

ServiceNav Intelligent insight software

API Web Service – Signatures

Version : 1.29

Créé le : 07/02/2017

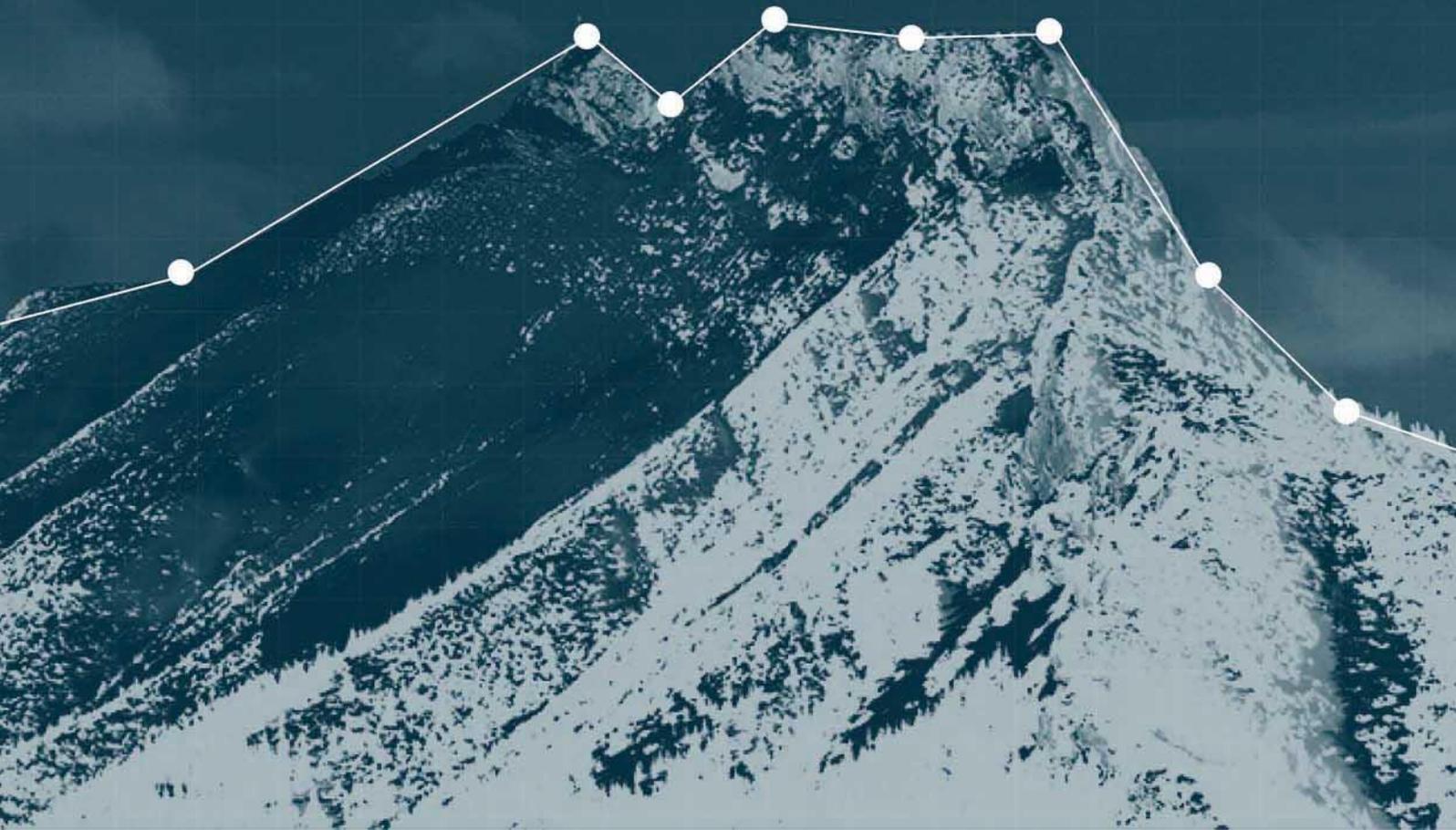


Table des matières

Table des matières	2
Avertissement	4
Champ d'application	4
Limitations de responsabilité	4
Copyright	4
1 Contrats des API Web service	5
1.1 Terminologie et logique de fonctionnement	5
1.2 Données génériques et typées	6
1.2.1 Gestion des erreurs	6
1.2.2 Types d'entité	6
1.2.3 Types d'attributs	7
1.2.4 Types d'équipements	7
1.2.5 Modules de services	7
1.2.6 Code icônes catégories d'équipements	7
1.2.7 Niveau de description	8
1.2.8 Couleurs du thème de l'application	8
1.2.9 KPI	8
1.2.10 Statuts des acquittements	8
1.2.11 Statuts des contrôles	8
1.2.12 LCA	9
1.2.13 Statuts des LCA	9
1.2.14 Statuts de Supervision	9
1.2.15 Statuts d'une notification	9
1.2.16 Tri	9
1.2.17 Relation entre éléments	9
1.2.18 Criticité d'une société	9
1.2.19 Criticité d'un CI	10
1.2.20 Statuts des entités	10
1.2.21 Système de l'application	10
1.2.22 Périmètre	10
1.3 Gestion de la pagination	11
1.4 Gestion de l'authentification	12
1.5 API Communes	13
1.5.1 API Authentification	13
1.5.2 API Déverrouillage	15
1.5.3 API Déconnexion	16
1.5.4 API Activation modules	17
1.5.5 API User	18
1.5.6 API Contact(s)	26
1.5.7 API Société(s)	31
1.5.8 API Thèmes	43
1.5.9 API Statistiques	45
1.6 API de Supervision	46

1.6.1	API Récupérer un historique de métriques.....	46
1.6.2	API Société(s)	48
1.6.3	API Equipements(s)	51
1.6.4	API Contact(s).....	67
1.6.5	API Service(s) Unitaire(s).....	69
1.6.6	API Service(s) Utilisateur	77
1.6.7	API Relations Service(s) Utilisateur	86
1.6.8	API Acquittements	90
1.6.9	API Contrôle	91
1.6.10	API Maintenance	91
1.6.11	API Reporting	94

Avertissement

Champ d'application

Cette documentation s'applique à l'ensemble des plateformes Coservit.

Limitations de responsabilité

L'usage des WebServices est effectuée sous responsabilité du Client. Coservit peut fournir un appui méthodologique ou technique dans le cadre d'un projet dont les modalités sont à définir entre le Client et Coservit.

La société Coservit ne saurait être tenue responsable d'un dysfonctionnement du produit ServiceNav si la cause en est l'usage des WebServices, ou toute autre modification apportée au produit en dehors de celles décrites dans la documentation utilisateur. Toute intervention de la société Coservit pour diagnostiquer ou réparer un dysfonctionnement ayant pour cause toute usage des WebServices ou modification apportée au produit en dehors de celles décrites dans la documentation utilisateur sera facturée en vertu du contrat établi avec le Client ou de la commande établie préalablement à l'intervention de Coservit.

L'ensemble des accès aux données présentes dans ServiceNav à travers les webservices sont sous la responsabilité du client.

Copyright

L'installation de tout fichier effectuée par le Client implique sa responsabilité quand au respect de la propriété intellectuelle en général, et du copyright en particulier. La responsabilité de la société Coservit ne saurait être engagée en cas de poursuite pour cause de non respect de la propriété intellectuelle concernant un fichier installé par le Client.

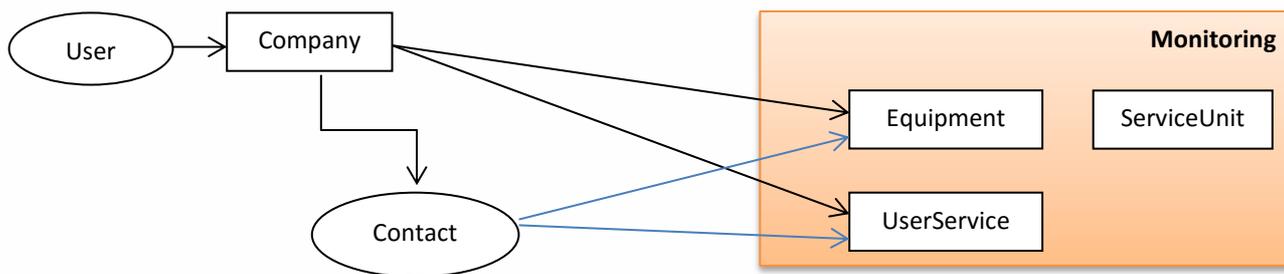
1 Contrats des API Web service

Ce paragraphe décrit les signatures pour chaque fonction de l'API.

D'une manière générale, la nomenclature utilisée est la suivante pour les fonctions de récupération de données (Equipements, Sociétés, Contacts, ...) :

- **getList<item_type>ShortDescription** : liste d'items renseignant un minimum d'informations
- **getList<item_type>FullDescription** : liste d'items renseignant l'ensemble des informations
- **getList<item_type><module>KPI** : liste des KPI d'une donnée, incluant leur état.

1.1 Terminologie et logique de fonctionnement



1.2 Données génériques et typées

Ce paragraphe regroupe l'ensemble des données servant à la communication avec l'API web service Mobilité.

