Guide de l'utilisateur

Webservice REST presenceRegistration

Check in and out at work

Version	Date	Nature des modifications
1.0	01/03/2023	Première version
1.1	03/06/2024	Réécriture des explications concernant OAuth
1.2	09/08/2024	Intégration des modifications de la release 1.4
1.3	19/09/2025	Ajout d'une explication concernant l'"erreur 400", avec une référence à la page https://www.rest-documentation.socialsecurity.be/ pour l'accès au WS presenceregistration
1.4	18/08/2026	Intégration des modifications de la release 2.0.0

Contenu

Enregistrement des présences dans l'application Check in and out at work 3

Préparation 4

- Swagger 5

Principe de fonctionnement 6

- Environnements 6
- Editeurs de logiciels 6
- Asynchronisme 8
- Déroulement classique 8
  - Création 8
  - Consultation 10
  - Délais de traitement 10
- Les remarques 11

Documentation technique 14

- RegisterInBulk 14
  - Description 14
  - Requête 14
  - Réponse 16
  - Exemple 17
  - Explications 19
- Read by id 19
  - Description 19
  - Requête 20
  - Réponse 20
  - Exemple 23
- Search 24
  - Description 24
  - Requête 24
  - Autres paramètres 26
  - Réponse 26
  - Exemple 27

Enregistrement des présences dans l'application Check in and out at work

Un enregistrement de présence correspond à l'entrée ou à la sortie d'un travailleur sur un lieu de travail.

Deux types d'enregistrements sont possibles en fonction du travail réalisé :

- IN : début d'une prestation ou fin d'une pause, ou arrivée sur le lieu de travail ;
- OUT : fin d'une prestation ou début d'une pause, ou départ du lieu de travail.

Chaque enregistrement de présence contient les informations suivantes :

- Le travailleur : identifié au moyen de son numéro d'identification à la sécurité sociale (NISS) ;
- Le lieu de travail : identifié soit par des coordonnées géographiques, soit par une adresse ;
- Le contrat entre le donneur d'ordre et l'entreprise: identifié par la référence dans la déclaration Catena ;
- La date et l'heure : correspondant au moment de l'enregistrement.

Check in and out at work (CIaO) est le service mis à disposition pour l'enregistrement et la consultation des présences des travailleurs des secteurs concernés.

Deux modes d'enregistrement sont disponibles :

- l'application mobile CIaO ;
- le webservice REST PresenceRegistration, éventuellement intégré à un système de pointage existant.

Le présent document décrit les aspects fonctionnels du webservice REST PresenceRegistration.

Préparation

L'accès au webservice REST PresenceRegistration nécessite la réalisation de plusieurs étapes préalables, décrites dans la documentation REST de la Sécurité sociale. (https://www.rest-documentation.socialsecurity.be/)

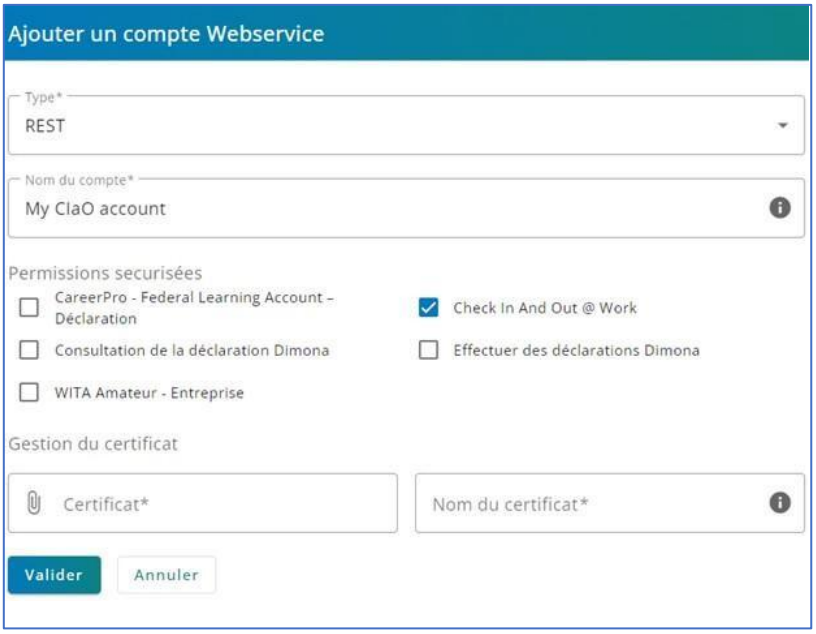
Les entreprises qui utilisent déjà d'autres services de la Sécurité sociale ont peut-être déjà effectué certaines des étapes décrites ci-dessous.

