

# Table des matières

0.1	Démarrer la configuration de Chill . . . . .	3
0.1.1	Se connecter en tant qu'administrateur . . . . .	3
0.1.2	Que faut-il créer? . . . . .	3
0.1.2.1	Premier utilisateurs, et les droits minimums . . . . .	3
0.1.2.2	Le nécessaire pour un premier parcours . . . . .	4
0.1.2.3	Une première activité! . . . . .	4
0.1.2.4	Une configuration minimale... . . . . .	4
0.2	Génération de documents . . . . .	5
0.2.1	Rappel du parcours des utilisateurs . . . . .	5
0.2.2	Syntaxe de génération d'un document . . . . .	5
0.2.3	Indiquer un champ dans un document . . . . .	6
0.2.4	Principes liés au variables . . . . .	6
0.2.4.1	Nommage de variables . . . . .	6
0.2.4.2	Cas où une variable est optionnelle . . . . .	6
0.2.5	Variables disponibles par document . . . . .	6
0.2.5.1	Pour chaque document . . . . .	7
0.2.5.2	Document générés pour un parcours . . . . .	7
0.2.5.3	Document générés pour un parcours, contexte "liste des activités" . . . . .	8
0.2.5.4	Document générés pour une évaluation . . . . .	8
0.2.5.5	Document générés pour un échange . . . . .	9
0.2.5.6	Documents générés dans le dossier d'une personne : contexte "personne basique" . . . . .	9
0.2.5.7	Documents générés dans le dossier d'une personne : contexte "personne avec un tiers" . . . . .	9
0.2.6	Champs par objet . . . . .	9
0.2.6.1	AccompanyingPeriod . . . . .	10
0.2.6.2	AccompanyingPeriodComment . . . . .	11
0.2.6.3	AccompanyingPeriodOrigin . . . . .	11
0.2.6.4	AccompanyingPeriodParticipation . . . . .	12
0.2.6.5	AccompanyingPeriodResource . . . . .	12

0.2.6.6	Activity (échange) . . . . .	12
0.2.6.7	ActivityType . . . . .	13
0.2.6.8	ActivityPresence . . . . .	13
0.2.6.9	ClosingMotive . . . . .	13
0.2.6.10	Date . . . . .	13
0.2.6.11	Person . . . . .	13
0.2.6.12	PersonResource (Ressources associée à l'utilisateur) . . . . .	21
0.2.6.13	PersonResourceKind (Type de personne ressource) . . . . .	21
0.2.6.14	StatutLogement . . . . .	21
0.2.6.15	NiveauEtude . . . . .	21
0.2.6.16	SituationProfessionnelle . . . . .	22
0.2.6.17	Civility . . . . .	22
0.2.6.18	Entourage . . . . .	22
0.2.6.19	Household . . . . .	22
0.2.6.20	HouseholdMember . . . . .	22
0.2.6.21	HouseholdPosition . . . . .	23
0.2.6.22	Mobilite, NiveauEtude, SituationProfessionnelle, StatutLogement, TempsDeTravail, TitreDeSejour . . . . .	23
0.2.6.23	ThirdParty . . . . .	23
0.2.6.24	ThirdPartyCategory (catégorie de tiers) . . . . .	24
0.2.6.25	ThirdPartyProfession (profession du tiers) . . . . .	24
0.2.6.26	Adresse . . . . .	24
0.2.6.27	Postcode . . . . .	25
0.2.6.28	Country . . . . .	25
0.2.6.29	User . . . . .	25
0.2.6.30	UserJob . . . . .	25
0.2.6.31	Location (lieu) . . . . .	26
0.2.6.32	LocationType (type de lieu) . . . . .	26
0.2.6.33	Social Issue (problématique sociale) . . . . .	26
0.2.6.34	Relationship (relations (filiations) entre usagers) . . . . .	26
0.2.6.35	AccompanyingPeriodWorkEvaluation (évaluation dans une action d'accompagnement) . . . . .	26
0.2.6.36	Evaluation (type d'évaluation) . . . . .	27
0.2.6.37	AccompanyingPeriodWork (Action d'accompagnement) . . . . .	27
0.2.6.38	SocialAction (type d'action d'accompagnement) . . . . .	28
0.2.6.39	Result (résultat d'une action) . . . . .	28
0.2.6.40	Goal (objectif d'une action) . . . . .	28

0.2.6.41	AccompanyingPeriodWorkGoal (objectifs et résultats d'une action d'accompagnement) . . . . .	28
0.2.6.42	Comment . . . . .	28

## 0.1 Démarrer la configuration de Chill

A la livraison, Chill ne dispose d'aucune configuration, vous recevez l'instance avec une base de donnée vide.

Voici les premières étapes à configurer pour pouvoir commencer à l'utiliser.

### 0.1.1 Se connecter en tant qu'administrateur

Sur la page d'accueil de votre nouvelle instance, utilisez le login d'administration et le mot de passe qui vous a été fourni.

Vous serez alors redirigé vers l'interface d'administration.

### 0.1.2 Que faut-il créer ?

#### 0.1.2.1 Premier utilisateurs, et les droits minimums

Pour qu'un utilisateur puisse utiliser Chill, voici les opérations à réaliser :

- Créer un Centre.  
Choisissez un nom qui représente votre service. Si vous n'en utilisez qu'un seul, il ne sera que très rarement visible des utilisateurs.
- Créer un Cercle;  
Si vous choisissez de laisser visibles tous les éléments, il ne sera jamais visualisés par les utilisateurs. Dans le cas inverse, utilisez un nom de service qui soit cohérent avec la séparation que vous voulez réaliser.
- Créer un Groupe de permissions  
C'est sans doute la phase la plus complexe : elle vise à créer un groupe de droits cohérents. Chaque droit s'exerce sur un cercle défini.  
Si vous choisissez de laisser accessibles toutes les informations, indiquez les permissions les plus larges, et le cercle que vous avez créé.
- Créer un utilisateur

Les seuls champs obligatoires sont son nom d'utilisateur (qu'il rentrera pour s'authentifier – il n'est pas sensible à la casse), son adresse de courriel et un libellé (le nom qui s'affichera dans tous les menus). Vous devez également indiquer un mot de passe.

Les autres champs sont optionnels.

Une fois créé, associez votre nouvel utilisateur avec le groupe de permission, et le centre.