1.2.1 Gestion des erreurs

Tout code http différent de 200 indique qu'une erreur a eu lieu.

Lorsqu'une erreur s'est produite, les attributs retournés seront :

Attributs de l'objet en sortie	Description
httpStatusCode	Code http de retour (utile si le consommateur REST ne peut pas parser les entêtes http).
messageCode	Code d'erreur applicatif
userMessage	Message à destination de l'utilisateur final
developerMessage	Message à destination du développeur du client REST
timestamp	Date de l'erreur
errors	Tableau contenant des informations supplémentaires sur l'erreur. Cete information est facultative Lors de la validation d'un formulaire, ce tableau d'erreur contient la liste des champs en erreur avec pour chaque champ un message d'erreur. Format : Tableau ([« champ1 »] => « Le champ champ1 est obligatoire », [« champ2 »] => « Le format du champ champ2 est invalide »)

1.2.2 Types d'entité

Donnée	Valeur
Contact	1
Utilisateur	2
Equipement	5
Notification	7
Service unitaire	8
Service utilisateur	9
Société	10
Module	11
Services Métiers	14
Élément de parc	15
Liste de choix	16
Maintenance	17
Modèle de service	18
Catégorie d'équipement	19

1.2.3 Types d'attributs

Donnée	Valeur
Recherche complète	0
ID	1
Nom	2
Statut	3
Kpi	4
reportStatus	5
Token	6
Supprimé	7
Date de mise à jour minimum	8
Durée de statut	10
Périmètre	11

1.2.4 Types d'équipements

Donnée	Valeur
box	box
agent	agent

1.2.5 Modules de services

Donnée	Valeur
Supervision	1
Ticketing	2

1.2.6 Code icônes catégories d'équipements

Donnée	Valeur
Autre	0
Serveur Windows	1
Serveur Linux	2
Serveur	3
Téléphone	4
Caméra	5
Onduleur	6
Switch	7
Routeur	8
Imprimante	9
SAN/NAS	10

1.2.7 Niveau de description

Paramètre « levelDescription ».

Niveau	Valeur
None	0
Short	1
Full	2

Pour les niveaux Short et Full, les informations remontées lorsque ce paramètre est défini sont indiquées dans les API `getList<item_type><level>Description.php`

1.2.8 Couleurs du thème de l'application

Niveau	Valeur
Foncé	1
Moyen	2
Clair	3
Barre d'outils	4

1.2.9 KPI

Paramètre « kpiFilter ».

Module	KPI	Valeur
Supervision	Utilisateur - Société	1
Supervision	Utilisateur - Equipement	2
Supervision	Utilisateur – Service Utilisateur	3
Supervision	Utilisateur – Service Unitaire	4
Supervision	Société - Equipement	5
Supervision	Société – Service Utilisateur	6
Supervision	Société – Service Unitaire	7
Supervision	Société – Activité de supervision	8
Supervision	Equipement – Service Unitaire	9
Supervision	Service utilisateur – Élément de configuration	24

1.2.10 Statuts des acquittements

Statut	Valeur
Non acquitté	0
Acquitté	1
En cours	2

1.2.11 Statuts des contrôles

Statut	Valeur
En cours	2

1.2.12 LCA

Paramètre « lcaFilter ». Valeurs issues de la base Pasi : table « topology »

LCA	Valeur
<i>topology::topology_name</i>	<i>topology:idtopology</i>

1.2.13 Statuts des LCA

LCA	Valeur
Actif	0
Non-actif	1

1.2.14 Statuts de Supervision

Statut	Valeur
Ok	0
Alerte	1
Critique	2
Inconnu	3
Indéterminé	4

1.2.15 Statuts d'une notification

Donnée	Valeur
Non-lu	0
Lu	1

1.2.16 Tri

Type de Tri	Valeur
Tri ascendant	0
Tri descendant	1

1.2.17 Relation entre éléments

Relation	Valeur
Dégradante	1
Bloquante	2

1.2.18 Criticité d'une société

Statut	Valeur
Non VIP	1
VIP	2

1.2.19 Criticité d'un CI

Statut	Valeur
Basse	0
Moyenne	1
Haute	2

1.2.20 Statuts des entités

Statut	Valeur
Non supprimé	0
Supprimé	1

1.2.21 Système de l'application

Système	Valeur
Android	android
iOS	ios

1.2.22 Périmètre

Système	Valeur
Organisation interne	1
Toutes les sociétés visibles	2

1.3 Gestion de la pagination

Toutes les APIs retournant une liste d'éléments peuvent être appelées avec des paramètres indiquant la pagination demandée.

Les informations de pagination retournées par le serveur sont situées dans le header http dans la balise `Content-Range`

Description de la balise :

`Content-Range : unit indexFirstItem-indexLastItem/ItemTotalNumberFiltered`

La valeur `unit` aura pour toutes nos APIs de liste la valeur `items`

Exemple :

- Les éléments retournés sont du 10^{ème} au 19^{ème} éléments sur un total de 200

`Content-Range : items 10-19/200`

- Si le nombre total d'éléments ne peut pas être fourni par le serveur, alors le caractère `*` représente le `ItemTotalNumberFiltered`

`Content-Range : items 10-19/*`

1.4 Gestion de l'authentification

L'authentification se fait via un couple login et mot de passe transmis via l'entête http « Authorization :Basic » encodée en base 64.

Si l'on veut garder une session active, il faut appeler l'API getSession.php qui retourne un cookie de session « PHPSESSID ». Ce cookie devra être utilisé dans les appels aux autres APIs afin de s'authentifier à la place du couple login / mot de passe.

1.5 API Communes

1.5.1 API Authentification

Nom de l'API : getSession.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
login	String		florian.letellier@coservit.fr
<i>provider</i>	String : contexte partenaire		
password	String		MyPassword
lang	String : préférence de langue (format : « pays_langue »)		FR_fr
app			
token	String : Token de l'application récupéré après inscription auprès du middleware de notification		
middlewareAppId	String : ID pushwoosh de l'application		
<i>alias</i>	String : nom de l'apk		
<i>version</i>	String : Version de l'application		
<i>type</i>	String : Système sur laquelle est installé l'application (cf §.1.2.21)		



L'attribut « lang » est transmis via l'entête http « Accept-Language ».

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
session	String : token de session	Abcdefghijklmnop12345
id	Entier : ID de l'utilisateur	5
externalUserId	Entier : ID de l'utilisateur lié dans le système externe. null si pas de liaison utilisateur	
firstName	String : Prénom de l'utilisateur	
lastName	String : Nom de l'utilisateur	
login	String : Adresse mail de login de l'utilisateur	

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "app": {
    "token" : "APA91bFqjwOw_CdKNKAM3tpCd8dOBROx"
    "middlewareAppId" : "669A1-T37JE"
  }
}
```

Exemple d'objet JSON en sortie du web service

```
{
  "session": "abcdefghijkl12345",
  "id": "5"
}
```

1.5.2 API Déverrouillage

Nom de l'API : unlockApplication.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String		Abcdefghijklmnop12345
<i>code</i>	String		MyPassword

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={  
  "code": "MyPass"  
}
```

Exemple d'objet JSON en sortie du web service

1.5.3 API Déconnexion

Nom de l'API : `removeSession.php`

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
session	String : token de session		Abcdefghijklmnop12345
app	Objet		
<i>token</i>	String : Token de l'application récupéré après inscription auprès du middleware de notification		

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service
<pre>param={ "app": { "token" : "APA91bFqjwOw_CdKNKAM3tpCd8dOBROx" } }</pre>

Exemple d'objet JSON en sortie du web service

1.5.4 API Activation modules

Cette API permet de vérifier l'activation des modules pour le client mobile.

Nom de l'API : getModulesPriviledge.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>id</i>	String : ID du module (cf. §1.2.5)	1
<i>status</i>	Booléen : statut d'activation du module	0 → inactif
<i>offline</i>	Booléen : indique si le module ticketing est compatible avec le mode offline 0 : non compatible 1 : compatible	0

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "1",
    "status": 1
  },
  {
    "id": "2",
    "status": 1,
    "offline": 0
  }
]
```

1.5.5 API User

1.5.5.1 Description courte

Nom de l'API : getListUserShortDescription.php

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nblItems</i>	Entier : nombre max d'items à récupérer	tous	10
<i>filter / sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>resource</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		2
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		7
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		[0]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>id</i>	Entier : ID de l'utilisateur	1
<i>lastName</i>	String : nom de l'utilisateur	Doe
<i>firstName</i>	String : prénom de l'utilisateur	John
<i>updatedAt</i>	Entier : timestamp de mise à jour	1391680932
<i>deletionStatus</i>	Entier : l'utilisateur est-il supprimé (voir §1.2.20)	0

Exemple d'objet JSON en entrée du web service

--

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id":1,
    "lastName":"Doe",
    "firstName":"John",
    "updatedAt":1391680932
    "deletionStatus":0
  },
  {
    "id":2,
    "lastName":"Doe",
    "firstName":"Jane",
    "updatedAt":1391680990
    "deletionStatus":1
  }
]
```

1.5.5.2 Monitoring KPI

Nom de l'API : getListUserMonitoringKPI.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des utilisateurs	Tous	
<i>kpiFilter</i>	Liste : filtre sur les KPI du module (cf. §1.2.9)	Pas de filtre	
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nblItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i><id></i>	Liste : ID de l'utilisateur	
<i>id</i>	Entier : ID de l'utilisateur	
<i>monitoringKPI : <kpi></i>	Liste des KPI (cf. §1.2.9)	
<i>value</i>	String : valeur du KPI	3+
<i>color</i>	String : couleur RGB en hexadécimal	000000
<i>total</i>	String : nb total d'item pour le KPI	10
<i>reportStatus</i>	Entier : statut du KPI (cf. §Erreur ! Source du renvoi introuvable.)	