Les prérequis principaux sont les suivants :

- identification de l'employeur ;
- obtention ou création d'un certificat ;
- création d'un compte webservice dans Channel Management. (Activating and managing your REST channel in Chaman | REST Documentation.)

Pour Check in and out at work, le compte webservice doit :

- être de type REST ;
- disposer de l'autorisation sécurisée « Check In And Out @ Work »



Swagger

Le Swagger est un fichier au format YAML, un standard de représentation de données. Le Swagger PresenceRegistration décrit les méthodes et les zones. Vous le trouverez sur le portail de la sécurité sociale, en bas de la page suivante :
https://www.socialsecurity.be/site_fr/employer/applics/check-in-and-out-at-work/general/how-to-register.htm

Principe de fonctionnement

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

- d'enregistrer les présences de leurs travailleurs dans Check in and out at work.
- de consulter les présences de leurs travailleurs, et de consulter les éventuelles remarques.

Environnements

Deux environnements sont mis à disposition. Les points de terminaison (endpoints) sont les suivants :

Environnement	Adresse
Simulation	https://services-sim.socialsecurity.be/REST/presenceRegistration/v1
Production	https://services.socialsecurity.be/REST/presenceRegistration/v1

Les tests doivent être réalisés sur l'environnement de Simulation. Concernant Chaman et OAuth, l'environnement de Production est utilisé à la fois pour la Simulation et la Production du webservice REST PresenceRegistration.

L'environnement de Production de REST PresenceRegistration ne doit contenir que des données de prestations réellement effectuées.

Editeurs de logiciels

Cette partie s'adresse aux éditeurs de logiciels qui souhaitent proposer leurs services aux entreprises concernées par Check in and out at work.

Cas d'usage : transmission d'enregistrements

Lorsqu'une solution logicielle se limite à transmettre des enregistrements de présence vers la Sécurité sociale, l'utilisation du certificat propre de l'éditeur est suffisante.

Cas d'usage : consultation des enregistrements

Lorsqu'une solution permet également la consultation des enregistrements de présence ou des remarques associées, l'utilisation du certificat du client est requise.

Dans ce contexte, le client doit :

- générer un certificat de type self-signed ;
- enregistrer ce certificat dans Channel Management ;
- mettre à disposition l'environnement hébergeant la solution.

Asynchronisme

Le calcul des remarques associées à un enregistrement de présence n'est pas réalisé de manière synchrone.

En effet, ce traitement repose sur plusieurs vérifications impliquant d'autres systèmes et services de la Sécurité sociale.

Pour cette raison, le webservice REST PresenceRegistration distingue :

- l'enregistrement des présences ;
- la consultation des présences et de leurs remarques.

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 Check in and out at work, 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**' : 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**.

