

# Guide utilisateur

## Web service REST presenceRegistration

Check In and Out at Work

---

## Contenu

Enregistrement des présences dans le secteur du nettoyage.....	4
Service SOAP et service REST.....	5
Préparation.....	6
S'identifier comme employeur.....	6
Demander ou créer un certificat.....	6
Créer un compte web service dans Chaman.....	6
Demander un token d'accès OAuth.....	7
Rechercher des fichiers YAML.....	7
Principe de fonctionnement.....	8
Asynchronisme.....	8
Déroulement classique.....	8
Création.....	8
Consultation.....	9
Délais de traitement.....	9
Statut.....	10
Que signifie exactement le statut 'Failed' ?.....	10
Batch.....	10
Documentation technique.....	12
RegisterInBulk.....	12
Description.....	12
Requête.....	12
Body.....	12
Réponse.....	14
Exemple.....	15
Requête.....	15
Réponse.....	17
Explications.....	19
Read by id.....	19

Description .....	19
Requête .....	19
Réponse.....	19
Exemple.....	22
Requête.....	22
Réponse.....	22
Search .....	24
Description .....	24
Requête .....	24
Body .....	24
Autres paramètres .....	26
Réponse.....	26
Exemple.....	27
Requête .....	27
Réponse.....	27
Explications.....	28
Environnements et URL.....	29

# Enregistrement des présences dans le secteur du nettoyage

Conformément à la loi programme du 26/12/2022, les travailleurs effectuant des activités spécifiques de nettoyage doivent obligatoirement enregistrer le début (IN) et la fin (OUT) de leurs prestations, ainsi que les périodes de repos.

Un enregistrement de présence représente l'entrée ou la sortie d'un employé sur son lieu de travail. Il existe donc 2 types de présence :

- **IN** : l'entrée du travailleur (mais également la fin d'une pause)
- **OUT** : la sortie du travailleur (mais également le début d'une pause)

Un enregistrement de présence typique regroupe les informations suivantes :

- **QUI** ? Qui est entré ou sorti du bâtiment ? Cette information est représentée par le numéro d'identification à la sécurité sociale (NISS) du travailleur.
- **OÙ** ? Sur quel lieu de travail le travailleur est-il entré ou d'où est-il sorti ? Cette information est représentée soit par les coordonnées de géolocalisation du travailleur au moment du 'check in' (entrée) ou du 'check out' (sortie), ou d'une description de l'adresse au format texte.
- **QUAND** ? À quelle heure le travailleur est-il entré ou sorti ? Cette information est représentée par une date et une heure, auxquelles le travailleur a effectué son check in ou son check out.

**Check In and Out at Work** (parfois abrégé 'ClaO') est le service en ligne de suivi des prestations des travailleurs des différents secteurs concernés. Elle permet la création et la consultation de 'presence registrations' (enregistrements de présences).

Deux méthodes permettent l'enregistrement des présences :

- enregistrement à l'aide d'un smartphone, et
- appel à un web service REST (couplé à votre éventuel système de badge existant), appelé aussi **PresenceRegistrations**.

Ce guide utilisateur décrit les aspects fonctionnels du web service REST.

# Service SOAP et service REST

Le portail de la sécurité sociale propose deux services avec le nom **PresenceRegistrations** :

- l'un utilise la technologie SOAP, et
- l'autre utilise la technologie REST.

Il s'agit de deux web services différents, destinés à des secteurs particuliers :

- Les secteurs de la **construction**, de la **viande** et du **gardiennage** doivent utiliser le service **SOAP**. L'enregistrement de présence se fait une fois par jour, par personne et par déclaration de travail.
- Le secteur du **nettoyage** doit utiliser le service **REST**. L'enregistrement de présence se fait à chaque entrée et sortie du travailleur, et également à chaque pause pour chaque lieu de travail.

D'autres secteurs seront ajoutés au cours de l'année 2024 et 2025. Ils devront également utiliser le service REST.

Comme mentionné ci-dessus, ce guide utilisateur décrit uniquement le service REST.

Pour le service SOAP, rendez-vous sur la page ['Comment enregistrer les présences ?' de Checkinetwork sur le portail de la sécurité sociale](#).

# Préparation

Avant d'accéder au service REST **PresenceRegistrations**, vous devez effectuer certaines démarches préalables. Si vous avez déjà utilisé un web service de la sécurité sociale, il est possible que certaines de ces étapes aient déjà été effectuées.

## S'identifier comme employeur

Pour utiliser les services en ligne de la sécurité sociale, l'entreprise doit être reconnue en tant qu'employeur sur le portail de la sécurité sociale. Le portail regroupe l'ensemble des services en ligne de la sécurité sociale.