Les données facultatives en sortie dépendent du niveau de description demandé (levelDescription)

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "123",
    "124"
  ],
  "kpiFilter": [
    "1",
    "2"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
{
  "123":{
    "id":"123",
    "monitoringKPI":{
      "1":{
        "value":"3+",
        "color":"FFFFFF",
        "total":"10"
      },
      "2":{
        "value":"3+",
        "color":"FFFFFF",
        "total":"10"
      }
    }
  },
  "124":{
    "id":"124",
    "monitoringKPI":{
      "1":{
        "value":"3+",
        "color":"FFFFFF",
        "total":"10"
      },
      "2":{
        "value":"3+",
        "color":"FFFFFF",
        "total":"10"
      }
    }
  }
}
```

1.5.5.3 Fonctions activées

Nom de l'API : `getListUserLCA.php`

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>lcaFilter</i>	Liste : filtre sur les LCA (cf. §1.2.10)	Pas de filtre	
<i>filter / sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>id</i>	Entier : ID de l'utilisateur	
<i>lca : <lca></i>	Liste des LCA (cf. §1.2.10)	
<i>name</i>	String : nom du LCA	WebService Mobility
<i>status</i>	Booléen : statut du LCA (cf. §1.2.13)	1

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "filter": [{
    "entity": "2",
    "attribut": "1",
    "values": ["124"]
  }],
  "lcaFilter": [
    "64",
    "76"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "123",
    "lca": {
      "64": {
        "name": "Reporting",
        "status": "0"
      },
      "76": {
        "name": "WebService Mobility",
        "status": "1"
      }
    }
  }, {
    "id": "124",
    "lca": [
      "64": {
        "name": "Reporting",
        "status": "1"
      },
      "76": {
        "name": "WebService Mobility",
        "status": "1"
      }
    ]
  }
]
```

1.5.6 API Contact(s)

1.5.6.1 Description courte

Nom de l'API : getListContactShortDescription.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des contacts (Couples d'ID pour les contacts <idMonitoring>,<idTicketing>)	Tous	["1,2","3,"]
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>id</i>	Couple : ID du contact sur ServiceNav, ID du contact sur CRM	"10,20"
<i>name</i>	String : nom du contact	Bruno Richoux
<i>function</i>	String : fonction du contact	Directeur Technique
<i>comm:<type></i>	Entier : Type de communication (cf. § Erreur ! Source du renvoi introuvable.)	
<i>value</i>	String : donnée correspondant au type (numéro de téléphone, mail...etc)	"+33 476700772"
<i>cat</i>	Entier : catégorie du type de communication (cf. § Erreur ! Source du renvoi introuvable.)	2
<i>default</i>	Booléen : 1 si coordonnées par défaut pour ce type de communication.	1

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "1,",
    ",2"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[{
  "id": "1,",
  "name": "Contact 1,",
  "function": "Directeur Technique",
},
{
  "id": ",2",
  "name": "Contact ,2",
  "function": "Directeur Technique",
}]
```

1.5.6.2 Description complète

Nom de l'API : getListContactFullDescription.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
session	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des contacts (Couples d'ID pour les contacts <idMonitoring>,<idTicketing>)	Tous	["1,2","3,"]
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter sorting</i>	Liste : Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
id	Couple : ID du contact sur ServiceNav, ID du contact sur CRM	"10,11"
name	String : nom du contact	Bruno Richoud
function	String : Service du contact	Ressources Humaines
<i>comm:<type></i>	Entier : Type de communication (cf. §Erreur ! Source du renvoi introuvable.)	
value	String : donnée correspondant au type (numéro de téléphone, mail...etc)	"+33 476700772"
<i>cat</i>	Entier : catégorie du type de communication (cf. §Erreur ! Source du renvoi introuvable.)	2
<i>default</i>	Booléen : 1 si coordonnées par défaut pour ce type de communication.	1
companyId	Couple : ID de la société sur VS, ID de la société sur CRM	"10","null"
companyName	String : nom de la société	Coservit
address	Objet	
<i>address</i>	String : adresse de la société	13 rue des trembles
<i>postcode</i>	String : code postale de la société	38100
<i>city</i>	String : ville de la société	Grenoble
<i>area</i>	String : région de la société	Rhône-Alpes

<i>country</i>	String : pays de la société	France
----------------	------------------------------------	--------

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "123, ",
    ", 124"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "123, ",
    "name": "Bruno Richoux",
    "function": "Directeur Technique",
    "companyId": {"123", "null"},
    "companyName": "Coservit",
    "address": {
      "address": "Champs-Élysée",
      "postcode": "75000",
      "city": "Paris",
      "area": "Ile-de-France",
      "country": "France",
    },
    "comm": [
      {
        "1": [{"value": "+33 476705708", "cat": "1"},
          {"value": "+33 676123962", "cat": "2", "default": "1"}
        ],
        "3": [{"value": "+33 476700772"}],
        "4": [{"value": prenom.nom@company.com}]
      }
    ],
  }, {
    "id": ", 124",
```

```
"name": "Rachid Mokhtari",
"function": "Directeur R&D",
"companyId": {"123", "null"},
"companyName": "Coservit",
"address": {
  "address": "Champs-Élysée",
  "postcode": "75000",
  "city": "Paris",
  "area": "Ile-de-France",
  "country": "France",
},
"comm": {
  "1": [{"value": "+33 476705708", "cat": "1"}],
  {"value": "+33 676123962", "cat": "2", "default": "1"}],
  "3": [{"value": "+33 476700772"}],
  "4": [{"value": prenom.nom@company.com}]
}
]
```

1.5.7 API Société(s)

1.5.7.1 Description courte

Nom de l'API : getListCompanyShortDescription.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
session	String : token de session		Abcdefghijklmnop12345
<i>gpsLocalization</i>	Liste		
<i>lat</i>	Décimal : latitude GPS du mobile		1.05
<i>lng</i>	Décimal : longitude GPS du mobile		0.06
<i>localizationRadius</i>	Entier : rayon de localisation (en mètres)		100
<i>items</i>	Liste : ID des sociétés (Couples d'ID pour les sociétés <idMonitoring>,<idTicketing>)	Toutes	["1,2","3,"]
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
id	Couple : ID de la société sur VS, ID de la société sur CRM	
name	String : nom de la société	Coservit
<i>address</i>	Objet	
<i>city</i>	String : ville de la société	Grenoble
<i>distance</i>	Entier : distance séparant le mobile de la société (en mètres)	1000



La distance par rapport au mobile est calculée par le web service : il faut prévoir de s'interfacer avec l'API de géolocalisation de VS.

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "123, ",
    ", 124"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "123, ",
    "name": "Coservit",
    "address": {
      "city": "Grenoble"
    }
  }, {
    "id": ", 124",
    "name": "Dell",
    "address": {
      "city": "Paris"
    }
  }
]
```

1.5.7.2 Description complète

Nom de l'API : `getListCompanyFullDescription.php`

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>gpsLocalization</i>	Liste		
<i>lat</i>	Décimal : latitude GPS du mobile		1.05
<i>lng</i>	Décimal : longitude GPS du mobile		0.06
<i>localizationRadius</i>	Entier : rayon de localisation (en mètres)		100
<i>items</i>	Liste : ID des sociétés (Couples d'ID pour les sociétés <idMonitoring>,<idTicketing>)	Toutes	["1,2","3,"]
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
id	Couple : ID de la société sur VS, ID de la société sur CRM	
name	String : nom de la société	Coservit
criticality	Entier : Criticité de la société. cf §1.2.18)	
<i>address</i>	Objet	
<i>address</i>	String : adresse de la société	13 rue des trembles
<i>postcode</i>	String : code postale de la société	38100
<i>city</i>	String : ville de la société	Grenoble
<i>area</i>	String : région de la société	Rhône-Alpes
<i>country</i>	String : pays de la société	France
<i>gps</i>		
<i>lat</i>	Décimal : latitude GPS de la société	1.05
<i>lng</i>	Décimal : longitude GPS de la société	0.06
<i>comm:<type></i>	Entier : Type de communication (cf. §Erreur ! Source du renvoi introuvable.)	
value	String : donnée correspondant au type (numéro de téléphone, mail...etc)	"+33 476700772"
<i>cat</i>	Entier : catégorie du type de communication (cf. §Erreur ! Source du renvoi introuvable.)	2
<i>default</i>	Booléen : 1 si coordonnées par défaut pour ce type de communication.	1
nbContacts	Entier : nb de contacts de la société	
nbSites	Entier : nb de sites de la société	
<i>logo</i>	Data : logo de la société	
<i>distance</i>	Entier : distance séparant le mobile de la société (en mètres)	1000
updatedDate	Entier : timestamp de mise à jour	1391680932
deletionStatus	Entier : la société est-elle supprimée (voir §1.2.20)	0

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "123, ",
    ", 124"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "123",
    "name": "Coservit",
    "address": {
      "address": "Champs-Élysée",
      "postcode": "75000",
      "city": "Paris",
      "area": "Ile-de-France",
      "country": "France",
    },
    "comm": {
      "1": [{"value": "+33 476705708", "cat": "1"}],
      {"value": "+33 676123962", "cat": "2", "default": "1"}],
      "4": [{"value": prenom.nom@company.com}]
    },
    "nbContacts": "13",
    "nbSites": "2"
  }, {
    "id": "124",
    "name": "Dell",
    "address": {
      "address": "13 rue des trembles",
      "postcode": "38100",
      "city": "Grenoble",
      "area": "Rhône-Alpes",
      "country": "France",
    },
    "comm": {
      "1": [{"value": "+33 476705708", "cat": "1"}],
      {"value": "+33 676123962", "cat": "2", "default": "1"}]
```

```

    ],
    "4": [{"value": prenom.nom@company.com}]
  },
  "nbContacts": "200000",
  "nbSites": "60"
}
]