- **400, 401, 403 …** : Le serveur ne peut pas traiter la requête. La cause provient du **client**. Deux raisons possibles:
    - **Non-conformité avec l'openAPI.** Quelques exemples:
        - Le numéro NISS n'a pas le format correct.
            - Exemple de code d'erreur lorsqu'un NISS est trop long: "[Path '/items/0/ssin'] ECMA 262 regex \"^\\d{11}$\" does not match input string \"904101963209\""
            - Signification : un numéro NISS doit toujours comporter exactement 11 caractères (NB : dans cet exemple, il y en a 12).
        - Le référence déclaration (Catena) n'a pas le format correct.
            - Exemple de code d'erreur lorsqu'on ajoute deux tirets à la référence déclaration (Catena) : "[Path '/items/0/contractualRelationshipReference'] ECMA 262 regex \"^[A-HJ-NP-Z0-9]{13}$\" does not match input string \" 2CE-0000001A9-Z \""
            - Signification : une référence déclaration (Catena) doit toujours comporter exactement 13 caractères (NB : dans cet exemple, il y en a 15).
        - Le numéro d'entreprise n'a pas le bon format.
            - Exemple de numéro d'entreprise où il manque un « 0 » au début: "[Path '/items/0/employer/enterpriseNumber'] ECMA 262 regex \"^[0|1]\\d{9}$\" does not match input string \"406798006\""
            - Signification : un numéro d'entreprise commence toujours par un « 0 » ou un « 1 », suivi de 9 chiffres.
        - Le type d'enregistrement (in ou out) est manquant.
            - Exemple de code d'erreur: "[Path '/items/0'] Object has missing required properties ([\"type\"])"
    - **Les règles de sécurité ne sont pas respectées** (problème d'**authentification/autorisation**).
        - Quelques exemples :
            - "401 unauthorized"
            - "403 forbidden"
        - Des informations complémentaires sont disponibles à l'adresse suivante : Additional guidelines for calling REST APIs | REST Documentation
- **500** : une erreur non gérée a eu lieu, les enregistrements de présence 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 des webservices 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 Check in and out at work traite les remarques avant d'essayer de les consulter.

Délais de traitement

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

- Chaque enregistrement de présence possède une propriété **"validity"**. Cette propriété permet de suivre le traitement de l'enregistrement. Les différentes valeurs suivantes sont possibles :
  - **PENDING** : 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.
  - **FAILED** : l'enregistrement a été traité et des remarques ont été générées. Ces remarques peuvent évoluer lors des recalculs ultérieurs (voir section « Batch »). L'enregistrement demeure valide et consultable.

Interprétation du statut "failed"

Le statut "failed" indique que l'enregistrement a été accepté et stocké avec succès par l'ONSS.
Toutefois, une ou plusieurs remarques ont été générées lors des contrôles effectués après l'enregistrement.
Ces remarques sont disponibles dans la propriété "remarks".

La présence d'un statut "failed" n'empêche pas l'existence de l'enregistrement de présence. Celui-ci demeure enregistré et consultable.

- 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, la correction sera automatiquement prise en compte dans Check in and out at work lors des traitements ultérieurs.
- Il peut s'agir d'un problème avec la déclaration dans Catena. Par exemple, la remarque "12. L'entreprise n'a pas de contrat dans la déclaration Catena". Si vous êtes entrepreneur-déclarant, cela signifie qu'un travailleur d'une entreprise non déclarée dans Catena a travaillé pour votre déclaration. Dans ce cas, la déclaration Catena doit être corrigée dans les meilleurs délais.
- 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 enregistrements de présence "IN" d'affilée pour ce travailleur.

Les remarques

Voici les différentes remarques possibles. Les actions à entreprendre sont décrites dans le document suivant : Procédures en cas de problème technique, négligence et erreur.

Code de la remarque	Libellé
CAW_1	Il n'existe pas de relation entre le travailleur et l'entreprise
CAW_2	La relation entre le travailleur et l'entreprise n'est plus active
CAW_3	L'entreprise n'est pas un employeur ou un indépendant
CAW_4	L'entreprise n'existe pas
CAW_5	La personne enregistrée n'est pas titulaire d'une fonction légale pour cette entreprise
CAW_6	La déclaration Limosa n'existe pas pour cet identifiant
CAW_7	La déclaration Limosa existe mais n'est pas active
CAW_8	Il n'existe pas de Limosa contenant ce NISS-BIS
CAW_9	Il n'existe pas de Limosa active contenant ce NISS-BIS
CAW_10	La déclaration dans Catena n'existe pas pour cet identifiant
CAW_11	La déclaration dans Catena existe mais n'est pas active