Afin de protéger les données des employeurs et des travailleurs, vous devez, en tant qu'utilisateur des services en ligne :

- être identifié en tant qu'employeur, et
- être enregistré sur le portail.

Pour ce faire, suivez les quatre étapes décrites sur la page [Identifiez-vous en tant qu'employeur sur le portail de la sécurité sociale](#).

## Demander ou créer un certificat

Pour invoquer un service REST de la sécurité sociale, vous devez utiliser un certificat.

Il n'est plus obligatoire d'utiliser un certificat GlobalSign pour accéder aux services REST de la sécurité sociale. Vous pouvez utiliser un certificat auto-signé (self-signed).

## Créer un compte web service dans Chaman

**Chaman** ou Channel Management est le service en ligne qui permet de gérer les canaux techniques (FTP, SFTP, SOAP, et REST) de la sécurité sociale.

Un [manuel est disponible sur le portail de la sécurité sociale](#).

Dans le service en ligne Chaman, vous devez créer un compte web service de type REST, sélectionner la permission 'Check in And Out at Work' et y enregistrer votre certificat.

**Ajouter un compte Webservice**

Type\*  
REST

Nom du compte\*  
My CiaO account

Permissions sécurisées

<input type="checkbox"/> CareerPro - Federal Learning Account - Déclaration	<input checked="" type="checkbox"/> Check In And Out @ Work
<input type="checkbox"/> Consultation de la déclaration Dimona	<input type="checkbox"/> Effectuer des déclarations Dimona
<input type="checkbox"/> WITA Amateur - Entreprise	

Gestion du certificat

Certificat\*

Nom du certificat\*

**Valider** Annuler

Utilisez les liens suivants :

- [Environnement d'acceptation Chaman](#)
- [Environnement de production Chaman](#)

## Demander un token d'accès OAuth

OAuth permet aux applications tierces d'obtenir un accès limité à un service. Avant de pouvoir utiliser le service REST PresenceRegistration, vous devez demander un token d'accès au service OAuth.

Vous trouverez la [documentation OAuth sur le portail de la sécurité sociale](#).

## Rechercher des fichiers YAML

YAML est un format standard de représentation de données. Le fichier YAML de PresenceRegistration décrit les méthodes et les zones. Vous le trouverez sur le portail de la sécurité sociale.

Actuellement, le fichier YAML ne se trouve pas sur le portail. Veuillez le demander à notre équipe support : envoyez un e-mail à [ciaogatb@smals.be](mailto:ciaogatb@smals.be).

# Principe de fonctionnement

L'ONSS met à disposition des employeurs un web service, leur permettant :

- d'enregistrer les présences de leurs travailleurs dans ClaO.
- de consulter les présences de leur travailleurs, et de consulter les éventuelles remarques.

## Asynchronisme

Il convient de souligner que l'ONSS ne peut pas assurer le traitement synchrone des remarques liées aux enregistrements de présences. En effet, le traitement des présences implique des appels à d'autres web services, rendant dès lors le traitement des remarques asynchrones.

Par conséquent, les web services ClaO décrivent deux opérations distinctes :

- la création de présences, et
- la consultation de présences.

## Déroulement classique

### Création

- L'utilisateur doit utiliser l'endpoint de création afin de soumettre ses présences.  
([/presenceRegistrations/registerInBulk](#))
- Il reçoit ensuite l'un des 2 codes suivants :
  - **200** : dans ce cas, la réponse comporte autant d'objets que d'enregistrements de présence créés par l'utilisateur. Pour chaque enregistrement, un objet associé présente deux propriétés mutuellement exclusives :
    - **createdPresenceRegistration** : cette propriété décrit l'enregistrement de présence qui a été créé. Il est à noter que la propriété 'remarks' est vide. En effet, comme mentionné précédemment, le traitement des remarques est asynchrone. Par conséquent, lors de la création de l'enregistrement de présence dans ClaO, les remarques ne sont pas encore calculées. Si cette propriété est nulle, cela indique qu'une erreur s'est produite lors de l'enregistrement de la présence. Cette erreur est décrite dans la propriété '[notCreatedPresenceRegistration](#)'.
    - **notCreatedPresenceRegistration** : si cette propriété n'est pas nulle, cela indique qu'une erreur s'est produite lors de l'enregistrement de la présence. Le problème est décrit dans les propriétés de cet objet :
      - '[presenceRegistrationSubmitted](#)' : comme son nom l'indique, cette propriété reprend l'enregistrement de présence telle qu'il a été soumis par



l'utilisateur. Cette présence contient donc une (ou des) erreur(s), décrite(s) dans la propriété '**errorList**'

- '**errorList**' : contient une liste d'erreurs décrivant les raisons pour lesquelles l'enregistrement n'a pas pu être créé. Il convient dès lors de corriger les erreurs mentionnées, et de soumettre à nouveau **uniquement les présences en erreur**.
- **500** : une erreur non gérée a eu lieu, les enregistrements de présences n'ont pas été créés.