Avec ces premières étapes, un utilisateur devrait déjà être capable de s'authentifier et, si les droits sont configurés correctement, créer un premier dossier de personne.

### **0.1.2.2 Le nécessaire pour un premier parcours**

Continuez pour pouvoir saisir un premier parcours (si vous utilisez cette fonctionnalité) :

- Créer un “type de localisation”. Le type de localisation permettra d'indiquer le lieu où sera localisé le parcours. Un des types les plus courants est, par exemple, un type “implantation”. Par la suite, vous pourrez créer des types de lieux pour, par exemple, le “domicile de l'usager”, “rendez-vous en hôpital”, etc.
- Créer une “localisation”, que vous allez attacher au “type de localisation” créé précédemment.
- Créer une “Origine” pour les parcours.  
Par exemple : “Appel téléphonique”, “Rencontre en maraude”, etc.
- Créer une première “problématique sociale”  
Impossible de confirmer un parcours sans problématique !
- Créer un Métier  
Le métier sera nécessaire pour pouvoir confirmer le parcours. Vous pouvez associer le premier utilisateur créé au métier : il sera plus aisément sélectionné.

Vous êtes maintenant en mesure de créer un usager et un parcours !

### **0.1.2.3 Une première activité !**

Pour pouvoir créer une activité (tant dans un parcours qu'auprès d'une personne), ajoutez un “type d'activité”.

### **0.1.2.4 Une configuration minimale...**

Cette première configuration est minimale ! Elle suffit pour commencer.

Par la suite, vous pourrez vous plonger dans les différents menus pour alimenter votre instance de Chill et qu'elle soit à votre image.

## 0.2 Génération de documents

L'administrateur fonctionnel prépare la génération des documents.

Cela consiste à configurer le gabarit et ses éventuelles options via l'interface d'administration. Pour chaque gabarit, un document est joint et contient des "zones substituantes", qui permettent d'insérer des informations issues de Chill.

Seuls les documents suivants peuvent être utilisés :

- .odt (LibreOffice Writer);
- .ods
- .odp

### 0.2.1 Rappel du parcours des utilisateurs

Lors de la génération de document, les utilisateurs parcourent trois étapes, dont l'un est optionnelle :

1. Étape optionnelle : un formulaire demande des précisions à l'utilisateur.  
Il peut s'agir, par exemple, de préciser les destinataires du document, de choisir un signataire, etc.  
Ce formulaire est soit :
  - natif au nœud. Dans ce cas, il apparaît systématiquement ou dans certaines conditions;
  - configuré par l'administrateur fonctionnel parmi des options disponibles;
2. le document est effectivement généré en arrière-plan. Cela peut nécessiter éventuellement quelques secondes;
3. le document est ouvert pour édition dans un éditeur en ligne. L'enregistrement est automatique. Lorsqu'ils ferment l'éditeur **depuis l'interface de l'éditeur**, l'utilisateur est redirigé vers l'interface de Chill, généralement la page de génération du document.  
Notez que, pour que la redirection soit effective, l'utilisateur doit fermer **dans l'interface de l'éditeur** : fermer la fenêtre ou l'onglet fait perdre les informations de redirection - cependant, le document est normalement enregistré.

### 0.2.2 Syntaxe de génération d'un document

Les documents sont préparés par l'administrateur fonctionnel.

### 0.2.3 Indiquer un champ dans un document

Aux endroits où cela est nécessaire, l'administrateur indique un "champ substituant".

Cela est accessible via le menu "Insertion > Renvoi...", puis choisir l'onglet "Fonction", "Substituant", "Texte", et indiquer la valeur du champ.

La valeur à indiquer dans le champ "substituant" est à déduire des informations ci-dessous.

### 0.2.4 Principes liés au variables

#### 0.2.4.1 Nommage de variables

TODO

**0.2.4.1.1 Exemple : application aux objets "date"** Les dates sont disponibles dans deux formats :

- le format "court", ou dd/mm/yyyy (par exemple, 15/06/1980, 18/08/2021, ...). Ce format est accessible par le champ `short`;
- le format "long" : 15 juin 1980, 18 août 2021, ... Ce format est accessible par le champ `long`;

lorsqu'une date est disponible, il est donc possible d'avoir accès à son format en précisant sa chaîne de caractère. Exemple avec la date de naissance de la personne :

```
person.Birthdate.Short=15/06/1980
person.Birthdate.Long=15 juin 1980
```

#### 0.2.4.2 Cas où une variable est optionnelle

Lorsqu'une variable est optionnelle, si sa valeur est inconnue ou vide, alors tout ses champs apparaissent avec une chaîne de caractère vide.

L'arbre des variables est toujours identique, sur toute la profondeur de celles-ci. L'administrateur est garanti qu'un champ existera, même si sa valeur n'est pas présente dans la base de donnée.

### 0.2.5 Variables disponibles par document

Pour chaque contexte où un document est généré, les variables disponibles sont listées dans la section suivante. Le type des variables est indiqué.

Les variables pour chaque type sont décrites dans la section suivante :

- si le type commence par une majuscule, alors cette variable comporte des sous-champs, et il faut se reporter à la description des variables pour ce type;
- si le type commence par une minuscule, alors cette variable peut être utilisée directement dans le document :
  - s’il s’agit d’un booléen (`bool`), le champ peut être utilisé dans des tests;
  - les champs `text` et `int` peuvent faire l’objet de test sur l’égalité;
  - les champs de type `int` peuvent faire l’objet de comparaison sur l’ordre de grandeur (par exemple, le champs `age` des objets de type `Person` peut être filtré `> 18` ou `< 18` pour distinguer les adultes des enfants).

### 0.2.5.1 Pour chaque document

**0.2.5.1.1 Paramètres pour l’administrateur fonctionnel** Les administrateurs fonctionnels peuvent activer les paramètres suivants :

- un champ “lieu”, qui permettra de choisir parmi les lieux pré-enregistrés. Le lieu pré-sélectionné sera celui choisi par l’utilisateur;

#### 0.2.5.1.2 Variables

- `creator` : (User) le créateur;
- `createdAt` (Date) : la date et l’heure de création;
- `createdAtDate` (Date) : la date de la création (sans l’heure). Utilisable pour indiquer la date d’un courrier, par exemple;
- `Location` (Location) : le lieu sélectionné par le créateur, au moment de la génération **ou** celui choisi par l’étape 1.