Code de la remarque	Libellé
CAW_12	L'entreprise n'a pas de contrat dans Catena
CAW_14	Un enregistrement similaire existe déjà
CAW_15	Il n'existe personne avec ce NISS
CAW_16	L'employeur est inactif dans le répertoire
CAW_17	Le contrat est inactif dans Catena
CIAO_21	Deux ou plusieurs IN d'affilée
CIAO_22	Deux ou plusieurs OUT d'affilée
CIAO_24	OUT sans IN dans les 24 heures précédentes
CIAO_25	Pas de coordonnées GPS remontées
CIAO_32	Délai de réception des enregistrements dépassé
CIAO_33	Il n'existe pas de Limosa contenant ce NISS-BIS
CIAO_35	Service geocoding indisponible
CIAO_36	Coordonnées GPS incorrectes
CIAO_37	Adresse non vérifiable en raison d'informations manquantes
CIAO_38	Impossible de déterminer l'adresse sur base de l'adresse encodée

Batch

Un système de traitement par lots (**batch**) s'exécute quotidiennement et recalcule les remarques des présences ayant la "validity": " **failed".** Chaque jour, le batch recalcule les remarques associées aux 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)
- **D - 30** (les remarques créées il y a 1 mois)

- **D - 90** (les remarques créées il y a 3 mois)

Afin d'éviter une charge inutile sur le service, il est recommandé de limiter les consultations aux moments où un recalcul des remarques est effectivement susceptible d'avoir lieu.

Un traitement batch est exécuté quotidiennement et recalcule les remarques des enregistrements dont la "validity" est "failed" selon les échéances définies ci-dessous.

Temps après la création de la présence	Fréquence d'appel pour récupérer les remarques
de 0 à 1 minute	Un seul appel toutes les 5 secondes tant que la validité est à **pending**. En effet, si sa validité est : • **failed** : les remarques n'évolueront plus avant J + 1, car elles ont été traitées. • **validated** : les remarques n'évolueront plus car la présence a été considérée comme valide.
à jour J + 1	Un appel unique. Les remarques ne seront plus traitées avant le prochain batch (à jour J + 7), et uniquement si la validité de la présence est **failed**.
à jour J + 7	Un appel unique. Les remarques ne seront plus traitées avant le prochain batch (à jour J + 30), et uniquement si la validité de la présence est **failed**.
à jour J + 30	Un appel unique. Les remarques ne seront plus traitées avant le prochain batch (à jour J + 30), et uniquement si la validité de la présence est **failed**.
à jour J + 90	Un appel unique. Les remarques ne seront plus traitées, et uniquement si la validité de la présence est **failed**.

Documentation technique

RegisterInBulk

Description

La méthode registerInBulk permet de créer une ou plusieurs présences (jusqu'à 200 par requête) à condition que les enregistrements aient lieu en temps réel. Les enregistrements avec un retard de plus de 10 minutes ne sont pas acceptés. Les enregistrements IN ou OUT sont ensuite traités de manière asynchrone par Check in and out at work.

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
registrationDate	string <date-time>	Oui	La date à laquelle le IN ou le OUT a eu lieu, au **format ISO 8601 standard** : **YYYY-MM-DD*T*HH:MM:SS*Z*** • **YYYY-MM-DD** pour la date • **HH:MM:SS** pour l'heure • **Z** pour représenter la zone ou le fuseau horaire dans lequel la date est donnée
ssin	string <^\d{11}$>	Oui	Le NISS du travailleur qui a fait le IN ou le OUT