## Consultation

L'utilisateur dispose de deux méthodes afin de consulter les enregistrements de présences :

- Sur la base d'un **ID**, afin d'obtenir les informations d'une présence en particulier ([/presenceRegistrations/{id}](#))
- Sur la base de **critères de recherche**, afin de lister la ou les présences qui correspondent aux critères mentionnés ([/presenceRegistration/search](#))

Un utilisateur utilisant les web services sera toujours associé à l'employeur propriétaire du certificat. Dès lors, la consultation des enregistrements de présences sera restreinte aux enregistrements pour ce dit employeur, ou sa chaîne de sous-traitance.

Comme mentionné précédemment, lors de la création des présences, le champ **remarks** sera vide. Il est donc nécessaire d'attendre que ClaO traite les remarques avant d'essayer de les consulter.

## Délais de traitement

Afin d'éviter de surcharger inutilement charger le web service de PresenceRegistration, il est essentiel de prendre en considération les informations suivantes :

- Chaque enregistrement de présence possède une propriété **status**. Cette propriété permet de suivre le statut du traitement des remarques. Les différentes valeurs suivantes sont possibles :
  - **REGISTERED** : la présence a été créée, mais elle n'a pas encore été traitée, ce qui signifie que la liste des remarques est vide mais susceptible d'évoluer.
  - **VALIDATED** : la présence a été traitée, et aucune remarque n'a été générée. La liste des remarques est donc vide, et les remarques ne seront plus calculées.
  - **CANCELLED** : la présence a été annulée, et par conséquent, ses remarques ne seront plus calculées.
  - **FAILED** : la présence a été traitée et des remarques ont été relevées. Cette liste de remarques est susceptible d'évoluer dans le futur (voir point 'batch'). Attention : l'enregistrement de présence est tout de même enregistré. Vous devez dans ce cas examiner les remarques et examiner si il est possible de les corriger.

## Statut

Afin d'effectuer des vérifications sur le statut de la présence, il faut regarder à la propriété **code** de l'objet **status**. La **date** permet quant à elle de savoir quand le statut a été changé pour la dernière fois :

```
...  
"status": {  
  "code": "registered",  
  "date": "2024-02-26T15:29:22.312422275+01:00"  
},
```

## Que signifie exactement le statut 'Failed' ?

Le statut 'Failed' signifie que l'enregistrement de présence a correctement été reçu par l'ONSS, et qu'il a bien été enregistré dans notre base de données.

Cependant, l'enregistrement a été contrôlé et des remarques ont été constatées qui doivent attirer votre attention. Les remarques se trouvent dans la propriété '**remarks**'.

- Il peut s'agir d'un problème avec la déclaration Dimona. Dans ce cas, le problème doit être réglé dans le service en ligne 'Dimona'. Lorsque ce problème est résolu dans Dimona, il suffit d'attendre pour que cela se répercute dans ClaO (voir le point suivant, 'Batch').
- Il peut s'agir d'un problème avec la déclaration de travaux. Cela se résout dans le service en ligne 'Déclaration de travaux'. Par exemple, la remarque '12. Pas de DDT pour le BCE'. Si vous êtes entrepreneur-déclarant, cela signifie qu'un travailleur d'une entreprise non déclarée dans Déclaration de travaux (DDT) a travaillé pour votre déclaration. Vous devez dans ce cas, corriger la déclaration de travaux au plus vite, ou avertir l'ONSS.
- Il peut aussi s'agir de problème d'utilisation de Check In and Out at Work par vos travailleurs. Par exemple, la remarque '21. Enregistrement OUT manquant', signifie que nous avons reçu deux pointages 'IN' d'affilé pour ce travailleur.

## Batch

Un système de traitement par lots (**batch**) s'exécute quotidiennement et recalcule les remarques des présences ayant le statut **REGISTERED** ou **FAILED**. Chaque jour, le batch recalculera toutes les remarques des présences pour les dates suivantes :

Soit D étant la date à laquelle le batch tourne :

- **D - 1** (les remarques de la veille)
- **D - 7** (les remarques créées il y a 1 semaine)
- **M - 1** (les remarques créées il y a 1 mois)
- **M - 3** (les remarques créées il y a 3 mois)

Il est donc inutile d'effectuer des appels chaque minute pour un enregistrement de présence créé il y a, par exemple, 5 jours. En effet, ces remarques ne seront pas re-traitées avant le prochain batch, qui sera exécuté 2 jours plus tard : le batch 'D - 7'.