```

Nom de l'API : getListCompanyMonitoringFullDescription.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type JSON :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des sociétés (Couples d'ID pour les sociétés <idMonitoring>,<idTicketing>)	Toutes	["1,2","3,"]
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter / sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
id	Entier : ID de la société	
name	String : nom de la société	Coservit
criticality	Entier : Criticité de la société. cf §1.2.18)	
<i>address</i>	Objet	
<i>address</i>	String : adresse de la société	13 rue des trembles
<i>postcode</i>	String : code postale de la société	38100
<i>city</i>	String : ville de la société	Grenoble
<i>country</i>	String : pays de la société	France
<i>gps</i>		
<i>lat</i>	Décimal : latitude GPS de la société	1.05
<i>lng</i>	Décimal : longitude GPS de la société	0.06
<i>comm:<type></i>	Entier : Type de communication (cf. §Erreur ! Source du renvoi introuvable.)	
value	String : donnée correspondant au type (numéro de téléphone, mail...etc)	"+33 476700772"
<i>cat</i>	Entier : catégorie du type de communication (cf. §Erreur ! Source du renvoi introuvable.)	2
<i>default</i>	Booléen : 1 si coordonnées par défaut pour ce type de communication.	1
externalId	String : ID de la société CRM liée	
nbContacts	Entier : nb de contacts de la société	
nbSites	Entier : nb de sites de la société	
updatedAt	Entier : timestamp de mise à jour	1391680932
deletionStatus	Entier : la société est-elle supprimée (voir §1.2.20)	0

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "123",
    "124"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "123",
    "name": "Coservit",
    "criticality": 1,
    "externalId": "1195"
    "address": {
      "address": "Champs-Élysée",
      "postcode": "75000",
      "city": "Paris",
      "country": "France",
    },
    "comm": [
      {
        "1": [{"value": "+33 476705708", "cat": "1"}],
        {"value": "+33 676123962", "cat": "2", "default": "1"}
      ],
      "4": [{"value": prenom.nom@company.com}]
    ],
    "nbContacts": "13",
    "nbSites": "2",
    "updatedAt": 1391680990,
    "deletionStatus": 1
  }
]
```

1.5.7.3 Monitoring KPI

Nom de l'API : `getListCompanyMonitoringKPI.php`

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
session	String : token de session		Abcdefghijklmnop12345
<i>gpsLocalization</i>	Liste		
<i>lat</i>	Décimal : latitude GPS du mobile		1.05
<i>lng</i>	Décimal : longitude GPS du mobile		0.06
<i>localizationRadius</i>	Entier : rayon de localisation (en mètres)		100
<i>items</i>	Liste : ID des sociétés (Couples d'ID pour les sociétés <idMonitoring>,<idTicketing>)	Toutes	["1,2","3,"]
<i>levelDescription</i>	Entier : niveau de description (cf. §0)	0	
<i>kpiFilter</i>	Liste : filtre sur les KPI du module (cf. §1.2.9)	Pas de filtre	
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<id>	Liste : couple : ID de la société sur VS, ID de la société sur CRM	
id	Couple : ID de la société sur VS, ID de la société sur CRM	
<i>name</i>	String : nom de la société	
<i>address</i>	Objet	
<i>address</i>	String : adresse de la société	13 rue des trembles
<i>postcode</i>	String : code postale de la société	38100
<i>city</i>	String : ville de la société	Grenoble
<i>area</i>	String : région de la société	Rhône-Alpes
<i>country</i>	String : pays de la société	France
<i>comm:<type></i>	Entier : Type de communication (cf. §Erreur ! Source du renvoi introuvable.)	
<i>value</i>	String : donnée correspondant au type (numéro de téléphone, mail...etc)	"+33 476700772"
<i>cat</i>	Entier : catégorie du type de communication (cf. §Erreur ! Source du renvoi introuvable.)	2
<i>default</i>	Booléen : 1 si coordonnées par défaut pour ce type de communication.	1
<i>administratorId</i>	Entier : ID du contact administrateur	
<i>nbContacts</i>	Entier : nb de contacts de la société	
<i>nbSites</i>	Entier : nb de sites de la société	
<i>logo</i>	Data : logo de la société	
<i>distance</i>	Entier : distance séparant le mobile de la société (en mètres)	1000
monitoringKPI : <kpi>	Liste des KPI (cf. §1.2.9)	
value	String : valeur du KPI	3+
color	String : couleur RGB en hexadécimal	000000
total	String : nb total d'item pour le KPI	10
<i>reportStatus</i>	Entier : statut du KPI (cf. §Erreur ! Source du renvoi introuvable.)	

 Les données facultatives en sortie dépendent du niveau de description demandé (levelDescription)

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "123,",
    ",124"
  ],
  "kpiFilter": [
    "1",
    "2"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
{
  "123," : {
    "id": "123,",
    "monitoringKPI": {
      "1": {
        "value": "3+",
        "color": "FFFFFF",
        "total": "10"
      },
      "2": {
        "value": "3+",
        "color": "FFFFFF",
        "total": "10"
      }
    }
  },
  ",124" {
    "id": ",124",
    "monitoringKPI": {
      "1": {
        "value": "3+",
        "color": "FFFFFF",
        "total": "10"
      },
      "2": {
        "value": "3+",
        "color": "FFFFFF",
        "total": "10"
      }
    }
  }
}
```

1.5.8 API Thèmes

Nom de l'API : getTheme.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>icone</i>	Data : image du bureau encodée en png	
<i>logo</i>	Data : logo de la société	
<i>colors</i>	Liste	
<i>id</i>	Entier : ID de la couleur (cf. §1.2.8)	
<i>value</i>	String : couleur RGB en hexadécimal	000000
<i>reportBI</i>	Objet	
<i>logo</i>	Data : logo à afficher	
<i>companynome</i>	String : Nom de la société	
<i>address</i>	String : Adresse complète de la société	
<i>tel</i>	String : Téléphone de la société	
<i>fax</i>	String : Télécopie de la société	
<i>email</i>	String : Email de destination du rapport	

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

Exemple d'objet JSON en sortie du web service

```
{
  "icone": "XASLKENDFPDLKSDFNFDSPSDOZEFKLNSDF1dfd8ed6v1",
  "logo": "DFIOJSDFOVDNOIDF1LD9KD03KDMVKDE03KD30",
  "colors": [{
    "id": 1,
    "value": "FFFFFF"
  }, {
    "id": 2,
    "value": "FFFFFF"
  }, {
    "id": 3,
    "value": "FFFFFF"
  }, {
    "id": 4,
    "value": "FFFFFF"
  }
]
```

1.5.9 API Statistiques

1.5.9.1 Sur les éléments à synchroniser

Cette API permet la récupération de statistiques sur les données à synchroniser.

Nom de l'API : `getSynchronizationStatistics.php`

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>updatedAt</i>	Entier : Timestamp de mise à jour minimum		1417709667

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i><id du module></i>	String : ID du module (cf. §1.2.5)	1
<i><id de l'entité></i>	Entier : Nombre d'éléments de l'entité à synchroniser	1

Exemple d'objet JSON en entrée du web service

Exemple d'objet JSON en sortie du web service

```
{
  "1": {
    "10": 3,
    "11": 6
  },
  "2": {
    "6": 90350,
    "13": 24691,
    "15": 774,
    "12": 52410,
    "2": 177,
    "1": 10953,
    "10": 9494,
    "16": 55
  }
}
```

1.6 API de Supervision

1.6.1 API Récupérer un historique de métriques

Cette API permet la récupération de l'historique des métriques de supervision pour un service unitaire, service utilisateur, item associé à un service utilisateur ou équipement.