Nom de la propriété	Type de valeur	Obligatoire	Description
type	string ["IN", "OUT"]	Oui	Le type de présence : • IN : début d'une prestation ou fin d'une pause, ou arrivée sur le lieu de travail. • OUT : fin d'une prestation ou début d'une pause, ou départ du lieu de travail.
employer	Objet	Oui	L'employeur pour lequel la présence a été enregistrée. Cet objet doit obligatoirement contenir une et une seule de ces 2 propriétés: • **enterpriseNumber** : **string <^[0|1]\d{9}$>** : le numéro BCE de l'employeur (employeur belge) • **foreignVatNumber** : **string <**longueur maximale 255**>** : le numéro de TVA de l'employeur (employeur étranger)
placeOfWork	Objet	Oui	Le lieu de travail où la présence a été enregistrée. Cet objet doit obligatoirement contenir une et une seule de ces 2 propriétés : • **coordinates** : **objet** : les coordonnées du lieu de prestation sur lequel le travailleur enregistre sa présence (au format **WGS84**) ○ **longitude** : **number** : la longitude de la coordonnées (X) ○ **latitude** : **number** : la latitude de la coordonnées (Y) • **address** : **objet** : l'adresse du lieu de prestation (sans abréviation). ○ **postcode** : **string** : code postal ○ **municipalityName** : **string** : ville ○ **streetName** : **string** : nom de la rue ○ **houseNumber** : **string** : numéro ○ **boxNumber** : **string** : numéro de boite
contractualRelationshipReference	**string <^[A-HJ-NP-Z0-9]{13}$>**	Oui	La référence déclaration dans Catena 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 ci-dessous :

Nom de la propriété	Type de valeur	Obligatoire	Description
createdPresenceRegistration	Objet	Non	• Si une erreur s'est produite, cette propriété est nulle • Si l'enregistrement de présence a bien été enregistré, cette propriété contient la présence enregistrée (avec, donc, son *id* unique)
notCreatedPresenceRegistration	Objet	Non	• Si une erreur s'est produite, cet objet possède 2 propriétés : ○ **presenceRegistrationSubmitted** : qui décrit l'enregistrement de présence tel que soumis par l'utilisateur (qu'il convient donc de corriger) ○ **errorList** : une liste d'erreur, expliquant pourquoi l'enregistrement de présence n'a pas été enregistré. Chaque erreur possède 2 propriétés: ▪ **errorCode** : un code technique, unique à chaque erreur ▪ **errorDescription** : une description textuelle de l'erreur

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

Exemple

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": "22343312345",
      "type": "in",
      "employer": {
        "enterpriseNumber": "450905686"
      },
      "placeOfWork": {
        "coordinates": {
          "longitude": 25.485606,
          "latitude": 20.673302
        }
      },
      "contractualRelationshipReference": "1Y1003SQ5VSSZ"
    },
    {
      "registrationDate": "2019-08-28T14:15:22Z",
      "ssin": "2299z7777000",
      "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,
        "name": "immo corp 123”
      },
      "placeOfWork": {
```

```
        "coordinates": {
          "longitude": 25.485606,
          "latitude": 20.673302
        },
        "address": null
      },
      "worker": {
        "givenName": “John”,
        "familyName": “Smith”
      },
      "contractualRelationshipReference": "1Y1003SQ5VSSZ",
      "activitySector": "REAL_ESTATE",
      "activity": null,
      "channel": "ws",
      "customReference": null,
      "status": {
        "code": "registered",
        "date": "2024-02-27T07:00:33.460412229+01:00"
      },
      "validity": "pending",
      "remarks": null
    },
    "notCreatedPresenceRegistration": null
  },
  {
    "createdPresenceRegistration": null,
    "notCreatedPresenceRegistration": {
      "presenceRegistrationSubmitted": {
        "id": null,
        "registrationDate": "2019-08-28T14:15:22Z",
        "ssin": "22663312345",
        "worker": null,
        "type": "in",
        "employer": {
          "enterpriseNumber": "4509056866666",
          "foreignVatNumber": null,
          "name": null
        },
        "placeOfWork": {
          "coordinates": {
            "longitude": 25.485606,
            "latitude": 20.673302
          },
          "address": null
        },
        "contractualRelationshipReference": "1Y1ZZZZZZZZZZ",
        "activitySector": null,
        "activity": null,
        "channel": null,
        "customReference": null,
        "status": null,
        "validity": 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 valide (il s’agit d’un exemple pour illustrer les deux cas ; en réalité le NISS utilisé n’est pas correct).
    - 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.
    - Le numéro de d'entreprise de l'employeur n'est pas valide.
    - La déclaration dans Catena 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	• l'**id** mentionné dans la requête ne correspond à aucune présence. OU • 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.