Bien que 95 % des enregistrement de présences soient traitées dans un délai de 10 secondes (en fonction du trafic), nous recommandons les maxima suivants en termes d'appels pour obtenir les remarques des présences nouvellement créées :

Temps après la création de la présence	Fréquence d'appel pour récupérer les remarques
de 0 à 1 minute	<p>Un seul appel toutes les 5 secondes tant que le statut de la remarque est à <b>REGISTERED</b>. En effet, si son statut est à :</p> <ul style="list-style-type: none"> <li>• <b>FAILED</b> : les remarques n'évolueront plus avant J + 1, car elles ont été traitées.</li> <li>• <b>VALIDATED</b> : les remarques n'évolueront plus car la présence a été considérée comme valide.</li> <li>• <b>CANCELLED</b>: les remarques n'évolueront plus car la présence a été annulée.</li> </ul>
à jour J + 1	Un appel unique. Les remarques ne seront plus traitées avant le prochain batch (à jour J + 7), et uniquement si le statut de la présence est <b>REGISTERED</b> ou <b>FAILED</b> .
à jour J + 7	Un appel unique. Les remarques ne seront plus traitées avant le prochain batch (à jour J + 30), et uniquement si le statut de la présence est <b>REGISTERED</b> ou <b>FAILED</b> .
à jour M + 1	Un appel unique. Les remarques ne seront plus traitées avant le prochain batch (à jour M + 1), et uniquement si le statut de la présence est <b>REGISTERED</b> ou <b>FAILED</b> .
à jour M + 3	Un appel unique. Les remarques ne seront plus traitées, et uniquement si le statut de la présence est <b>REGISTERED</b> ou <b>FAILED</b> .

# Documentation technique

## RegisterInBulk

### Description

La méthode **registerInBulk** permet de créer une ou plusieurs présences (jusqu'à 200 par requête). Ces présences seront ensuite traitées de manière asynchrone par ClaO.

### Requête

HTTP POST /presenceRegistrations/registerInBulk

### Body

Le body de la requête doit contenir une liste d'objets 'presence', semblable à celui-ci :

```
{
  "items": [
    {...},
    {...}
  ]
}
```

Chaque objet doit correspondre à la description ci-dessous :

Nom de la propriété	Type de valeur	Obligatoire	Description
<b>registrationDate</b>	string <date-time>	Oui	La date à laquelle le IN ou le OUT a eu lieu, au <b>format ISO 8601 standard</b> : <b>YYYY-MM-DDTHH:MM:SSZ</b> <ul style="list-style-type: none"><li>• <b>YYYY-MM-DD</b> pour la date</li><li>• <b>HH:MM:SS</b> pour l'heure</li><li>• <b>Z</b> pour représenter la zone ou le fuseau horaire dans lequel la date est donnée</li></ul>

<b>ssin</b>	string <^\d{11}\$>	Oui	Le NISS du travailleur qui a fait le IN ou le OUT
<b>type</b>	string ["IN", "OUT"]	Oui	Le type de présence : <ul style="list-style-type: none"> <li>• IN : début de travail ou fin de pause</li> <li>• OUT : début de pause ou fin de travail</li> </ul>
<b>employer</b>	Objet	Oui	L'employeur pour lequel la présence a été enregistrée. Cet objet doit obligatoirement contenir une et une seule de ces 2 propriétés: <ul style="list-style-type: none"> <li>• <b>enterpriseNumber</b> : <b>string</b> &lt;^[0 1]\d{9}\$&gt; : le numéro BCE de l'employeur (employeur belge)</li> <li>• <b>foreignVatNumber</b> : <b>string</b> &lt;longueur maximale 255&gt; : le numéro de TVA de l'employeur (employeur étranger)</li> </ul>
<b>placeOfWork</b>	Objet	Oui	Le lieu de travail où la présence a été enregistrée. Cet objet doit obligatoirement contenir une et une seule de ces 2 propriétés : <ul style="list-style-type: none"> <li>• <b>coordinates</b> : <b>objet</b> : les coordonnées du lieu de prestation sur lequel le travailleur enregistre sa présence (au format <b>WGS84</b>) <ul style="list-style-type: none"> <li>○ <b>longitude</b> : <b>number</b> : la longitude de la coordonnées (X)</li> <li>○ <b>latitude</b> : <b>number</b> : la latitude de la coordonnées (Y)</li> </ul> </li> <li>• <b>description</b> : <b>string</b> &lt;longueur maximale 255&gt; : champ libre pour décrit le lieu de prestation</li> </ul>