### 0.2.5.2 Document générés pour un parcours

**0.2.5.2.1 Paramètres pour l’administrateur fonctionnel** Les administrateurs fonctionnels peuvent activer les paramètres suivants :

- un champ “usager 1”, qui permet ensuite à l’utilisateur de choisir un usager parmi ceux concernés par le parcours, les interlocuteurs privilégiés qui sont des usagers (à l’exclusion des tiers), et les personnes ressources associées à un usager concerné du parcours (à l’exclusion des ressources tiers et “texte libre”);
- un champ “usager 2”, qui permet aux utilisateurs de choisir un deuxième usager parmi ceux concernés par le parcours, les interlocuteurs privilégiés qui sont des usagers (à l’exclusion des tiers), et les personnes ressources associées à un usager concerné du parcours (à l’exclusion des ressources tiers et “texte libre”);

- un champ “usager principal du parcours”, qui permet, cette fois, de choisir parmi les usagers concernés par le parcours, les interlocuteurs privilégiés qui sont des usagers (à l’exclusion des tiers), et les personnes ressources associées à un usager concerné du parcours (à l’exclusion des ressources tiers et “texte libre”);

#### **0.2.5.2.2 Variables** Le document présente :

- une variable `course`, de type `AccompanyingPeriod`;
- si `usager principal du parcours` est coché, une variable `mainPerson`, de type `Person`, avec les variants `relations`, `household` (ménage) et `budget`;
- si `usager 1` est coché, une variable `person1`, de type `Person`, avec les variants `relations`, `household` (ménage) et `budget`;
- si `usager 2` est coché, une variable `person2`, de type `Person`, avec les variants `relations`, `household` (ménage) et `budget`;

#### **0.2.5.3 Document générés pour un parcours, contexte “liste des activités”**

Le contexte présente les mêmes variables et paramètre que les documents générés par un parcours.

La variable suivante est ajoutée :

- `activities` (liste de `Activity`) : Liste d’activités, variant “light”. Aucun filtre n’est appliqué sur les échanges récupérés.

#### **0.2.5.4 Document générés pour une évaluation**

Le document présente :

- une variable `evaluation` de type `AccompanyingPeriodWorkEvaluation` : l’évaluation concernée;
- une variable `work` de type `AccompanyingPeriodWork` : l’action d’accompagnement au sein de laquelle l’évaluation est générée;
- une variable `course`, de type `AccompanyingPeriod` : le parcours au sein duquel l’évaluation est générée;
- si `usager principal du parcours` est coché, une variable `mainPerson`, de type `Person`, avec les variants `relations`, `household` (ménage) et `budget`;
- si `usager 1` est coché, une variable `person1`, de type `Person`, avec les variants `relations`, `household` (ménage) et `budget`;
- si `usager 2` est coché, une variable `person2`, de type `Person`, avec les variants `relations`, `household` (ménage) et `budget`;



### 0.2.5.5 Document générés pour un échange

Le document présente :

- une variable `activity`, de type `Activity` : l'évaluation concernée
- une variable `course`, de type `AccompanyingPeriod` : le parcours concerné, à condition que l'échange ait été créé dans un contexte parcours
- une variable `person`, de type `Person` : la personne concernée, à condition que l'échange ait été créée dans un contexte d'utilisateur.

Il est possible également d'injecter des dossiers d'utilisateurs, parmi ceux associés **à l'échange** (l'utilisateur peut choisir parmi les utilisateurs de l'échange, et pas les utilisateurs concernés du parcours).

- si `usager principal` du parcours est coché, une variable `mainPerson`, de type `Person`, avec les variants `relations`, `household` (ménage) et `budget`;
- si `usager 1` est coché, une variable `person1`, de type `Person`, avec les variants `relations`, `household` (ménage) et `budget`;
- si `usager 2` est coché, une variable `person2`, de type `Person`, avec les variants `relations`, `household` (ménage) et `budget`;

### 0.2.5.6 Documents générés dans le dossier d'une personne : contexte "personne basique"

- une variable `person`, de type `Person`, avec les variants `relations`, `household` (ménage) et `budget`.

### 0.2.5.7 Documents générés dans le dossier d'une personne : contexte "personne avec un tiers"

Ce contexte permet de générer un courrier avec, en paramètre, un tiers.

Cela peut être utile pour, par exemple, générer un courrier vers un tiers déjà enregistré dans la base de données de Chill.

Les variables disponibles sont les suivantes :

- une variable `person`, de type `Person`, avec les variants `relations`, `household` (ménage) et `budget`.
- une variable `thirdParty`, de type `ThirdParty`;

## 0.2.6 Champs par objet

Les champs ci-dessous étendent les variables disponibles dans les documents.

Par exemple, le document généré à partir d'un parcours présente la variable `startDate` qui contient la date de début du parcours. Ce champ peut être exploité de la manière suivante :

```
${courseStartDateLong} // 15 janvier 2021
```

ou dans la forme courte:

```
${courseStartDateShort} // 15/01/2021
```

### 0.2.6.1 AccompanyingPeriod

- `id` (texte) : l'identifiant du parcours
- `type` (texte) : toujours égal à `accompanying_course`
- `isNull` (bool) renvoie `true` si est une entité nulle (toutes les variables sont vides);
- `closingDate` (Date) : date de clôture;
- `confidential` (bool) : `true` si confidentiel;
- `confidentialText` (texte) : "Confidentiel" si le parcours est confidentiel, texte vide sinon;
- `createdAt` (Date) : date de création;
- `createdBy` (User) : utilisateur ayant créé le parcours;
- `emergency` (bool) : `true` si parcours en urgence;
- `emergencyText` (texte) : "Urgent" si le parcours est urgent, texte vide sinon;
- `openingDate` (texte) : la date de confirmation du parcours;
- `origin` (AccompanyingPeriodOrigin) : l'origine du parcours;
- `originText` (texte) : Le titre de l'origine, directement utilisable dans le document;
- `participations` (liste de AccompanyingPeriodParticipation) : les participations du parcours, actuelles **et anciennes**;
- `currentParticipations` (liste de AccompanyingPeriodParticipation) : les participations actuelles du parcours;
- `requestorAnonymous` (bool) : `true` si le demandeur du parcours est anonyme;
- `hasRequestor` (bool) : `true` si le parcours a un demandeur (il peut être un tiers ou une personne);
- `requestorKind` (texte) : `person` si le demandeur est une personne, `thirdparty` si c'est un tiers;
- `requestorPerson` (Person) : demandeur, si le demandeur est une personne;
- `hasRequestorPerson` (bool) : `true` si le demandeur est une personne;
- `requestorThirdParty` (ThirdParty) : demandeur, si le demandeur est un tiers;
- `hasRequestorThirdParty` (bool) : `true` si le demandeur est un tiers;
- `resources` (liste de AccompanyingPeriodResource) : les ressources du parcours;

