Maarch Digital Flow Documentation

La gestion de contenu intégrée

Maarch

Copyright © 2026 Maarch

Table of contents

1. Version 5.0 - Documentation en ligne 3
2. Documentation intégrateur 5
   2.1 Documentation intégrateur 5
   2.2 Guides How To 6
   2.3 Références 0
   2.4 Tutoriels 0
3. Documentation utilisateur 0
   3.1 Documentation utilisateur 0
   3.2 Comprendre 0
   3.3 Guides How To 0
   3.4 Références 0
   3.5 Tutoriels 0

Copyright © 2026 Maarch

1. Version 5.0 - Documentation en ligne

Maarch Digital Flow est un ensemble de **services de gestion des contenus**. Il permet aux organisations de centraliser et rationnaliser la gestion et le traitement des contenus d'information produits ou reçus dans le cadre de leurs activités, tout en assurant le respect des obligations réglementaires.

Maarch Digital Flow

- centralise la collecte,
- assure le stockage,
- permet l'organisation, le partage, la collaboration, la diffusion
- appuie les politiques de gestion

de l'information

- produite ou reçue,
- quelle que soit sa forme,
- quel que soit son mode de production ou son canal de réception

tout en

- enregistrant les éléments de preuve de l'activité,
- garantissant le respect des obligations réglementaires et
- appuyant les politiques qualité.

Ses fonctions couvrent un large périmètre allant de la capture multicanale à la diffusion, en passant par la structuration en dossiers, la complétude, le classement, le stockage sécurisé, la collaboration ou encore l'édition des contenus.

Les contenus gérés sont aussi très variés, depuis les documents numériques jusqu'aux dossiers organiques des services, les groupes, contacts, événements de calendrier et messages d'échange entre les acteurs.

Cette documentation est structurée en 2 volumes principaux :

- La documentation utilisateur

Guide de visite, Guide de l'utilisateur, Guide de l'administrateur

- La documentation intégrateur

Paramétrage avancé de la solution

Copyright © 2026 Maarch

Chaque volume est lui-même structuré en 4 chapitres principaux :

- Cibler un objectif précis via les **guides How-to**
- Apprendre via les **Tutoriels** : pas à pas de prise en main
- Trouver les informations dans les **Références**
- **Comprendre** le fonctionnement général du logiciel

2 février 2026

9 juin 2021

Copyright © 2026 Maarch

2. Documentation intégrateur

2.1 Documentation intégrateur

Ce volume est organisé en 4 chapitres principaux :

- Cibler un objectif précis via les **guides How-to**
- Apprendre via les **Tutoriels**
- Trouver les informations dans les **Références**
- **Comprendre** le fonctionnement général du logiciel

1 décembre 2025

9 juin 2021

Copyright © 2026 Maarch

2.2 Guides How To

2.2.1 Personnalisation de la recherche

Paramétrage des attributs affichés dans une liste de résultats

INTRODUCTION

Cette documentation explique comment configurer les attributs affichés dans une liste de résultats en utilisant l'interface `SearchResultAttributeDisplayedInterface`. Elle décrit les différents paramètres disponibles et leurs fonctionnalités afin de personnaliser l'affichage des données.

INTERFACE DE PARAMÉTRAGE

L'interface de configuration permet de définir les attributs affichés ainsi que leur présentation.

```
export interface SearchResultAttributeDisplayedInterface {
    identifier?: string;
    title: string;
    expression: string;
    type?: string;
    view?: SubSetListModelViewInterface;
}
```

Propriétés

Propriété	Type	Obligatoire	Description
`identifier`	string	Non	Identifiant unique de l'attribut.
`title`	string	Oui	Label affiché pour l'attribut.
`expression`	string	Oui	Expression permettant de récupérer la valeur de l'attribut.
`type`	string	Non	Type de donnée de l'attribut.
`view`	object	Non	Configuration de l'affichage (voir détails ci-dessous).

CONFIGURATION DE L'AFFICHAGE (`view`)

La propriété `view` permet de personnaliser l'affichage d'un attribut, en définissant si le label doit être affiché, si une icône doit être ajoutée et si une mise en forme est appliquée.

Paramètres de `view`

Propriété	Type	Obligatoire	Description
`showTitle`	boolean	Non	Détermine si le titre est affiché (sinon visible au survol).
`icon`	string	Non	Classe CSS d'une icône affichée avant la valeur.
`format`	object	Non	Format de la valeur affichée (voir ci-dessous).

Configuration du `format`

Le `format` permet d'appliquer un formatage à la valeur affichée. Deux types de formatage sont disponibles :

1. **Format de date** : Permet d'afficher une date sous un format précis.

```
"format": {
  "id": "date",
  "value": "dd/MM/YYYY HH:mm"
}
```

2. **Traduction** : Permet de convertir une valeur en un libellé via un dictionnaire de traduction.