<b>contractualRelationshipReference</b>	<code>string &lt;^[A-HJ-NP-Z0-9]{13}\$&gt;</code>	Oui	La référence DDT du contrat entre le client/donneur d'ordre et le déclarant
---	---	-----	---

## Réponse

Pour chaque présence envoyée dans la requête, un objet associé est présent dans la réponse :

```
{
  "items": [
    {...},
    {...}
  ]
}
```

Chaque objet correspond à la description si dessous :

Nom de la propriété	Type de valeur	Obligatoire	Description
<b>createdPresenceRegistration</b>	Objet	Non	<ul style="list-style-type: none"> <li>• Si une erreur s'est produite, cette propriété est nulle</li> <li>• Si l'enregistrement de présence a bien été enregistré, cette propriété contient la présence enregistrée (avec, donc, son <i>id</i> unique)</li> </ul>

<b>notCreatedPresenceRegistration</b>	Objet	Non	<ul style="list-style-type: none"> <li>• Si une erreur s'est produite, cet objet possède 2 propriétés : <ul style="list-style-type: none"> <li>○ <b>presenceRegistrationSubmitted</b> : qui décrit l'enregistrement de présence telle que soumis par l'utilisateur (qu'il convient donc de corriger)</li> <li>○ <b>errorList</b> : une liste d'erreur, expliquant pourquoi l'enregistrement de présence n'a pas été enregistré. Chaque erreur possède 2 propriétés: <ul style="list-style-type: none"> <li>▪ <b>errorCode</b> : un code technique, unique à chaque erreur</li> <li>▪ <b>errorDescription</b> : une description textuelle de l'erreur</li> </ul> </li> </ul> </li> </ul>
---------------------------------------	-------	-----	---

Une présence ainsi créée possède l'ensemble des propriétés décrite dans le endpoint '**Read by id**'.

## Exemple

### Requête

Voici un exemple de requête contenant deux enregistrements de présence. Le premier est valide, le second ne l'est pas.

```
HTTP POST /presenceRegistrations/registerInBulk
```

```
{
  "items": [
    {
      "registrationDate": "2019-08-28T14:15:22Z",
      "ssin": "22663312345",
      "type": "in",
      "employer": {
        "enterpriseNumber": "450905686"
      }
    }
  ]
}
```

```
    },
    "placeOfWork": {
      "coordinates": {
        "longitude": 25.485606,
        "latitude": 20.673302
      }
    },
    "contractualRelationshipReference": "1Y1ZZZZZZZZZZ"
  },
  {
    "registrationDate": "2019-08-28T14:15:22Z",
    "ssin": "22663312345",
    "type": "in",
    "employer": {
      "enterpriseNumber": "4509056866666"
    },
    "placeOfWork": {
      "coordinates": {
        "longitude": 25.485606,
        "latitude": 20.673302
      }
    },
    "contractualRelationshipReference": "1Y1ZZZZZZZZZZ"
  }
]
}
```



## Réponse

```
[
  {
    "createdPresenceRegistration": {
      "id": 17611,
      "registrationDate": "2019-08-28T15:15:22+02:00",
      "ssin": "22663312345",
      "type": "in",
      "employer": {
        "enterpriseNumber": "450905686",
        "foreignVatNumber": null
      },
      "placeOfWork": {
        "coordinates": {
          "longitude": 25.485606,
          "latitude": 20.673302
        },
        "description": null
      },
      "contractualRelationshipReference": "1Y1ZZZZZZZZZZ",
      "activity": "cleaning",
      "channel": "ws",
      "customReference": null,
      "status": {
        "code": "registered",
        "date": "2024-02-27T07:00:33.460412229+01:00"
      },
      "remarks": []
    },
    "notCreatedPresenceRegistration": null
  },
  {
    "createdPresenceRegistration": null,
```

```
"notCreatedPresenceRegistration": {
  "presenceRegistrationSubmitted": {
    "id": null,
    "registrationDate": "2019-08-28T14:15:22Z",
    "ssin": "22663312345",
    "type": "in",
    "employer": {
      "enterpriseNumber": "4509056866666",
      "foreignVatNumber": null
    },
    "placeOfWork": {
      "coordinates": {
        "longitude": 25.485606,
        "latitude": 20.673302
      },
      "description": null
    },
    "contractualRelationshipReference": "1Y1ZZZZZZZZZZ",
    "activity": null,
    "channel": null,
    "customReference": null,
    "status": null,
    "remarks": []
  },
  "errorList": [
    {
      "errorCode": "error.presence-registration.creation.enterprise-
number",
      "errorDescription": "enterprise number is not valid"
    },
    {
      "errorCode": "error.presence-registration.creation.contractual-
relationship-reference",
      "errorDescription": "contractual relationship reference is not valid"
```

```
    }
  ]
}
}
```