- `scopes` (liste de `Scope`) : services associés au parcours ;
- `scopesText` (texte) : titre des services associés au parcours, séparés par une virgule ;
- `socialIssues` (liste de `SocialIssues`) : problématiques sociales associées au parcours ;
- `socialIssuesText` (texte) : Liste des problématiques sociales, séparées par une virgule ;
- `intensity` (texte) : texte traduit de l'intensité : "ponctuel" ou "régulier" ;
- `step` (texte) : texte traduit de l'étape du parcours : "Brouillon", ou "En cours" ;
- `isClosed` (bool) : `true` si le parcours est fermé (la date de clotûre est renseignée) ;
- `closingMotiveText` (texte) : titre du motif de clotûre ;
- `closingMotive` (`ClosingMotive`) : motif de clotûre
- `ref` (`User`) : référent du parcours ;
- `hasRef` (bool) : `true` si un référent est désigné ;
- `hasLocation` (bool) : `true` si un parcours a une localisation ;
- `hasLocationPerson` (bool) : `true` si un parcours a une localisation auprès d'une personne.  
`false` si c'est une localisation temporaire ;
- `location` (Adresse) : l'adresse de localisation du parcours
- `locationPerson` (`Person`) : l'utilisateur qui localise le parcours (s'il y en a un, peut être vide) ;
- `administrativeLocation` (`Location`) : localisation administrative du parcours ;
- `hasAdministrativeLocation` (bool) : `true` si une localisation administrative est définie ;
- `works` (liste de `AccompanyingPeriodWork`) : liste des actions d'accompagnement créées dans le cadre de ce parcours ;
- `comments` (liste de `AccompanyingPeriodComment`) : liste des commentaires ;
- `pinnedComment` (`AccompanyingPeriodComment`) : commentaire épinglé ;

#### 0.2.6.2 AccompanyingPeriodComment

- `id` (int) : identifiant
- `content` (texte) : contenu du commentaire
- `createdAt` (`Date`) : date de création
- `creator` (`User`) : Créateur du commentaire

#### 0.2.6.3 AccompanyingPeriodOrigin

- `id` (int) : identifiant ;
- `label` (texte) : libellé de l'origine ;

#### 0.2.6.4 AccompanyingPeriodParticipation

- `id` (int) : identifiant
- `person` (Person) : usager associé au parcours;
- `startDate` (Date) : date de début;
- `endDate` (Date) : date de fin. Vide si la participation est toujours en cours;

#### 0.2.6.5 AccompanyingPeriodResource

- `id` (int) : identifiant;
- `person` (Person) : usager ressource (vide, avec le paramètre `isNull` à `true`, si la ressource est un tiers);
- `thirdParty` (ThirdParty) : tiers ressource (vide, avec le paramètre `isNull` à `true`, si la ressource est une personne);
- `comment` (texte) : commentaire associé à la personne ressource.

#### 0.2.6.6 Activity (échange)

- `id` (int) : identifiant;
- `activityType` (ActivityType) : Type d'échange
- `attende` (ActivityPresence) : Présence à l'échange
- `comment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `date` (Date) : Date de l'échange
- `durationTimeMinute` (int) : Durée de l'échange, en minutes
- `emergency` (bool) : True si en urgence
- `location` (Location) : Lieu de l'échange
- `reasons` (liste de ActivityReason) : Sujets de l'échange (pas utilisé en Vendée)
- `scope` (Scope) : Service (pas utilisé en Vendée)
- `sentReceived` (text) : sera `received` si reçu, et `sent` si envoyé;
- `socialActions` (liste de SocialAction) : liste d'actions sociales associées
- `socialIssues` (liste de SocialIssue) : liste de problématiques sociales associées
- `thirdParties` (liste de ThirdParty) : liste de tiers associés
- `travelTimeMinute` (int) : durée du trajet en minutes
- `user` (User) : l'utilisateur pour lequel l'échange a été créé;
- `users` (liste de Users) : les utilisateurs qui sont associés à l'échange.

**0.2.6.6.1 variant Light** Un variant `Light` est utilisé dans les listes d'`Activity`, comme par exemple pour le contexte "Liste des activités pour un parcours".

Ce variant comporte les mêmes attributs, avec les différences suivantes :

- le champ `user` n'est pas présent;
- le champ `scope` n'est pas présent;
- le champ `reasons` n'est pas présent;
- le champ `comment` est également présenté avec le variant `Light`.

### 0.2.6.7 ActivityType

Type d'activité.

- `id` (int) : identifiant;
- `name` (texte) : libellé

### 0.2.6.8 ActivityPresence

Présence à l'échange

- `id` (int) : identifiant;
- `name` (texte) : libellé

### 0.2.6.9 ClosingMotive

- `id` (int) : identifiant;
- `name` (texte) : libellé

### 0.2.6.10 Date

- `short` (texte) : la date au format dd/mm/yyyy, vide si la date est absente;
- `long` (texte) : la date au format jour mois années : 15 août 1980, vide si la date est absente;

### 0.2.6.11 Person

- `id` (texte) : Identifiant du dossier usager;
- `isNull` (bool) : `true` si le dossier usager est vide;
- `civility` (Civility) : Civilité

- `firstName` (texte) : Prénom
- `lastName` (texte) : Nom
- `altNames` (texte) : noms supplémentaire (nom de naissance, ...)
- `text` (texte) : représentation du nom, prénom, et noms alternatifs : prénom, nom, et noms alternatifs.
- `birthdate` (Date)
- `age` : (int) age de la personne;
- `deathdate` (Date)
- `gender` (texte) : genre (texte traduit);
- `maritalStatus` (texte) : représentation textuelle de l'état civil;
- `maritalStatusDate` (Date) : date à laquelle le statut de l'état civil a été mis à jour;
- `maritalStatusComment` (Comment) : Commentaire sur l'état civil (avec la date de la dernière mise à jour)
- `email` (texte) : adresse email
- `firstPhoneNumber` (texte) : soit le numéro de mobile s'il existe, soit le numéro fixe s'il existe, sinon vide;
- `fixPhoneNumber` (texte) : le numéro de téléphone fixe;
- `mobilePhoneNumber` (texte) : le numéro de téléphone portable;
- `nationality` (texte) : nationalité (nom du pays);
- `placeOfBirth` (texte) : le lieu de naissance
- `memo` (texte) : le mémo
- `numberOfChildren` (texte) : le nombre d'enfants
- `address` (Adresse) : l'adresse actuelle
- `resources` (PersonResource) : les personnes ressources, ajoutées depuis le dossier de l'utilisateur.