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

Nom de la propriété	Type de valeur	Description
id	**nombre**	ID technique unique de l'enregistrement de présence (à utiliser dans le endpoint de consultation)
registrationDate	**string <date-time>**	La date à laquelle le IN ou le OUT a eu lieu
ssin	**string <^\d{11}$>**	Le NISS du travailleur qui a fait le IN ou le OUT
worker	**Object**	Il s'agit du travailleur. Il contient ces 2 propriétés : • givenName : **string** : le prénom • familyName : **string** : le nom de famille

Nom de la propriété	Type de valeur	Description
type	**string ["IN", "OUT"]**	Le type de présence : • IN • OUT
employer	Objet	L'employeur pour lequel l'enregistrement de présence a été enregistré. Cet objet possède 3 propriétés. Les propriétés **enterpriseNumber** et **foreignVatNumber** sont exclusives. • **enterpriseNumber** : **string <^[0|1]\d{9}$>** : le numéro BCE de l'employeur (employeur belge) • **foreignVatNumber** : **string <**longueur maximale 255**>** : le numéro de TVA de l'employeur (employeur étranger) • **name** : **string** : le nom de l'employeur

placeOfWork	Objet	L'endroit où la présence a été enregistrée. Cet objet doit **obligatoirement** contenir **une et une seule** de ces 2 propriétés :
		• **coordinates** : objet : les coordonnées du lieu de prestation sur lequel le travailleur enregistre sa présence (au format **WGS84**)
		  ○ **longitude** : **number** : la longitude de la coordonnées (X)
		  ○ **latitude** : **number** : la latitude de la coordonnées (Y)
		• **address** : **objet** : l'adresse du lieu de prestation sans abréviation.
		  ○ **postcode** : **string** : code postal
		  ○ **municipalityName** : **string** : ville
		  ○ **streetName** : **string** : nom de la rue
		  ○ **houseNumber** : **string** : numéro
		  ○ **boxNumber** : **string** : numéro de boite

contractualRelationshipReference	**string <^[A-HJ-NP-Z0-9]{13}$>**	La référence déclaration dans Catena du contrat entre le client/donneur d'ordre et le déclarant

Nom de la propriété	Type de valeur	Description
activitySector	**string**	Le secteur d'activité de la déclaration
activity (deprecated)	**string**	Décrit le type d'activité réalisée par le travailleur
channel	**string**	Le canal via lequel la présence a été enregistrée : • **MOBILE_URL** • **MOBILE_MANUAL** • **WS** • **WEB_APP**
customReference	**string**	Référence personnalisable pour éventuellement faciliter l'intégration (cela permet de garder un lien entre l'ID technique de l'application source par exemple)

status	Objet	Le statut de la présence. Cet objet possède 2 propriétés:
		• **code** : le code du statut actuel de la présence:
		  ○ **REGISTERED**
		  ○ **EDITED**
		  ○ **CANCELLED**
		• **date** : la date à laquelle le statut a été changé la dernière fois

validity	Object	La validité de la présence: • **pending** • **failed** • **validated**

remarks	Liste d'objet 'remarque'	Détient une liste de toutes les remarques calculées pour cette présence. Un objet remarque possède les propriétés suivantes:
		• **code** : le code unique de la remarque
		• **labels** :
		  ○ **nl** : le nom de la remarque en néerlandais
		  ○ **fr** : le nom de la remarque en français
		  ○ **de** : le nom de la remarque en allemand
		  ○ **en** : le nom de la remarque en anglais