## Explications

- Le premier enregistrement de présence est tout à fait valide.
  - Dans la réponse, l'objet correspondant possède donc bien une propriété **createdPresenceRegistration** qui décrit la présence créée.
  - Dans la réponse, la propriété **notCreatedPresenceRegistration** est nulle, car il n'y a pas eu d'erreur.
- Le second enregistrement de présence n'est pas valide.
  - En effet, son NISS (**SSIN**) n'est pas valide.
  - Le numéro de d'entreprise de l'employeur n'est pas valide.
  - Ainsi, l'objet correspondant dans la réponse possède une propriété **createdPresenceRegistration** nulle.
  - L'objet possède également une propriété **notCreatedPresenceRegistration**, qui décrit:
    - **presenceRegistrationSubmitted** : la présence soumise, possédant donc une (ou des) erreur(s).
    - **errorList** : la liste des erreurs en question.

## Read by id

### Description

Cette méthode permet d'obtenir les informations relatives à une présence particulière, sur base de son ID technique.

### Requête

Où **id** est l'ID de la présence dont vous voulez obtenir les informations :

```
HTTP GET /presenceRegistrations/{id}
```

### Réponse

Cette méthode peut répondre de différentes façons :

Code de réponse	Interprétation
200	La présence a bien été trouvée.
404	<ul style="list-style-type: none"> <li>l'<b>id</b> mentionné dans la requête ne correspond à aucune présence.</li> </ul> OU <ul style="list-style-type: none"> <li>la présence existe, mais elle n'est pas liée à l'employeur (ou sa chaîne de sous traitance) lié au token utilisé lors de la requête.</li> </ul>

Dans le cas d'un code 200, la réponse contient donc un objet comme tel :

Nom de la propriété	Type de valeur	Description
<b>id</b>	<b>nombre</b>	ID technique unique de l'enregistrement de présence (à utiliser dans le endpoint de consultation)
<b>registrationDate</b>	<b>string &lt;date-time&gt;</b>	La date à laquelle le IN ou le OUT a eu lieu
<b>ssin</b>	<b>string &lt;^\d{11}\$&gt;</b>	Le NISS du travailleur qui a fait le IN ou le OUT
<b>type</b>	<b>string ["IN", "OUT"]</b>	Le type de présence : <ul style="list-style-type: none"> <li>IN</li> <li>OUT</li> </ul>
<b>employer</b>	Objet	L'employeur pour lequel l'enregistrement de présence a été enregistré. Cet objet doit obligatoirement contenir une et une seule de ces 2 propriétés : <ul style="list-style-type: none"> <li><b>enterpriseNumber</b> : <b>string &lt;^[01]\d{9}\$&gt;</b> : le numéro BCE de l'employeur (employeur belge)</li> <li><b>foreignVatNumber</b> : <b>string &lt;longueur maximale 255&gt;</b> : le numéro de TVA de l'employeur (employeur étranger)</li> </ul>

<b>placeOfWork</b>	Objet	<p>L'endroit où la présence a été enregistrée. Cet objet doit obligatoirement contenir une et une seule de ces 2 propriétés :</p> <ul style="list-style-type: none"> <li>• <b>coordinates</b> : objet : les coordonnées du lieu de prestation sur lequel le travailleur enregistre sa présence (au format <b>WGS84</b>) <ul style="list-style-type: none"> <li>○ <b>longitude</b> : <b>number</b> : la longitude de la coordonnées (X)</li> <li>○ <b>latitude</b> : <b>number</b> : la latitude de la coordonnées (Y)</li> </ul> </li> <li>• <b>description</b> : <b>string</b> &lt;longueur maximale 255&gt; : champ libre qui décrit le lieu de prestation</li> </ul>
<b>contractualRelationshipReference</b>	<b>string</b> <^[A-HJ-NP-Z0-9]{13}\$>	La référence DDT du contrat entre le client/donneur d'ordre et le déclarant
<b>activity</b>	<b>string</b>	Décrit le type d'activité réalisée par le travailleur
<b>channel</b>	<b>string</b>	<p>Le canal via lequel la présence a été enregistrée:</p> <ul style="list-style-type: none"> <li>• <b>MOBILE_URL</b></li> <li>• <b>MOBILE_MANUAL</b></li> <li>• <b>WS</b></li> <li>• <b>WEB_APP</b></li> </ul>
<b>customReference</b>	<b>string</b>	Référence personnalisable pour éventuellement faciliter l'intégration (cela permet de garder un lien entre l'ID technique de l'application source par exemple)
<b>status</b>	Objet	<p>Le statut de la présence. Cet objet possède 2 propriétés:</p> <ul style="list-style-type: none"> <li>• <b>Code</b> : le code du statut actuel de la présence: <ul style="list-style-type: none"> <li>○ <b>REGISTERED</b></li> <li>○ <b>VALIDATED</b></li> <li>○ <b>CANCELLED</b></li> <li>○ <b>FAILED</b></li> </ul> </li> <li>• <b>Date</b> : la date à laquelle le statut a été changé la dernière fois</li> </ul>