Au 23 mars 2022, ces champs ne sont plus disponibles :

- `numCaf` (texte) : le numéro CAF;
- `numSS` (texte) : le numéro de sécurité sociale;
- `numMSA` (texte) : le numéro MSA;
- `mutuelle` (texte) : le nom de la mutuelle
- `doctor` (ThirdParty) : le médecin traitant;
- `situationProf` (texte) : la situation professionnelle;
- `situationProfDate` (Date) : date de la situation professionnelle;

Par ailleurs, la nature de ces champs a changé :

- `statutLogement`
- `niveauEtude`

Ces champs sont disponibles, dès le 22 mars :

- `extras.majeur.isNull` (bool) : si un dossier “majeur” a été créé
- `extras.majeur.numeroCAF` (texte) : le numéro CAF;
- `extras.majeur.numeroSecuriteSociale` (texte) : le numéro de sécurité sociale;
- `extras.majeur.numMSA` (texte) : le numéro MSA;
- `extras.majeur.mutuelle` (texte) : le nom de la mutuelle
- `extras.majeur.medecinTraitant` (ThirdParty) : le médecin traitant;
- `extras.majeur.situationProfessionnelle` (texte) : la situation professionnelle;
- `extras.majeur.situationProfessionnelleComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.situationProfessionnelleStartDate` (Date)
- `extras.majeur.statutLogement` (StatutLogement) : statut au regard du logement;
- `extras.majeur.niveauEtude` (NiveauEtude) : le niveau d’ étude;
- `extras.majeur.identifiantPoleEmploi` (texte)
- `extras.majeur.accompagnementMissionLocale` (texte)
- `extras.majeur.gensDuVoyage` (bool) : true si la personne fait partie des gens du voyage
- `extras.majeur.titreDeSejour` (TitreDeSejour) : entité TitreDeSejour
- `extras.majeur.titreDeSejourComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.situationProfessionnelle` (SituationProfessionnelle) : entité SituationProfessionnelle
- `extras.majeur.aidantFamilial` (bool) : true si la personne fait partie des gens du voyage
- `extras.majeur.tempsDeTravail` (TempsDeTravail) : entité TempsDeTravail
- `extras.majeur.niveauEtude` (NiveauEtude) : entité NiveauEtude
- `extras.majeur.mobilite` (mMbilite) : entité Mobilite
- `extras.majeur.permanenceDeConduire` (bool) : true si la personne a un permis de conduire
- `extras.majeur.moyenDeTransport` (bool) : true si la personne a un moyen de transport
- `extras.majeur.infoSocioProComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.entourage` (Entourage) : entité Entourage
- `extras.majeur.entourageComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.numeroSecuriteSociale` (texte)

- `extras.majeur.caisseSecuriteSociale` (texte)
- `extras.majeur.pasDeMedecinTraitant` (bool) : true si la personne n'a pas de médecin traitant
- `extras.majeur.medecinTraitant` (Tiers) : entité ThirdParty
- `extras.majeur.medecinTraitantTexteLibre` (texte)
- `extras.majeur.mdph` (texte)
- `extras.majeur.mdphStartDate` (Date)
- `extras.majeur.mdphEndDate` (Date)
- `extras.majeur.mdphComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.aah` (texte)
- `extras.majeur.aahStartDate` (Date)
- `extras.majeur.aahEndDate` (Date)
- `extras.majeur.aahComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.rqth` (texte)
- `extras.majeur.rqthStartDate` (Date)
- `extras.majeur.rqthEndDate` (Date)
- `extras.majeur.rqthComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.pcdh` (texte)
- `extras.majeur.pcdhStartDate` (Date)
- `extras.majeur.pcdhEndDate` (Date)
- `extras.majeur.pcdhComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.adpa` (texte)
- `extras.majeur.adpaStartDate` (Date)
- `extras.majeur.adpaEndDate` (Date)
- `extras.majeur.adpaComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.gir` (texte)
- `extras.majeur.girStartDate` (Date)
- `extras.majeur.girEndDate` (Date)
- `extras.majeur.girComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.infoMedicaleComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.statutLogement` (StatutLogement) : entité StatutLogement



- `extras.majeur.statutLogementComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.caisseDeRetraitePrincipale` (texte)
- `extras.majeur.caisseDeRetraiteComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.majeur.caisseDeRetraiteComplementaire` (liste de texte) : les caisses de retraite complémentaires, en liste de texte;
- `extras.majeur.caissesDeRetraiteComplementairesText` (texte) : les caisses de retraite complémentaires, dans un seul champ texte, séparé par des virgules; En outre, si la personne est mineure, on a accès aux champs suivants :
- `extras.majeur.isNull` (bool) : si un dossier “mineur” a été créé
- `extras.mineur.enfantConfie` (bool) : true si enfant confié
- `extras.mineur.dossierASE` (texte)
- `extras.mineur.autoriteParentaleFr` (texte)
- `extras.mineur.autoriteParentaleComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.mineur.identificationAutoriteParentale1Fr` (texte)
- `extras.mineur.identificationAutoriteParentale2Fr` (texte)
- `extras.mineur.autoriteParentale1Date` (Date)
- `extras.mineur.autoriteParentale2Date` (Date)
- `extras.mineur.retraitAutoriteParentale` (bool) : true si retraitAutoriteParentale
- `extras.mineur.retraitAutoriteParentaleDate` (Date)
- `extras.mineur.retraitAutoriteParentaleComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.mineur.delegationBool` (bool) : true si délégation
- `extras.mineur.delegationDate` (Date)
- `extras.mineur.delegationComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.mineur.delegationThirdParty` (tiers) : entité ThirdParty
- `extras.mineur.delegationPerson` (person) : entité Person
- `extras.mineur.tuteurBool` (bool) : TRUE si tuteur
- `extras.mineur.tuteurThirdParty` (tiers) : entité ThirdParty
- `extras.mineur.administrateurAdhoc` (bool) : true si administrateurAdhoc
- `extras.mineur.administrateurAdhocDate` (Date)
- `extras.mineur.administrateurAdhocComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.mineur.pupilleEtat` (bool) : true si pupilleEtat