Exemple

Requête

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

Réponse


```
{
      "id": 17053,
      "registrationDate": "2024-01-30T13:58:53.774+01:00",
      "ssin": "22663312345",
      "worker": {
         "givenName": "John",
         "familyName": "Doe"

      "type": "in",
      "employer": {
        "enterpriseNumber": "70010100188",
        "foreignVatNumber": null,
        "name": "immo corp 123”
      },
      "placeOfWork": {
        "coordinates": {
          "longitude": 4.348314,
          "latitude": 50.839552
        },
        "address": {
            "postCode": null,
            "municipalityName": null,
            "streetName": null,
            "houseNumber": null,
            "boxNumber": null
          }
     },
      "contractualRelationshipReference": "1Y1ZZZZZZZZZZ",
      "activitySector": "REAL_ESTATE",
      "activity": null,
      "channel": "mobile_manual",
      "customReference": null,
      "status": {
        "code": "failed",
        "date": "2024-01-30T13:58:59.930499+01:00"
      },
      "validity": "failed",
      "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"
          }
        }
      ]
}
```

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. Pour rechercher les enregistrements d'un sous-traitant, la zone de référence déclaration dans Catena doit être complétée (contractualRelationshipReference).

La consultation des enregistrements de présence nécessite l'utilisation du certificat du client. Le certificat de l'éditeur permet la création d'enregistrements mais ne permet pas leur consultation.

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

criteria	objet	Oui		Il s'agit des critères de recherche. Ces critères possèdent pour la plupart exactement les mêmes propriétés que l'objet présence décrit au point **Read by id** auxquelles s'ajoutent la propriété obligatoire suivante:
				
				• **registrationDate** :
				  ○ **startDate** : string <date-time>: la date de début sur base de laquelle filtrer les enregistrements de présences
				  ○ **endDate** : string <date-time>: la date de fin sur base de laquelle filtrer les enregistrements de présences
				
				Les résultats possèderont donc une date d'enregistrement comprise entre la "registrationDate.startDate" et la "registrationDate.endDate".
				
				En plus de la **registrationDate**, un des critères suivants est obligatoire : **ssin**, **employer**, **contractualRelationshipReference**.

sort	objet	Non	`{ "direction": "desc", "ignoreCase": false, "property": "registrationDate" }`	Un objet permettant de décrire la façon dont les résultats sont triés: • **direction** : ["ASC", "DESC"] pour ascendant ou descendant • **ignoreCase** : pour préciser si la recherche doit tenir compte de la casse ou pas • **property** : le nom de la propriété sur laquelle faire le tri

Autres paramètres

La requête peut contenir les paramètres de requête suivants :

Nom de la propriété	Type de valeur	Obligatoire	Valeur par défaut	Description
page	nombre	Non	1	La page à laquelle vous voulez accéder
pageSize	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
items	Liste d'objets	La liste des présences répondant aux critères de recherche. Les présences sont au format décrit au point **Read by ID**
first	string	Le lien vers la première page des résultats de la recherche
last	string	Le lien vers la dernière page des résultats de la recherche
prev	string	Le lien vers la page précédente des résultats de la recherche
next	string	Le lien vers la page suivante des résultats de la recherche
page	nombre	Le numéro de la page actuelle des résultats de recherche
pageSize	nombre	La taille de la page actuelle des résultats de recherche

Nom de la propriété	Type de valeur	Description
sort	objet	Un objet permettant de décrire la façon dont les résultats sont triés: • **direction** : ["ASC", "DESC"] pour ascendant ou descendant • **ignoreCase** : pour préciser si la recherche doit tenir compte de la casse ou pas • **property** : le nom de la propriété sur laquelle faire le tri
total	nombre	Le nombre total d'éléments résultant de cette recherche
totalPages	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 enregistrements 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.