat	0	MDL	MDLCTCP.CTCTCT	1
CTCTP1	Risque 'O' 'N'	0	MDL	MDLCTCP.CTCTP1	1
CTCTP2	Suivi médical 'O' 'N'	0	MDL	MDLCTCP.CTCTP2	1
CTCGES	Gestion ou délégation 'G' 'D'	0	MDL	MDLCTCP.CTCGES	1
CTCCC1	Forfait jour	0	MDL	MDLCTCP.CTCCC1	6
CTCRXS	Salaire de référence annuel	0	MDL	MDLCTCP.CTCRXS	8

Membre	Signification	Mention	Impacté par	Destination	Taille
CTCRXH	Ann Nb hres forfait		MDL	MDLCTCP.CTCRXH	4
CTCRXT	Ann paramètres		MDL	MDLCTCP.CTCRXT	15
CTDTX1	Justification ligne 1		MDL	MDLCTDP.CTDTX1	70
CTDTX2	Justification ligne 2		MDL	MDLCTDP.CTDTX2	70
CTDTX3	Caractéristique de la mission ligne 1		MDL	MDLCTDP.CTDTX3	70
CTDTX4	Caractéristique de la mission Ligne 2		MDL	MDLCTDP.CTDTX4	70
CTDTX5	Caractéristique de la mission Ligne 3		MDL	MDLCTDP.CTDTX5	70
CTDTX6	Equipement ligne 1		MDL	MDLCTDP.CTDTX6	70
CTDTX7	Equipement ligne 2		MDL	MDLCTDP.CTDTX7	70
CTDTX8	Détail risque		MDL	MDLCTDP.CTDTX8	70
CTCTX9_1	Personne demandée		MDL	Les 35 premiers car de MDLCTDP.CTDTX9	35
CTCTX9_2	Trajet		MDL	Les 35 car suivants de MDLCTDP.CTDTX9	35
CTDT10	Texte 35 heures ligne 1		MDL	MDLCTDP.CTDT10	70
CTDT11	Texte 35 heures ligne 2		MDL	MDLCTDP.CTDT11	70
CTDT12	Texte 35 heures ligne 3		MDL	MDLCTDP.CTDT12	70
CTCT13_1	Horaires		MDL	Les 35 prem car de MDLCTDP.CTDT13	35
CTCT13_2	Période non travaillée		MDL	Les 35 car suivants de MDLCTDP.CTDT13	35
CTCT14_1	Adresse 1		MDL ou chantier	25 premiers car de MDLCTDP.CTDT14	25
CTCT14_2	Adresse 2		MDL ou chantier	20 car suivants de MDLCTDP.CTDT14	20
CTCT14_3	Code postal		MDL ou chantier	5 car suivants de MDLCTDP.CTDT14	5
CTCT14_4	Ville		MDL ou chantier	2à car suivants de MDLCTDP.CTDT14	20
CTCT15_1	Libellé de qualification libre		MDL	25 premiers car de MDLCTDP.CTDP15	25

Membre	Signification	Mention	Impacté par	Destination	Taille
CTDT16	Installation ligne 1		MDL	MDLCTDP.CTDT16	70
CTDT17	Installation ligne 2		MDL	MDLCTDP.CTDT17	70
CTDT18	Complément salarial ligne 1		MDL	MDLCTDP.CTDT18	70
CTDT19	Complément salarial ligne 2		MDL	MDLCTDP.CTDT19	70
CTDT20	Contact		MDL	35 premiers car de MDLCTDP.CTDT20	70
CTDMEM	Mémo		MDL	MDLCTDP.CTDMEM	

La réponse de la fonction est une structure JSON comportant les membres "response", "message" et "CDE". Ce dernier donne en cas de succès le numéro de commande attribué.

En modification de ligne, si CTCAVE n'est pas renseigné, il y aura création d'une nouvelle ligne. Si CTCAVE est renseigné, il y aura modification de la ligne concernée.

MDL_CTR et MDL_IND permettent d'identifier le modèle de poste à utiliser pour la ligne de commande.

En cas d'utilisation d'un modèle de poste, tous les membres dont la colonne "impacté par" comportant la mention "MDL" reprendront la valeur présente dans le modèle de poste, sauf si une valeur a été transmise pour ce membre.

En cas d'utilisation d'un modèle de poste, en création de commande, le financier et les caractéristiques sont créés à partir du financier et des caractéristiques du modèle de poste.