- `extras.mineur.pupilleEtatDate` (Date)
- `extras.mineur.pupilleComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.mineur.tuteurPerson` (person) : entité Person
- `extras.mineur.enfantEmancipe` (bool) : true si enfantEmancipe
- `extras.mineur.enfantEmancipeDate` (Date)
- `extras.mineur.gardeAlternee` (bool)
- `extras.mineur.decisionJAF` (bool) : true si decisionJAF
- `extras.mineur.decisionJAFDate` (Date)
- `extras.mineur.decisionJAFEnfantConfie` (bool) : true si decisionJAFEnfantConfie
- `extras.mineur.decisionJAFDateEnfantConfie` (Date)
- `extras.mineur.fratrieComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.mineur.droitDeVisiteComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.mineur.infoFamilialeComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.mineur.residenceMineur` (texte)
- `extras.mineur.derniereVisiteMedicaleDate` (Date)
- `extras.mineur.visiteMedicaleText` (texte)
- `extras.mineur.prochaineVisiteMedicaleDate` (Date)
- `extras.mineur.autresRDV` (texte)
- `extras.mineur.infoMedicaleComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.mineur.scolarise` (bool) : true si scolarisé
- `extras.mineur.derniereClasseScolaire` (texte)
- `extras.mineur.etablissementScolaire` (texte)
- `extras.mineur.classeScolaire` (texte)
- `extras.mineur.alternance` (bool) : true si alternance
- `extras.mineur.alternanceEntreprise` (texte)
- `extras.mineur.enseignantReferant` (texte)
- `extras.mineur.scolariteComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.mineur.mineurComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.mineur.autresIntervenantsComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)

- `extras.mineur.dispositifs` (texte)
- `extras.mineur.dossierMDPH` (texte)
- `extras.mineur.notificationMDPH` (texte)
- `extras.mineur.notificationMDPHStartDate` (Date)
- `extras.mineur.notificationMDPHEndDate` (Date)
- `extras.mineur.dispositifsComment` (Comment) : Commentaire (avec la date de la dernière mise à jour)
- `extras.mineur.adresseDeRelais` (adresse) : liste d'adresses

#### **0.2.6.11.1 variant household**

- `household` (Household) : le ménage actuel de l'utilisateur;

#### **0.2.6.11.2 variant relations**

- `relations` (liste de Relationship) : les relations (filiiations) de l'utilisateur;

**0.2.6.11.3 variant budget** Le budget des usagers et de leur ménage est accessible de manière synthétisée : la somme globale de chaque "ligne" de budget est effectuée et disponible pour l'inclusion.

Chaque "ligne" dépend de la configuration de votre instance.

Toutes les lignes de budget possibles sont toujours présentes. Il est possible d'ajouter une condition dans le document pour n'afficher que les lignes ayant une somme supérieure à zéro.

**0.2.6.11.3.1 Attributs pour une Ligne de budget** Pour chaque ligne, les informations suivantes sont disponibles :

- `sum` (nombre décimal) : la somme de toutes les lignes du budget du même type;
- `label` (texte) : le libellé de cette ligne dans l'interface;
- `comment` (texte) : les commentaires des lignes du budget de même type. Tous les commentaires sont concaténés les uns à la suite des autres, et séparés par le caractère |.

**0.2.6.11.3.2 Méthode de globalisation de chaque ligne** Pour chaque ligne de budget, la somme de toutes les lignes de budget du même type est disponible. Ainsi, si une entrée dans les ressources permet de saisir une charge "frais de communication téléphonique" :

- pour un usager, si plusieurs lignes `frais de communication` sont enregistrées, alors c'est la somme de celles-ci qui apparaîtra dans la génération de documents;

- pour un ménage, alors ce sera la somme de tous les frais de communication de chaque membre du ménage **au moment de la génération du document** qui apparaîtra, mais également la somme des frais de communications enregistrés directement dans le dossier du ménage.

On verra donc apparaître non seulement les frais payés par chaque usager (pour leur abonnement individuel au téléphone mobile, par exemple), mais également la somme payée par le ménage (pour le téléphone fixe ou l'accès à internet).

Enfin, seules les lignes *actives* **au moment de la génération du document** sont prises en compte. Ainsi, par exemple, si un CDI a expiré la veille de la génération, son montant ne sera pas pris en compte.

Le calcul du caractère actif d'une ligne est effectué comme suit :

- la date de début est antérieure à la date de génération du document;
- la date de fin est vide, ou postérieure à la date de génération du document.

**0.2.6.11.3.3 Attributs supplémentaire avec le variant budget :** Les attributs suivants sont disponibles, pour chaque objet Person qui a un variant budget :

- `budget.person.resources.<clé du type de ligne de budget>` (Ligne de budget) : une ligne de budget pour la ressource dont la clé est indiquée par `<clé du type de ligne de budget>`. La liste des clés est à obtenir auprès de l'administrateur technique. Seules les lignes du budget associées au dossier de l'utilisateur sont présentes ici;
- `budget.person.charges.<clé du type de ligne de budget>` (Ligne de budget) : une ligne de budget pour la charge dont la clé est indiquée par `<clé du type de ligne de budget>`. La liste des clés est à obtenir auprès de l'administrateur technique. Seules les lignes du budget associées au dossier de l'utilisateur sont présentes ici;
- `budget.household.resources.<clé du type de ligne de budget>` (Ligne de budget) : une ligne de budget pour la ressource dont la clé est indiquée par `<clé du type de ligne de budget>`. La liste des clés est à obtenir auprès de l'administrateur technique. Les lignes de budget de tous les usagers actuels du ménages sont globalisées, ainsi que les lignes de budget associées directement au dossier ménage;
- `budget.household.charges.<clé du type de ligne de budget>` (Ligne de budget) : une ligne de budget pour la charge dont la clé est indiquée par `<clé du type de ligne de budget>`. La liste des clés est à obtenir auprès de l'administrateur technique. Les lignes de budget de tous les usagers actuels du ménages sont globalisées, ainsi que les lignes de budget associées directement au dossier ménage;