Nom de l'API : `getMetricMonitoringHistory.php`

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>id</i>	Entier : ID de l'item à historier		
<i>type</i>	Entier : type de l'item à historier (cf. §1.6.1.1)		
<i>startDate</i>	Entier : timestamp du début de l'historique		123456789
<i>endDate</i>	Entier : timestamp de fin de l'historique		123456789
<i>format</i>	Entier : format du jeu de données en sortie (cf. 1.6.1.2)		1

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>name</i>	String : Nom de la métrique	
<i>unit</i>	String : unité de mesure	
<i>color</i>	String : Couleur principale (hexadécimal)	
<i>ts</i>	Liste	
<i><date></i>	Entier : timestamp	
<i><value></i>	Décimal : Valeur de la métrique	
<i><color></i>	String : Couleur du point (hexadécimal)	

1.6.1.1 Types d'items historiables

Type d'items à lister	Valeur
Equipement	5
Service utilisateur	9
Service unitaire	8

1.6.1.2 Format du jeu de données en sortie

Format	Valeur
Données brutes	1
amcharts3 with business rules	2

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "id":123,
  "type":3,
  "format":1,"startDate":123456789,
  "endDate":123456789
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "name" : "cpu"
    "unit":"%",
    "color":"#008DDC",
    "ts":[
      [123456789, "50"],
      [123456799, "15", "#FFFFFF"],
      [123456829, "30"]
    ]
  }
]
```

1.6.2 API Société(s)

1.6.2.1 Description complète

Nom de l'API : getListCompanyMonitoringFullDescription.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des sociétés (Couples d'ID pour les sociétés <idMonitoring>,<idTicketing>)	Toutes	["1,2","3,"]
<i>refCompanyld</i>	Entier : ID de societe de reference – racine de sous-arbre de societes pris en compte – 0 = racine est la societe d'utilisateur.	0	84
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbltems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
id	Entier : ID de la société	
name	String : nom de la société	Coservit
criticality	Entier : Criticité de la société. cf §1.2.18)	
<i>address</i>	Objet	
<i>address</i>	String : adresse de la société	13 rue des trembles
<i>postcode</i>	String : code postale de la société	38100
<i>city</i>	String : ville de la société	Grenoble
<i>country</i>	String : pays de la société	France
<i>gps</i>		
<i>lat</i>	Décimal : latitude GPS de la société	1.05
<i>lng</i>	Décimal : longitude GPS de la société	0.06
<i>comm:<type></i>	Entier : Type de communication (cf. §Erreur ! Source du renvoi introuvable.)	
value	String : donnée correspondant au type (numéro de téléphone, mail...etc)	"+33 476700772"
<i>cat</i>	Entier : catégorie du type de communication (cf. §Erreur ! Source du renvoi introuvable.)	2
<i>default</i>	Booléen : 1 si coordonnées par défaut pour ce type de communication.	1
externalId	String : ID de la société CRM liée	
nbContacts	Entier : nb de contacts de la société	
nbSites	Entier : nb de sites de la société	
updatedAt	Entier : timestamp de mise à jour	1391680932
deletionStatus	Entier : la société est-elle supprimée (voir §1.2.20)	0

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "123",
    "124"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "123",
    "name": "Coservit",
    "criticality": 1,
    "externalId": "1195"
    "address": {
      "address": "Champs-Élysée",
      "postcode": "75000",
      "city": "Paris",
      "country": "France",
    },
    "comm": [
      {
        "1": [{"value": "+33 476705708", "cat": "1"}],
        {"value": "+33 676123962", "cat": "2", "default": "1"}
      ],
      "4": [{"value": prenom.nom@company.com}]
    ],
    "nbContacts": "13",
    "nbSites": "2",
    "updatedAt": 1391680990,
    "deletionStatus": 1
  }
]
```

1.6.3 API Equipements(s)

1.6.3.1 Description courte

Nom de l'API : getListHostShortDescription.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des équipements	Tous	
<i>confirmedState</i>	Booléen : 1 si on ne souhaite afficher que les statuts confirmés	0	
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>refCompanyId</i>	Entier : ID de societe de reference – racine de sous-arbre de societes pris en compte – 0 = racine est la societe d'utilisateur.	0	84
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
id	Entier : ID de l'équipement	
name	String : nom de l'équipement	Cosvgre01
token	String : token	xA0rzRYTsUaHWy3pDBJ7Em51Q
iconCat	Entier : code de l'icône de la catégorie de l'équipement (cf. §1.2.6)	2
companyName	String : Nom de la société	Coservit
status	Objet	3
<i>code</i>	Entier : statut de l'équipement (cf. §1.2.14)	
<i>color</i>	String : Code couleur hexadécimal	
<i>date</i>	Entier : timestamp de changement de statut	
<i>duration</i>	String : durée de l'état courant	
<i>ack</i>	Entier : statut d'acquiescement (cf. §1.2.10)	0
<i>forcedCheck</i>	Entier : statut du contrôle immédiat (cf. §1.2.11)	2
<i>lastTicket</i>	Objet : Présent que si le ticket référencé est ouvert sur le CRM	
<i>id</i>	String : Id du ticket	
<i>ref</i>	String : Référence du ticket	
<i>color</i>	String : Code couleur hexadécimal	
underMaintenance	Bool : équipement sous maintenance	

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "123",
    "124"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "1",
    "name": "Equipment 1",
    "token": "aviDhSQTXBz41nW58gE9uFAwp",
    "iconCat": 8,
    "companyName": "Coservit",
    "status":
    {
      "code": 2,
      "color": "F91D05",
      "date": 1390907642,
      "duration": "2 days"
    },
    "ack": 0,
    "lastTicket":
    {
      "ref": "1-62579",
      "color": "F8C706"
    },
    "underMaintenance": true
  }, {
    "id": "2",
    "name": "Equipment 2",
    "token": "xA0rzRYTsUaHWy3pDBJ7Em51Q",
    "iconCat": 6,
    "companyName": "Coservit",
    "status":
    {
      "code": 1,
      "color": "F8C706",
      "date": 1390913674,
      "duration": "2 days"
    },
  },
]
```

```
"ack":1,  
"lastTicket":  
{  
  "ref":"1-21227",  
  "color":"F8C706"  
},  
"underMaintenance":false  
}  
]
```

1.6.3.2 Description complète

Nom de l'API : getListHostFullDescription.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des Equipements		
<i>confirmedState</i>	Booléen : 1 si on ne souhaite afficher que les statuts confirmés	0	
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>refCompanyId</i>	Entier : ID de societe de reference – racine de sous-arbre de societes pris en compte – 0 = racine est la societe d'utilisateur.	0	84
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
id	Entier : ID de l'équipement	
name	String : nom de l'équipement	Cosvgre01
company	Objet	
<i>id</i>	Couple : ID de la société sur VS, ID de la société sur CRM	123,"null"
<i>name</i>	String : Nom de la société	Coservit
iconCat	Entier : code de l'icône de la catégorie de l'équipement (cf. §1.2.6)	
status	Objet	3
<i>code</i>	Entier : statut de l'équipement (cf. §1.2.14)	
<i>color</i>	String : Code couleur hexadécimal	
<i>date</i>	Entier : timestamp de changement de statut	
<i>duration</i>	String : durée de l'état courant	
<i>label</i>	String : Label du statut	CRITICAL
<i>desc</i>	String : libellé du statut	
<i>lastCheck</i>	Entier : timestamp de dernier check	123456789
<i>ack</i>	Objet	
<i>status</i>	Entier : statut d'acquiescement (cf. §1.2.10)	0
<i>date</i>	String : Timestamp de l'acquiescement	
<i>author</i>	String : auteur de l'acquiescement	
<i>forcedCheck</i>	Objet	
<i>status</i>	Entier : statut du contrôle immédiat (cf. §1.2.11)	2
<i>date</i>	String : Timestamp du contrôle immédiat	
<i>author</i>	String : auteur du contrôle immédiat	
<i>lastTicket</i>	Objet : Présent que si le ticket référencé est ouvert sur le CRM	
<i>id</i>	String : Id du ticket	
<i>ref</i>	String : Référence du ticket	
<i>color</i>	String : Code couleur hexadécimal	
underMaintenance	Bool : équipement sous maintenance	

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "123",
    "124"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "23",
    "name": "Cosvgre01",
    "companyId": "123",
    "companyName": "Coservit",
    "iconCat": 2,
    "status":
    {
      "code": 0,
      "color": "55cd30",
      "date": 1390908030,
      "duration": "2 days",
      "desc": "172.16.159: Host unreachable."
    },
    "underMaintenance": false,
  }, {
    "id": "24",
    "name": "Cosvgre04",
    "companyId": "123",
    "companyName": "Coservit",
    "iconCat": 2,
    "status":
    {
      "code": 2,
      "color": "F91D05",
      "date": 1390846033,
      "duration": "2 days",
      "desc": "172.16.159: Host unreachable.",
      "lastCheck": 123456789,
    },
    "underMaintenance": false,
    "lastTicket":
    {
      "ref": "1-22742",
    }
  }
]
```

```
    "color": "F8C706"  
  }  
}  
]
```