Fonction cde_affectation : Affectation d'un personnel

La fonction cde_affectation permet d'affecter un personnel à une commande, et de faire évoluer les statuts de la ligne de commande et de la commande.

Exemple d'appel :

http://host/V1/cde_affectation?soc=001&age=114&cde=101097&ave=01&mat=114000001

Paramètres :

- Code société "soc" obligatoire

- Code agence "age" obligatoire
- Numéro de commande "cde" obligatoire
- Numéro de ligne de commande "ave" facultatif. En cas d'absence, prendra la valeur "01"
- Matricule de du salarié "mat" obligatoire

Réponse

En réponse, la fonction cde_affectation envoie une structure JSON comportant les membres "response" et "message".

Fonction cde_affectationraz

La fonction cde_affectationraz permet de supprimer l'affectation existante d'un personnel à une ligne de commande, et de mettre à jour le statut de la ligne de commande et de la commande.

Exemple d'appel : http://host/V1/cde_affectationraz?soc=001&age=114&cde=101097&ave=01

Paramètres :

- Code société "soc" obligatoire
- Code agence "age" obligatoire
- Numéro de commande "cde" obligatoire
- Numéro de ligne de commande "ave" facultatif. En cas d'absence, prendra la valeur "01"

Réponse

En réponse, la fonction cde_affectation envoie une structure JSON comportant les membres "response" et "message".

Chapitre 6 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/age1st					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/wsparm					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 :

```

✓ object {3}
  response : "OK"
  message : ""
  ✓ ADRESSE [1]
    ✓ 0 {14}
      AD1SOC : "001"
      AD1IND : "AGE"
      AD1AGE : "001"
      AD1COD : "001"
      AD1SIR : ""
      AD1TIT : ""
      AD1RSO : ""
      AD1RU1 : "13 rue de Paris"
      AD1RU2 : ""
      AD1RU3 : ""
      AD1RU4 : ""
      AD1CPO : "92150"
      AD1VIL : "Suresnes"
      AD1FIL : ""

```

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 :

```
✓ object {3}
  response : "OK"
  message : ""
  ✓ ADRESSE [1]
    ✓ 0 {14}
      ADDSOC : "001"
      ADDIND : "ORS"
      ADDAGE : "AGI"
      ADDMAT : ""
      ADDCOD : "001"
      ADDTIT : ""
      ADDRSO : "Retraite AGIRC IREPS"
      ADDR1 : "33 Quai Paul Doumer"
      ADDR2 : ""
      ADDR3 : ""
      ADDR4 : ""
      ADDCPO : "92672"
      ADDVIL : "Courbevoie"
      ADDFIL : ""
```

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", ainsi 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',
      CP0: '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) :

```
▼ {
  "response": "OK",
  "message": "",
  "dates": [
    ▼ {
      "date": "2020-01-01",
      "nom": "Jour de l'an",
      "n": 3
    },
    ▼ {
      "date": "2020-04-10",
      "nom": "Vendredi Saint",
      "n": 5
    },
    ▼ {
      "date": "2020-04-12",
      "nom": "Pâques",
      "n": 7
    },
    ▼ {
      "date": "2020-04-13",
      "nom": "Lundi de Pâques",
      "n": 1
    },
  ],
}
```

Fonction férie : un jour férié dans ce département ?

La fonction "ferié" indique en retour si 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 :

```
{
  "response": "OK",
  "message": "",
  "date": "2020-07-14",
  "ferie": true,
  "nom": "Fête nationale"
}
```

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

```
{
  "response": "OK",
  "message": "",
  "VIL": [
    {
      "CP": "71240",
      "INSEE": "71384",
      "Ville": "SAINT-AMBREUIL"
    },
    {
      "CP": "71300",
      "INSEE": "71390",
      "Ville": "SAINT-BERAIN-SOUS-SANVIGNES"
    },
    {
      "CP": "71640",
      "INSEE": "71403",
      "Ville": "SAINT-DENIS-DE-VALX"
    }
  ]
}
```

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 connaître 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

```
[ANAELETT]
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": "ANAELETT",
    "server": "192.168.1.1",
    "type": "WinDevClientServeurHF",
    "database": "ANAELETT",
    "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>

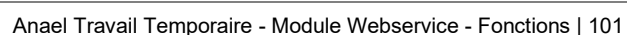
Paramètre :

- 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

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

Fonction wsdirect : l'accès aux données de la base

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": "000111100022200000110000015 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 7 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