remarks	Liste d'objet 'remarque'	<p>Détient une liste de toutes les remarques calculées pour cette présence. Un objet remarque possède les propriétés suivantes:</p> <ul style="list-style-type: none"> <li>• <b>Code</b> : le code unique de la remarque</li> <li>• <b>Labels</b> : <ul style="list-style-type: none"> <li>○ <b>nl</b> : le nom de la remarque en néerlandais</li> <li>○ <b>fr</b> : le nom de la remarque en français</li> <li>○ <b>de</b> : le nom de la remarque en allemand</li> <li>○ <b>en</b> : le nom de la remarque en anglais</li> </ul> </li> </ul>
---------	--------------------------	--

## Exemple

### Requête

```
HTTP GET /presenceRegistrations/17053
```

### Réponse

```
{
  "id": 17053,
  "registrationDate": "2024-01-30T13:58:53.774+01:00",
  "ssin": "22663312345",
  "type": "in",
  "employer": {
    "enterpriseNumber": "70010100188",
    "foreignVatNumber": null
  },
  "placeOfWork": {
    "coordinates": {
      "longitude": 4.348314,
      "latitude": 50.839552
    },
    "description": null
  },
  "contractualRelationshipReference": "1Y1ZZZZZZZZZZ",
```

```

"activity": "cleaning",
"channel": "mobile_manual",
"customReference": null,
"status": {
  "code": "failed",
  "date": "2024-01-30T13:58:59.930499+01:00"
},
"remarks": [
  {
    "code": "ciao_26",
    "labels": {
      "nl": "Gps-coördinaten niet gevonden in AVW",
      "fr": "Coordonnées GPS introuvables pour la DDT",
      "de": "GPS-Koordinaten für Arbeitsmeldung nicht gefunden",
      "en": "GPS coordinates not found for DOW"
    }
  },
  {
    "code": "ciao_34",
    "labels": {
      "nl": "Registratie geweigerd door CAW, controleer de gegevens",
      "fr": "Enregistrement refusé par CAW, veuillez vérifier les
données",
      "de": "Registrierung von CAW abgelehnt, bitte überprüfen Sie die
Daten",
      "en": "Registration refused by CAW, please check data"
    }
  }
]
}

```

## Search

### Description

Cette méthode permet de rechercher des présences correspondant à une liste de critères. En réponse, elle fournit une liste paginée des enregistrements de présences qui satisfont ces critères.

Une entreprise peut faire des recherches pour ses propres travailleurs ou ses sous-traitants.

Vous êtes éditeur de software ? Dans ce cas, vous devrez utiliser le certificat de votre client pour récupérer les enregistrements de présence. En effet, votre certificat permet la création d'enregistrements, mais pas la lecture de ceux-ci.

### Requête

```
HTTP POST /presenceRegistrations/search
```

### Body

Le **body** de la requête doit contenir un objet contenant une propriété **criteria** :

```
{
  "criteria": {
    ...
  }
}
```



Le corps de la requête doit répondre à la description ci-dessous :