#### 0.2.6.11.4 Exemple d'utilisation

Nom de la personne: `${personLastname}` // Dupuis  
Prénom de la personne: `${personFirstname}` // Gérard  
Date de naissance: `${personBirthdateLong}` // le 7 juillet 1974  
Nom du père: `${personFatherLastName} ${personFatherFirstname}` // Dupuis Marcel  
Téléphone du père: `${personFatherFirstphonenum}` // 01 23 45 67 89

#### **0.2.6.12 PersonResource (Ressources associée à l'utilisateur)**

Pour rappel, les ressources peuvent être :

- des tiers;
- des usagers;
- ou un champ de texte libre.
- comment (Comment)
- freeText (texte) : commentaire, quand la personne ressources un texte libre
- id (int)
- kind (PersonResourceKind) : le type de personne ressources (voisin, etc.)
- thirdParty (ThirdParty) : le tiers, quand la ressources est liée à un tiers
- person (Person) : l'utilisateur, quand la ressources liée à un autre usager
- resourceKind (texte) : un discriminateur qui permet de vérifier si la ressources est lié à un usager (alors égal à person, un tiers (thirdparty) ou un texte libre (freetext)). A utiliser dans les conditions dans les documents.

#### **0.2.6.13 PersonResourceKind (Type de personne ressource)**

- id (int)
- title (texte)

#### **0.2.6.14 StatutLogement**

- id (int) : identifiant (utile pour des comparaisons);
- active (bool) : si le statut est encore utilisé;
- name (texte) : titre

#### **0.2.6.15 NiveauEtude**

- id (int) : identifiant (utile pour des comparaisons);

- `active` (bool) : si le niveau d'étude est encore utilisé;
- `name` (texte) : titre

#### **0.2.6.16 SituationProfessionelle**

- `id` (int) : identifiant (utile pour des comparaisons);
- `active` (bool) : si la situation professionnelle est encore utilisée
- `name` (texte) : titre

#### **0.2.6.17 Civility**

- `abbreviation` (texte) : abbréviation;
- `id` (int) : identifiant (utile pour des comparaisons);
- `name` (texte) : titre, label de la civilité;

#### **0.2.6.18 Entourage**

- `person` (person) : entité Person
- `thirdParty` (tiers) : entité ThirdParty
- `text` (texte)
- `comment` (comment) : Commentaire (avec la date de la dernire mise à jour)
- `aidantProche` (bool) : true si aidantProche

#### **0.2.6.19 Household**

- `id` (int) : l'identifiant du ménage;
- `members` (liste de HouseholdMember) : liste des membres du ménages;
- `waitingForBirth` (bool) : true si une naissance est attendue;
- `waitingForBirthDate` (Date) : date de la naissance attendue;

#### **0.2.6.20 HouseholdMember**

- `comment` (texte) : texte du commentaire;
- `id` (int) : identifiant de la participation au ménage;
- `endDate` (Date) : date de fin de participation au ménage;
- `holder` (bool) : true si le participant est titulaire;

- `person` (Person) : l'utilisateur qui est membre du ménage;
- `position` (HouseholdPosition) : la position du membre dans le ménage;
- `startDate` (Date) : date de début de participation au ménage;

#### 0.2.6.21 HouseholdPosition

- `id` (int) : identifiant de la position;
- `label` (texte) : label de la position dans le ménage;

#### 0.2.6.22 Mobilité, NiveauEtude, SituationProfessionnelle, StatutLogement, TempsDeTravail, TitreDeSejour

- `active` (bool) : true si active
- `name` (texte)

#### 0.2.6.23 ThirdParty

- `id` (int) : identifiant du tiers;
- `acronym` (text) : acronyme;
- `address` (Adress) : adresse du tiers
- `categories` (liste de ThirdPartyCategory) : liste des catégories;
- `civility` (Civility) : civilité
- `contactDataAnonymous` (bool) : true si le tiers est anonyme;
- `email` (texte) : email;
- `name` (texte) : nom du tiers;
- `firstname` (texte) : Prénom du tiers;
- `parent` (ThirdParty) : tiers parent (pour les contacts)
- `profession` (ThirdPartyProfessionion)
- `telephone` (texte) : numéro de téléphone du tiers
- `kind` (texte) : permet de distinguer les personnes morales, les contacts et les personnes physiques (voir ci-après)
- `child` (bool) vaut TRUE s'il existe un parent (contact), false sinon;
- `parent` (ThirdParty) : s'il s'agit d'un contact d'une personne morale, contient la fiche d'une personne morale

`kind` vaut :

- `company` lorsqu'il s'agit d'une personne morale

- `child` lorsqu'il s'agit d'un contact d'une personne morale
- `contact` lorsqu'il s'agit d'une personne **physique** (à ne pas confondre avec un contact d'une personne morale)

#### 0.2.6.24 **ThirdPartyCategory (catégorie de tiers)**

- `id` (int) : identifiant
- `name` (texte)

#### 0.2.6.25 **ThirdPartyProfession (profession du tiers)**

- `id` (int) : identifiant
- `name` (texte)

#### 0.2.6.26 **Adresse**

- `address_id` (int) : l'identifiant de l'adresse;
- `text` (texte) : une chaîne de caractère représentant l'adresse complète;
- `street` (texte) : le nom de la rue (exemple : "RUE DES ÉGLANTIERS");
- `streetNumber` (texte) : le numéro de police;
- `postcode` (PostCode) : le code postal;
- `country` (Country) : le pays de l'adresse;
- `floor` (texte) : l'étage;
- `corridor` (texte) : le couloir;
- `flat` (texte) : l'appartement;
- `buildingName` (texte) : le nom du bâtiment / résidence;
- `distribution` (texte) : service particulier de distribution;
- `extra` (texte) : champs extras
- `validFrom` (Date) : date de début de validité;
- `validTo` (Date) : date de fin de validité (vide si toujours valide);
- `lines` (liste de texte) : les lignes de l'adresses, comme elles devraient être formatées dans une adresses