1.6.3.3 Monitoring KPI

Nom de l'API : getListHostMonitoringKPI.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
session	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des équipements	Tous	
<i>levelDescription</i>	Entier : niveau de description (cf. §0)	0	
<i>kpiFilter</i>	Liste : filtre sur les KPI du module (cf. §1.2.9)	Pas de filtre	
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>refCompanyId</i>	Entier : ID de société de référence – racine de sous-arbre de sociétés pris en compte – 0 = racine est la société d'utilisateur.	0	84
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<id>	Liste : ID de l'équipement	5
id	Entier : ID de l'équipement	5
<i>name</i>	String : nom de l'équipement	Cosvgre01
<i>companyId</i>	Couple : ID de la société sur VS, ID de la société sur CRM	123,"null"
<i>companyName</i>	String : Nom de la société	Coservit
<i>iconCat</i>	Entier : Code de l'icône de la catégorie de l'équipement (cf. §1.2.6)	
<i>status</i>	Objet	3
<i>code</i>	<i>Entier</i> : statut de l'équipement (cf. §1.2.14)	
<i>color</i>	<i>String</i> : Code couleur hexadécimal	
<i>date</i>	<i>Entier</i> : timestamp de changement de statut	
<i>duration</i>	<i>String</i> : durée de l'état courant	
<i>label</i>	<i>String</i> : Label du statut	CRITICAL
<i>desc</i>	<i>String</i> : libellé du statut	
<i>lastCheck</i>	<i>Entier</i> : timestamp de dernier check	123456789
<i>checkDate</i>	Entier : timestamp de dernier check	123456789
<i>ack</i>	Objet	
status	Entier : statut d'acquiescement (cf. §1.2.10)	0
date	String : Date de l'acquiescement	
author	String : auteur de l'acquiescement	
<i>forcedCheck</i>	Objet	
status	Entier : statut du contrôle immédiat (cf. §1.2.11)	0
date	String : Date de l'acquiescement	
author	String : auteur de l'acquiescement	
<i>lastTicket</i>	Objet : Présent que si le ticket référencé est ouvert sur le CRM	
<i>ref</i>	String : Référence du ticket	
<i>color</i>	String : Code couleur hexadécimal	
monitoringKPI : <kpi>	Liste des KPI (cf. §1.2.9)	
value	String : valeur du KPI	3+
color	String : couleur RGB en hexadécimal	000000
total	String : nb total d'item pour le KPI	10
duration	String : durée de la KPI	
<i>reportStatus</i>	Entier : statut du KPI (cf. §Erreur ! Source du renvoi introuvable.)	



Les données facultatives en sortie dépendent du niveau de description demandé (levelDescription)

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "123",
    "124"
  ],
  "kpiFilter": [
    "1",
    "2"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
{
  "123":{
    "id":"123",
    "monitoringKPI":{
      "1":{
        "value":"3+",
        "color":"FFFFFF",
        "total":"10"
      },
      "2":{
        "value":"3+",
        "color":"FFFFFF",
        "total":"10"
      }
    }
  },
  "124":{
    "id":"124",
    "monitoringKPI":{
      "1":{
        "value":"3+",
        "color":"FFFFFF",
        "total":"10"
      },
      "2":{
        "value":"3+",
        "color":"FFFFFF",
        "total":"10"
      }
    }
  }
}
```

1.6.3.4 Créer un équipement

Nom de l'API : host.php

Méthode HTTP : POST

Cette API permet de créer un équipement

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
session	String : token de session		Abcdefghijklmnop12345
mode	Entier : type de l'équipement. cf 1.2.4		box
name	String : Nom de l'équipement		myHost
address	String : type box : Adresse IP/DNS type agent : Adresse IP publique		192.168.0.1
companyId	Entier : ID de la société de rattachement		1
hostCategoryId	Entier : ID de la catégorie de l'équipement		1
<i>monitoringServerId</i>	Entier : ID du boîtier de supervision		1
<i>hostTemplateId</i>	Entier : ID du modèle d'équipement		
<i>criticite</i>	Entier : ID de la criticité de l'équipement. Cf 1.2.19	1	2
<i>comment</i>	String : Description		description
<i>txDispo</i>	Entier : Taux de disponibilité cible (pourcentage)		95
<i>timePeriodDispold</i>	Entier : ID de la plage horaire de disponibilité		1
<i>timePeriodControllId</i>	Entier : ID de la plage horaire de contrôle		1
<i>serviceTemplateId</i>	Entier : ID du modèle de contrôle de l'équipement		1
<i>checkInterval</i>	Entier : Intervalle normal de contrôle		123456789
<i>agentCheckInterval</i>	Entier : type agent : Intervalle d'exécution des points de contrôle de l'équipement en minutes		5
<i>agentMaxExecTime</i>	Entier : type agent : Délai maximum d'exécution des scripts en secondes		60



Les attributs autres que "session" sont transmis en POST avec les informations du formulaire dans le corps du message

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>id</i>	Entier : id de l'équipement créé	

Voici des exemples complets d'un objet JSON en sortie :

Exemple d'objet JSON en sortie du web service
<pre>{ "id": "23" }</pre>

1.6.3.5 Modifier un équipement

Nom de l'API : host.php

Méthode HTTP : PUT

Cette API permet de modifier un équipement

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>id</i>	Entier id de l'équipement à modifier		4
<i>name</i>	String : Nom de l'équipement		myHost
<i>address</i>	String : Adresse IP/DNS		192.168.0.1
<i>hostCategoryId</i>	Entier : ID de la catégorie de l'équipement		1
<i>hostTemplateId</i>	Entier : ID du modèle d'équipement		
<i>criticite</i>	Entier : ID de la criticité de l'équipement. Cf 1.2.19	1	2
<i>comment</i>	String : Description		description
<i>txDispo</i>	Entier : Taux de disponibilité cible (pourcentage)		95
<i>timePeriodDispold</i>	Entier : ID de la plage horaire de disponibilité		1
<i>timePeriodControlld</i>	Entier : ID de la plage horaire de controle		1
<i>serviceTemplateId</i>	Entier : ID du modèle de contrôle de l'équipement		1
<i>checkInterval</i>	Entier : Intervalle normal de controle		123456789
<i>agentCheckInterval</i>	Entier : type agent : Intervalle d'exécution des points de contrôle de l'équipement en minutes		5
<i>agentMaxExecTime</i>	Entier : type agent : Délai maximum d'exécution des scripts en secondes		60



Les attributs autres que "session" sont transmis en POST avec les informations du formulaire dans le corps du message

1.6.3.6 Supprimer un équipement

Nom de l'API : host.php

Méthode HTTP : DELETE

Cette API permet de supprimer un équipement

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>id</i>	Entier id de l'équipement à supprimer		4

1.6.4 API Contact(s)

1.6.4.1 Description complète

Nom de l'API : `getListContactMonitoringFullDescription.php`

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
session	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des contacts (Couples d'ID pour les contacts <idMonitoring>,<idTicketing>)	Tous	["1,2","3,"]
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter sorting</i>	Liste : Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
id	Couple : ID du contact sur VS, ID du contact sur CRM	"10,11"
lastName	String : nom du contact	Bruno Richoud
updatedAt	Entier : timestamp de mise à jour	1391680932
deletionStatus	Entier : la société est-elle supprimée (voir §1.2.20)	0
company	Objet	
id	Entier : ID de la société VS	123,"null"
<i>comm:<type></i>	Entier : Type de communication (cf. §Erreur ! Source du renvoi introuvable.)	
value	String : donnée correspondant au type (numéro de téléphone, mail...etc)	"+33 476700772"
<i>cat</i>	Entier : catégorie du type de communication (cf. §Erreur ! Source du renvoi introuvable.)	2
<i>default</i>	Booléen : 1 si coordonnées par défaut pour ce type de communication.	1

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    123,
    124
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id":123,
    "lastName":"Bruno Richoux",
    "company" :
    {
      id":22
    }
    "comm":
    {
      "1":[{"value":"+33 476705708","cat":"1"},
        {"value":"+33 676123962","cat":"2","default":"1"}
      ],
      "3":[{"value":"+33 476700772"}],
      "4":[{"value":prenom.nom@company.com}]
    }
  }
]
```

1.6.5 API Service(s) Unitaire(s)

1.6.5.1 Description courte

Nom de l'API : getListServiceShortDescription.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des services unitaires	Tous	
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit", "dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>id</i>	Entier : ID du service unitaire	
<i>name</i>	String : nom du service unitaire	MS Win – Mémoire vive
<i>token</i>	String : token	xA0rzRYTsUaHWy3pDBJ7Em51Q
<i>status</i>	Objet	3
<i>code</i>	Entier : statut du service unitaire (cf. §1.2.14)	
<i>color</i>	String : Code couleur hexadécimal	
<i>date</i>	Entier : timestamp de changement de statut	
<i>duration</i>	String : durée de l'état courant	
<i>cat</i>	String : catégorie du service	RAM
<i>ack</i>	Entier : statut d'acquiescement (cf. §1.2.10)	0
<i>forcedCheck</i>	Entier : statut du contrôle immédiat (cf. §1.2.11)	0
<i>lastTicket</i>	Objet : Présent que si le ticket référencé est ouvert sur le CRM	
<i>id</i>	String : Id du ticket	
<i>ref</i>	String : Référence du ticket	
<i>color</i>	String : Code couleur hexadécimal	
<i>host</i>	Objet : équipement lié au service unitaire	
<i>name</i>	String : nom de l'équipement	

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "1",
    "2"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "1",
    "name": "Service unitaire 1",
    "status":
    {
      "code": 2,
      "color": "F91D05",
      "date": 1391030739,
      "duration": "2 days"
    },
    "cat": "RAM",
    "ack": 1,
    "lastTicket":
    {
      "ref": "1-65545",
      "color": "F8C706"
    }
  },
  {
    "id": "2",
    "name": "Service unitaire 2",
    "status":
    {
      "code": 0,
      "color": "55cd30",
      "date": 1391020258,
      "duration": "2 days"
    },
    "cat": "RAM",
    "ack": 0,
    "lastTicket":
    {
      "ref": "1-98577",
```