Nom de la propriété	Type de valeur	Obligatoire	Valeur par défaut	Description
<b>criteria</b>	objet	Oui		<p>Il s'agit des critères de recherche. Ces critères possèdent exactement les mêmes propriétés que l'objet présence décrit au point <b>Read by id</b> auxquelles s'ajoutent la propriété obligatoire suivante:</p> <ul style="list-style-type: none"> <li>• <b>registrationDate</b> : <ul style="list-style-type: none"> <li>○ <b>startDate</b> : string &lt;date-time&gt;: la date de début sur base de laquelle filtrer les enregistrement de présences</li> <li>○ <b>endDate</b> : string &lt;date-time&gt;: la date de fin sur base de laquelle filtrer les enregistrements de présences</li> </ul> </li> </ul> <p>Les enregistrement de présences qui résulteront de recherche posséderont donc une date d'enregistrement comprise entre registrationDate.startDate et registrationDate.endDate</p>
<b>sort</b>	objet	Non	<pre>{   "direction": "desc",   "ignoreCase": false,   "property": "registrationDate" }</pre>	<p>Un objet permettant de décrire la façon dont les résultats sont triés:</p> <ul style="list-style-type: none"> <li>• <b>direction</b> : ["ASC", "DESC"] pour ascendant ou descendant</li> <li>• <b>ignoreCase</b> : pour préciser si la recherche doit tenir compte de la casse ou pas</li> <li>• <b>property</b> : le nom de la propriété sur laquelle faire le tri</li> </ul>

## Autres paramètres

La requête peut contenir les paramètres de requête suivants :

Nom de la propriété	Type de valeur	Obligatoire	Valeur par défaut	Description
<b>page</b>	nombre	Non	1	La page à laquelle vous voulez accéder
<b>pageSize</b>	nombre	Non	50	Le nombre d'éléments par page

## Réponse

Code de réponse	Interprétation
200	La recherche s'est bien déroulée, les résultats sont présents dans la réponse
500	Une erreur est survenue, vérifiez que les critères de recherche sont bien correctement formatés

Dans le cas d'un code 200, la recherche renvoie un objet correspondant à la description suivante :

Nom de la propriété	Type de valeur	Description
<b>items</b>	Liste d'objets	La liste des présences répondant aux critères de recherche. Les présences sont au format décrit au point <b>Read by ID</b>
<b>first</b>	string	Le lien vers la première page des résultats de la recherche
<b>last</b>	string	Le lien vers la dernière page des résultats de la recherche
<b>prev</b>	string	Le lien vers la page précédente des résultats de la recherche
<b>next</b>	string	Le lien vers la page suivante des résultats de la recherche
<b>page</b>	nombre	Le numéro de la page actuelle des résultats de recherche
<b>pageSize</b>	nombre	La taille de la page actuelle des résultats de recherche

<b>sort</b>	objet	<p>Un objet permettant de décrire la façon dont les résultats sont triés:</p> <ul style="list-style-type: none"> <li>• <b>direction</b> : ["ASC", "DESC"] pour ascendant ou descendant</li> <li>• <b>ignoreCase</b> : pour préciser si la recherche doit tenir compte de la casse ou pas</li> <li>• <b>property</b> : le nom de la propriété sur laquelle faire le tri</li> </ul>
<b>total</b>	nombre	Le nombre total d'éléments résultant de cette recherche
<b>totalPages</b>	nombre	Le nombre total de pages résultant de cette recherche

## Exemple

### Requête

HTTP POST /presenceRegistrations/search

```
{
  "criteria": {
    "registrationDate": {
      "startDate": "2024-01-30T10:12:52+01:00",
      "endDate": "2024-02-15T10:12:54+01:00"
    },
    "type": "in"
  }
}
```

### Réponse

```
{
  "items": [
    {
      "id": 17053,
      ...
    },
  ],
}
```

```

    {
      "id": 17054,
      ...
    }
  ],
  "first":
"/REST/presenceRegistration/v1/presenceRegistrations/search?page=1&pageSize=50"
,
  "last":
"/REST/presenceRegistration/v1/presenceRegistrations/search?page=2&pageSize=50"
,
  "prev": null,
  "next":
"/REST/presenceRegistration/v1/presenceRegistrations/search?page=2&pageSize=50"
,
  "page": 1,
  "pageSize": 50,
  "sort": {
    "direction": "desc",
    "ignoreCase": false,
    "property": "registrationDate"
  },
  "total": 52,
  "totalPages": 2
}

```

## Explications

Cette requête liste l'ensemble des enregistrement de présences, qui :

- possèdent une date d'enregistrement comprise entre le 30/01/2024 à 10:12 et le 15/02/2024 à 12:54 (champ '**registrationDate**' dans la requête)
- sont de type 'in' (champ '**type**' dans la requête)

Un critère est ajouté de manière transparente à la requête. Ces présences ont toutes été enregistrées pour l'employeur lié au token utilisé dans la requête.

## Environnements et URL

Environment	Adresse
Acceptation	<a href="https://services-acpt.socialsecurity.be/REST/presenceRegistration/v1">https://services-acpt.socialsecurity.be/REST/presenceRegistration/v1</a>
Production	<a href="https://services.socialsecurity.be/REST/presenceRegistration/v1">https://services.socialsecurity.be/REST/presenceRegistration/v1</a>