Il est possible de représenter l'adresse au format postal en effectuant une boucle sur les lignes. Par exemple, pour l'adresse d'une personne (chaque ligne est un "renvoi > substituant" dans l'exemple ci-dessous) :

```
for each="line in v.person.address.lines
```



line  
/for

#### 0.2.6.27 Postcode

- id (int) : identifiant
- name (texte) : localité
- code (texte) : code postal

#### 0.2.6.28 Country

- id (int) : identifiant
- name (texte) : nom du pays
- code (texte) : deux lettres du code ISO pays

#### 0.2.6.29 User

- id (int) : identifiant
- type (texte) : vaut toujours user
- username (texte) : le login
- email (texte) : l'email de l'utilisateur
- text (texte) : le libellé complet de l'utilisateur (avec son métier et son service entre parenthèse);
- label (texte) : le libellé de l'utilisateur, tel qu'il est enregistré par l'administrateur fonctionnel
- main\_scope (Scope) : service principal
- user\_job (UserJob) : métier principal
- current\_location (Location) : lieu de l'utilisateur, celui actuellement choisi par l'utilisateur par le menu "utilisateur";
- main\_location (Location) : localisation de l'utilisateur définie par l'administrateur fonctionnel (parfois appelé "résidence administrative"). L'utilisateur ne peut pas la modifier lui-même;
- civility (Civility) : la civilité de l'utilisateur;

#### 0.2.6.30 UserJob

- id (int) : identifiant
- label (texte) : nom du métier

**0.2.6.31 Location (lieu)**

- `id` (int) : identifiant;
- `name` (texte) : nom;
- `address` (Address) : adresse du lieu;
- `phoneNumber1` (texte) : numéro de téléphone;
- `phoneNumber2` (texte) : numéro de téléphone;
- `email` (email) : email
- `locationType` (LocationType) : le type de lieu

**0.2.6.32 LocationType (type de lieu)**

- `id` (int) : identifiant;
- `title` (texte) : titre du type de lieu;

**0.2.6.33 Social Issue (problématique sociale)**

- `name` : le nom de la problématique seule
- `text` : le nom de la problématique et des problématiques parentes

**0.2.6.34 Relationship (relations (filiations) entre usagers)**

Pour faciliter l'usager, les relations sont toujours représentée à partir d'un usager d'origine.

Les variables présentent le nom de la relation (`text`), et la personne avec qui la relation est formée (`opposite`).

- `id` (int) : identifiant;
- `fromPerson` (Person) : personne d'origine;
- `toPerson` (Person) : personne de destination;
- `text` (texte) : le titre ou le titre inversé de la `Relation`, selon l'usager d'origine;
- `opposite` (Person) : l'usager avec qui la relation est formée;
- `relationId` (int) : identifiant du lien de filiation;

**Note** : la différence entre `fromPerson` et `toPerson` est arbitraire.

**0.2.6.35 AccompanyingPeriodWorkEvaluation (évaluation dans une action d'accompagnement)**

- `id` (texte) : l'identifiant de l'évaluation;

- `type` (texte) : le libellé de l'évaluation;
- `startDate` (Date) : la date de début de l'évaluation;
- `endDate` (Date) : la date de fin de l'évaluation;
- `maxDate` (Date) : la date d'échéance de l'évaluation;
- `comment` (texte) : le commentaire de l'évaluation;
- `createdBy` (User) : le créateur de l'évaluation;
- `createdAt` (Date) : la date de création de l'évaluation;
- `evaluation` (Evaluation) : le type d'évaluation;

#### **0.2.6.36 Evaluation (type d'évaluation)**

- `id` (texte) : l'identifiant du type d'évaluation;
- `title` (texte) : le titre du type d'évaluation;

#### **0.2.6.37 AccompanyingPeriodWork (Action d'accompagnement)**

- `id` (texte) : l'identifiant de l'action;
- `note` (texte) : la note;
- `persons` (liste de Person) : liste des usagers concernés par l'action
- `results` (liste de Result) : liste des résultats directement associé au parcours (résultats sans objectifs)
- `socialAction` (SocialAction) : type d'action sociale;
- `startDate` (Date) : la date de début de l'action (vide si inexistant)
- `endDate` (Date) : la date de fin de l'action (vide si inexistant)
- `thirdParties` (liste de ThirdParty) : les tiers associés à l'action comme tiers intervenants;
- `updatedBy` (User) : utilisateur ayant fait la mise à jour;
- `updatedAt` (Date) : date de mise à jour;
- `handlingThirdParty` (ThirdParty) : tiers traitant;
- `goals` (AccompanyingPeriodWorkGoal) : objectifs
- `createdBy` (Date) : date de mise à jour;
- `createdAt` (User) : utilisateur ayant fait la mise à jour;
- `createdAutomatically` (bool) : `true` si l'action a été créée automatiquement (par exemple, en créant un échange);
- `evaluations` (AccompanyingPeriodWorkEvaluation) : liste des évaluations générées;
- `referrers` (liste de User) : liste des agents traitants;

**0.2.6.38 SocialAction (type d'action d'accompagnement)**

- `id` (texte) : identifiant
- `text` (texte) : texte, avec les parents inclus, séparés par un `>` ;
- `title` : titre du type d'action, sans les parents ;

**0.2.6.39 Result (résultat d'une action)**

- `id` (texte) : identifiant
- `title` (texte) : le titre du résultat

**0.2.6.40 Goal (objectif d'une action)**

- `id` (texte) : identifiant
- `title` (texte) : le titre du résultat

**0.2.6.41 AccompanyingPeriodWorkGoal (objectifs et résultats d'une action d'accompagnement)**

- `id` (texte) : identifiant
- `goal` (Goal) : objectif d'une action
- `results` (liste de Result) : liste des résultats

**0.2.6.42 Comment**

Certains champs commentaire enregistrent également l'utilisateur qui a effectué la dernière mise à jour, et la date de celle-ci, en plus du commentaire en tant que tel. Ces champs ont les attributs suivants :

- `comment` (texte) : le commentaire en tant que tel ;
- `date` (Date) : la date de la dernière modification ;
- `user` (User) : l'utilisateur qui a effectué la dernière modification ;

**0.2.6.42.1 Variant Light** Dans le variant `Light`,

- le champ `date` n'est pas présent ;
- le champ `user` n'est pas présent.

Seul le champ `comment` est donc disponible.