```
        "color": "F8C706"  
    }  
}  
]
```

1.6.5.2 Description complète

Nom de l'API : `getListServiceFullDescription.php`

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des services unitaires	Tous	
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit", "dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
id	Entier : ID du service unitaire	
name	String : nom du service unitaire	MS Win – Mémoire vive
token	String : token	xA0rzRYTsUaHWy3pDBJ7Em51Q
<i>cat</i>	String : catégorie du service unitaire	RAM
checkInterval	Entier : Interval normal de contrôle (en minutes)	5
status	Objet	3
code	Entier : statut du service unitaire (cf. §1.2.14)	
color	String : Code couleur hexadécimal	
<i>date</i>	Entier : timestamp de changement de statut	
<i>duration</i>	String : durée de l'état courant	
label	String : Label du statut	CRITICAL
<i>desc</i>	String : libellé du statut	
<i>lastCheck</i>	Entier : timestamp de dernière maj	123456789
company	Objet	
id	Couple : ID de la société sur VS, ID de la société sur CRM	"123,"
name	String : Nom de la société	Coservit
<i>ack</i>	Objet	
status	Entier : statut d'acquiescement (cf. §1.2.10)	0
date	Entier : Timestamp de de l'acquiescement	
author	String : auteur de l'acquiescement	
<i>forcedCheck</i>	Objet	
status	Entier : statut du contrôle immédiat (cf. §1.2.11)	0
date	String : Timestamp du contrôle immédiat	
author	String : auteur du contrôle immédiat	
<i>lastTicket</i>	Objet : Présent que si le ticket référencé est ouvert sur le CRM	
<i>id</i>	String : Id du ticket	
<i>ref</i>	String : Référence du ticket	
<i>color</i>	String : Code couleur hexadécimal	
host		
id	Entier : ID de l'équipement associé	
name	String : nom de l'équipement associé	

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "1",
    "2"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "1",
    "name": "Service unitaire 1",
    "cat": "RAM",
    "status":
    {
      "code": 3,
      "color": "C7C8CA",
      "date": 1392636731,
      "desc": "Out of Memory 87% > 80%",
      "duration": "1h 3m 32s"
    },
    "checkDate": 1392729001,
    "ack":
    {
      "ack": 0,
      "date": 1392643059,
      "author": "Jimmy KLEIN"
    },
    "host":
    {
      "id": 1,
      "name": "Copcgre23"
    },
    "lastTicket":
    {
      "ref": "1-19329",
      "color": "F8C706"
    }
  }
]
```

1.6.6 API Service(s) Utilisateur

1.6.6.1 Description courte

Nom de l'API : getListServiceUserShortDescription.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des services utilisateur	Tous	
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit", "dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>id</i>	Entier : ID du service utilisateur	
<i>name</i>	String : nom du service utilisateur	Messagerie
<i>shared</i>	Boolean : service utilisateur de type cloud	
<i>companyName</i>	String : nom de la société	
<i>ack</i>	Entier : statut d'acquiescement (cf. §1.2.10)	0
<i>status</i>	Objet	3
<i>code</i>	Entier : statut du service utilisateur (cf. §1.2.14)	
<i>color</i>	String : Code couleur hexadecimal	
<i>date</i>	Entier : timestamp de changement de statut	
<i>duration</i>	String : durée de l'état courant	
<i>comment</i>	String : commentaire indiqué comme « Afficher » pour le service utilisateur	

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={  
  "items": [  
    "1",  
    "2"  
  ]  
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "1",
    "name": "Service Utilisateur 1",
    "shared": 0,
    "companyName": "Company 1",
    "ack": 0,
    "status":
    {
      "code": 4,
      "color": null,
      "date": 1390856993,
      "duration": "2 days"
    }
  },
  {
    "id": "2",
    "name": "Service Utilisateur 2",
    "shared": 1,
    "companyName": "Company 2",
    "ack": 0,
    "status":
    {
      "code": 1,
      "color": null,
      "date": 1390847819,
      "duration": "2 days"
    }
  },
]
```

1.6.6.2 Description complète

Nom de l'API : `getListServiceUserFullDescription.php`

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des services utilisateur	Tous	
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>includeServiceUserNotActivated</i>	Boolean : Service utilisateur activé ?	false	true
<i>filter <u>sorting</u></i>	Liste : Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit","dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>id</i>	Entier : ID du service utilisateur	
<i>name</i>	String : nom du service utilisateur	
<i>shared</i>	Boolean : service utilisateur de type cloud	
<i>company</i>	Objet	
<i>id</i>	Couple : ID de la société sur VS, ID de la société sur CRM	
<i>name</i>	String : nom de la société	
<i>ack</i>	Objet	
<i>status</i>	Entier : statut d'acquittement (cf. §1.2.10)	0
<i>status</i>	Objet	3
<i>code</i>	Entier : statut du service utilisateur(cf. §1.2.14)	
<i>color</i>	String : Code couleur hexadecimal	
<i>date</i>	Entier : timestamp de changement de statut	
<i>duration</i>	String : durée de l'état courant	
<i>lastCheck</i>	Entier : timestamp de la dernière maj	123456789

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "12",
    "2"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "1",
    "name": "Service Utilisateur 1",
    "shared": 1,
    "companyId": "1",
    "companyName": "Company 69",
    "status": {
      "code": 4,
      "color": null,
      "date": 1390891827,
      "duration": "2 days"
    },
    "checkDate": 1390854571,
    "ack": 0
  },
  {
    "id": "2",
    "name": "Service Utilisateur 2",
    "shared": 0,
    "companyId": "2",
    "companyName": "Company 37",
    "status": {
      "code": 0,
      "color": null,
      "date": 1390889458,
      "duration": "2 days"
    },
    "checkDate": 1390911561,
    "ack": 0
  }
]
```

1.6.6.3 Monitoring KPI

Nom de l'API : `getListServiceUserMonitoringKPI.php`

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
session	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des services utilisateur	Tous	
<i>levelDescription</i>	Entier : niveau de description (cf. §0)	0	
<i>kpiFilter</i>	Liste : filtre sur les KPI du module (cf. §1.2.9)	Pas de filtre	
<i>includeServiceUserNotActivated</i>	Boolean : Service utilisateur activé ?	false	true
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter / sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit", "dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<id>	Liste : ID du service utilisateur	
id	Entier : ID du service utilisateur	
<i>name</i>	String : nom du service utilisateur	
<i>shared</i>	Boolean : service utilisateur de type cloud	
<i>checkDate</i>	Entier : timestamp de la dernière maj	123456789
<i>company</i>	Objet	
<i>id</i>	Couple : ID de la société sur VS, ID de la société sur CRM	
<i>name</i>	String : nom de la société	
<i>ack</i>	Objet	
status	Entier : statut d'acquittement (cf. §1.2.10)	0
date	String : Date de l'acquittement	
author	String : auteur de l'acquittement	
<i>status</i>	Objet	3
<i>code</i>	Entier : statut de l'équipement (cf. §1.2.14)	
<i>color</i>	String : Code couleur hexadécimal	
<i>date</i>	Entier : timestamp de changement de statut	
<i>desc</i>	String : libellé du statut	
monitoringKPI : <kpi>	Liste des KPI (cf. §1.2.9)	
value	String : valeur du KPI	3+
color	String : couleur RGB en hexadécimal	000000
total	String : nb total d'item pour le KPI	10
duration	String : durée de la KPI	
<i>reportStatus</i>	Entier : statut du KPI (cf. §Erreur ! Source du renvoi introuvable.)	

 Les données facultatives en sortie dépendent du niveau de description demandé (levelDescription)

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "123",
    "124"
  ],
  "kpiFilter": [
    "1",
    "2"
  ]
}
```

```
}
```

Exemple d'objet JSON en sortie du web service

```
{  
  "123":{  
    "id":"123",  
    "monitoringKPI":{  
      "1":{  
        "value":"3+",  
        "color":"FFFFFF",  
        "total":"10"  
      },  
      "2":{  
        "value":"3+",  
        "color":"FFFFFF",  
        "total":"10"  
      }  
    }  
  },  
  "124":{  
    "id":"124",  
    "monitoringKPI":{  
      "1":{  
        "value":"3+",  
        "color":"FFFFFF",  
        "total":"10"  
      },  
      "2":{  
        "value":"3+",  
        "color":"FFFFFF",  
        "total":"10"  
      }  
    }  
  }  
}
```


1.6.7 API Relations Service(s) Utilisateur

1.6.7.1 Description courte

Nom de l'API : getListServiceUserRelationShortDescription.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>item</i>	Entier : ID du service utilisateur		
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nblItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>includeServiceUserNotActivated</i>	Boolean : Service utilisateur activé ?	false	true
<i>filter sorting</i>	Liste . Le filtre ou le tri est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit", "dell"]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
id	Entier : ID de l'item en relation	
type	Entier : type d'item (cf. §1.2.2)	
name	String : nom de l'item	Messagerie
link	Entier : type de relation (cf §1.2.17)	
<i>shared</i>	Boolean : service utilisateur de type cloud	
<i>company</i>	Objet	
id	Couple : ID de la société sur VS, ID de la société sur CRM	
<i>name</i>	String : nom de la société	
<i>host</i>	Objet : objet présent pour les items de type Services unitaires	
<i>name</i>	String : Nom de l'équipement lié	
ack	Booléen : statut d'acquittement	0
status	Objet	3
code	Entier : statut de l'équipement (cf. §1.2.14)	
color	String : Code couleur hexadécimal	
date	Entier : timestamp de changement de statut	
duration	String : durée de l'état courant	

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "item": "123"
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "3",
    "type": 5,
    "name": "Copcgre01",
    "link": 2,
    "company":
    {
      "id": "1,",
      "name": "Coservit"
    },
    "ack": 0,
    "status":
    {
      "code": 0,
      "color": "55cd30",
      "date": null,
      "desc": "172.16.159: Host unreachable.",
      "duration": "44 années 1 mois 4 semaines 14h 19m 27s"
    }
  },
  {
    "id": "3",
    "type": 9,
    "name": "Messagerie",
    "link": 1,
    "company":
    {
      "id": "1,",
      "name": "Coservit"
    },
    "ack": 0,
    "status":
    {
```

```
        "code": 0,  
        "color": "55cd30",  
        "date": 1392690954,  
        "duration": "44 années 1 mois 4 semaines 14h 19m 27s"  
    }  
}
```

1.6.8 API Acquittements

1.6.8.1 Créer un acquittement

Nom de l'API : `postAcknowledgement.php`

Cette API permet l'acquiescement d'une alerte sur un service unitaire ou sur un équipement.

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>id</i>	Entier : ID de l'item		101
<i>type</i>	Entier : type de l'item (cf. §1.2.2)		4
<i>comment</i>	String : commentaire associé à l'acquiescement		



Les attributs « id, type, comment » sont transmis en POST avec les informations du formulaire dans le corps du message

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
---	------	---------

Voici des exemples complets d'un objet JSON en sortie :

```
Exemple d'objet JSON en sortie du web service  
  
[  
  {  
    "id": "23",  
    "status": "0"  
  }, {  
    "id": "24",  
    "status": "0"  
  }  
]
```

1.6.9 API Contrôle

1.6.9.1 Planifier un contrôle

Nom de l'API : postCheck.php

Cette API permet de lancer un contrôle immédiat sur un service unitaire ou sur un équipement.

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>type</i>	Entier : type de l'item (cf. §1.2.2)		5
<i>id</i>	Entier : ID de l'item		101
<i>date</i>	Entier : timestamp de planification	Timestamp actuelle (contrôle immédiat)	



Les attributs « type, id et date » sont transmis en POST avec les informations du formulaire dans le corps du message

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
---	------	---------

Voici des exemples complets d'un objet JSON en sortie :

Exemple d'objet JSON en sortie du web service
<pre>{ "id": "23", "status": "2" }</pre>

1.6.10 API Maintenance

1.6.10.1 Supprimer une ou plusieurs maintenances

Nom de l'API : maintenance.php

Méthode HTTP : DELETE

Cette API permet de supprimer un équipement

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des maintenances	Aucune	

1.6.10.2 Ajouter une maintenance

Nom de l'API : maintenance.php

Méthode HTTP : POST

Cette API permet d'ajouter une maintenance.

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>dateStart</i>	Int : timestamp		
<i>dateStop</i>	Int : timestamp		
<i>requesterType</i>	String : 01 = Partenaire, 02 = Client		
<i>typeImpact</i>	Int : 0 = Aucune, 1 = Mask impact, 2 = Force le status critique		
<i>hostListId</i>	Liste : ID des équipements pour lesquelles ajouter la maintenance.		
<i>serviceUnitListId</i>	Liste : ID des services pour lesquelles ajouter la maintenance.		
<i>contributorListId</i>	Liste : ID des intervenants de la maintenance.		
<i>description</i>	String : la description		

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>id</i>	Entier : id de la maintenance créée	

1.6.10.3 Liste des maintenances

Nom de l'API : `getListMaintenanceFullDescription.php`

Méthode HTTP : GET

Cette API permet de récupérer les maintenances

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	-1 (tous)	10
<i>filter</i>	Liste . Le filtre est réalisé sur chaque attribut avec l'opérateur « ET »)		
<i>entity</i>	Entier : Type d'entité sur laquelle sera appliqué le filtre ou le tri (cf. §1.2.2)		10
<i>attribut</i>	Entier : attribut de l'entité sur lequel sera appliqué le filtre ou le tri (cf. §1.2.3)		2
<i>values</i>	Liste : valeurs des attributs. Le filtre ou le tri est réalisé sur chaque valeur avec l'opérateur « OU »).		["coservit", "dell"]

Entités possibles : équipements, services ou maintenances. Les ID de ces entités doivent être passés dans la liste de valeur.

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>id</i>	Entier : id de la maintenance	
<i>impactedCompanyListId</i>	Liste : Liste des id des sociétés impactées par la maintenance	
<i>hostListId</i>	Liste : Liste des équipements dans la maintenance	
<i>serviceUnitListId</i>	Liste : Liste des services dans la maintenance	
<i>impactedServiceUnitListId</i>	Liste : Liste des services impactés par la maintenance	
<i>contributorListId</i>	Liste : Liste des intervenants (utilisateurs) de la maintenance	
<i>contactListId</i>	Liste : Liste des contacts de la maintenance	
<i>participantListId</i>	Liste : Liste des intervenants (contacts) de la maintenance	
<i>description</i>	String : Description de la maintenance	
<i>dateStart</i>	Entier : Timestamp de début de la maintenance	
<i>dateStop</i>	Entier : Timestamp de début de la maintenance	
<i>typeImpact</i>	Entier : 0 = Aucune, 1 = Mask impact, 2 = Force le status critique	
<i>requesterType</i>	String : 01 = Partenaire, 02 = Client	
<i>Status</i>	String : Status de la maintenance (Planifiée, En cours, Terminée, Annulée)	

1.6.10.4 Liste des intervenants

Nom de l'API : `getListUserMaintenanceShortDescription.php`

Méthode HTTP : GET

Cette API permet de récupérer les intervenants disponibles pour les maintenances.

Les tableaux suivants définissent l'interface webservice qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>firstItem</i>	Entier : premier item de la liste à récupérer	0	5
<i>nbItems</i>	Entier : nombre max d'items à récupérer	Tous	10
<i>refCompanyId</i>	Entier : ID de société de référence – racine de sous-arbre des sociétés pris en compte – 0 = racine de la société d'utilisateur	0	2

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
id	Entier : ID de l'intervenant	1
lastName	String : Nom de l'intervenant	Doe
firstName	String : Prénom de l'intervenant	John
company	Objet	
id	Entier : Id de la société	2
name	String : Nom de la société	Coservit

1.6.11 API Reporting

1.6.11.1 Description courte

Nom de l'API : `getLisReportingShortDescription.php`

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des rapports		[10,11]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
id	Entier : ID du rapport	

name	String : nom du rapport	Rapport 1
recipient	Entier : type de destinataire - constante	2
templateId	Entier : id du modèle	13

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "10",
    "11"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "10",
    "name": "Rapport1",
    "recipient": 2,
    "templateId": 13,
  }, {
    "id": "11",
    "name": "Rapport2",
    "recipient": 2,
    "templateId": 13,
  }
]
```

1.6.11.2 Description complète

Nom de l'API : getLisReportingFullDescription.php

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>items</i>	Liste : ID des rapports		[10,11]

Attributs de l'objet en sortie (en italique les données facultatives)	Type	Exemple
<i>id</i>	Entier : ID du rapport	
<i>name</i>	String : nom du rapport	Rapport 1
<i>recipient</i>	Entier : type de destinataire - constante	2
<i>templateId</i>	Entier : id du modèle	13
<i>extraData</i>	Objet : objet json (depends utilisation) – peut-être null	

Voici des exemples complets d'un objet JSON en entrée et en sortie :

Exemple d'objet JSON en entrée du web service

```
param={
  "items": [
    "10",
    "11"
  ]
}
```

Exemple d'objet JSON en sortie du web service

```
[
  {
    "id": "10",
    "name": "Rapport1",
    "recipient": 2,
    "templateId": 13,
    "extraData": {
      "map": null,
      "metadata": {
        "width": 1920,
        "height": 1080
      }
    }
  }, {
    "id": "11",
    "name": "Rapport2",
    "recipient": 2,
    "templateId": 13,
    "extraData": {
      "map": null,
      "metadata": {
        "width": 1920,
        "height": 1080
      }
    }
  }
]
```

1.6.11.3 Modifier un rapport

Nom de l'API : reporting.php

Méthode http : PUT

Les tableaux suivants définissent l'interface du web service qui consomme et produit un objet de type **JSON** :

Attributs de l'objet en entrée (en italique les données facultatives)	Type	Valeur par défaut	Exemple
<i>session</i>	String : token de session		Abcdefghijklmnop12345
<i>id</i>	Entier : id du rapport à modifier		4
<i>extraData</i>	String : objet extra_data en format JSON – valeur null possible		

Les attributs autres que "session" sont transmis en PUT avec les informations du formulaire dans le corps du message