Copyright © 2026 Maarch

```
"format": {
  "id": "toTranslate"
}
```

EXEMPLES DE CONFIGURATION

Exemple 1 : Attribut affiché sans titre (visible au survol)

```
{
  "title": "Type:",
  "expression": "subType.displayName",
  "view": {
    "showTitle": false
  }
}
```

Exemple 2 : Attribut avec traduction

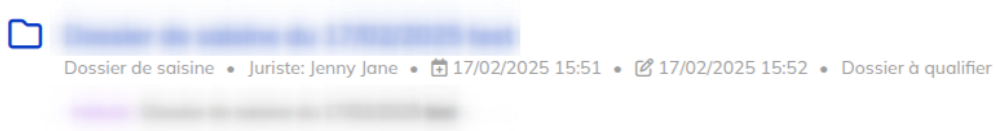
```
{
  "title": "Signalé:",
  "expression": "attributes.attributesModel_binary_content.signale",
  "view": {
    "format": {
      "id": "toTranslate"
    }
  }
}
```

Exemple 3 : Attribut avec icône et format de date

```
{
  "title": "Créé le",
  "expression": "createdAt",
  "view": {
    "icon": "fa-regular fa-calendar-plus",
    "showTitle": false,
    "format": {
      "id": "date",
      "value": "dd/MM/YYYY HH:mm"
    }
  }
}
```

Exemple 4 : Attribut tronqué avec données au delà de 50 caractères

```
{
  "title": "Long texte",
  "expression": "displayName",
  "view": {
    "format": {
      "id": "truncate",
      "value": 50
    }
  }
}
```



CONCLUSION

Cette configuration permet d'afficher dynamiquement les attributs d'une liste de résultats avec des options avancées de personnalisation, notamment le masquage du titre, l'ajout d'icônes et le formatage des valeurs. En utilisant ces paramètres, il est possible d'adapter l'affichage aux besoins des utilisateurs tout en assurant une présentation cohérente et ergonomique.

Historique des versions

ir l'historique

Version	Changements
`4.0`	Introduction des expressions multiples dans les vues avec mécanisme de fallback.

Copyright © 2026 Maarch

10 novembre 2025

13 mars 2025

Copyright © 2026 Maarch

Modèle de Filtres de Recherche

MODES D'UTILISATION

Il existe deux modes d'utilisation des filtres :

Mode 1 : Utilisation Littérale du Filtre sur un Attribut

Format de requête consommé :

```
/search?entityType[]=binaryContent...
```

Mode 2 : Utilisation via le Modèle lui-même

Format de requête consommé :

```
/search?template.my_filter_model.identifier[]=value...
```

ATTRIBUTS COMMUNS

Attribut	Description
`type`	Type du champ (`string` / `array` / `date`)
`title`	Libellé du champ
`position`	Ordre d'affichage
`readOnly`	Champ en lecture seule
`default`	Valeur définie par défaut lors de initialisation du champ
`paths` (optionnel)	Permet de spécifier plusieurs identifiants cibles (active le mode 2) → Peut cibler un attribut système ou un attribut d'un modèle spécifique
`view.type`	Type d'entrée pour l'affichage
`view.hidden`	Champs "caché" ou non

TYPES DE CHAMPS DISPONIBLES

1. Champ Texte

```
{
  "identifier": {
    "type": "string",
    "view": {
      "type": "input-text",
      "size": "small-block"
    },
    "paths": ["displayName"],
    "title": "Nom d'affichage",
    "position": 1
  }
}
```

Avec valeur(s) par défaut :

```
{
  "identifier": {
    "type": "string",
    "default": "LY2025",
    "view": {
      "type": "input-text",
      "size": "small-block"
    },
    "paths": ["displayName"],
    "title": "Nom d'affichage",
    "position": 1
  }
}
```


Copyright © 2026 Maarch

Mot clé @me

Le mot-clé `@me` peut être utilisé en valeur par défaut. Tous les champs décrits ici sont disponibles.

Exemple :

```
{
  "identifier": {
    "type": "string",
    "default": "@me.displayName",
    "view": {
      "type": "input-text",
      "size": "small-block"
    },
    "paths": ["displayName"],
    "title": "Nom d'affichage",
    "position": 1
  }
}
```

---

2. Plage de Date

```
{
  "lifecycle.publishedAt": {
    "type": "date",
    "view": {
      "type": "input-date"
    },
    "title": "Période de mise en ligne",
    "position": 6
  }
}
```

Avec valeur(s) par défaut :

```
{
  "lifecycle.publishedAt": {
    "default": {
      "start": "2025-01-01"
      "end": "2025-01-15",
    },
    "type": "date",
    "view": {
      "type": "input-date"
    },
    "title": "Période de mise en ligne",
    "position": 6
  }
}
```

`start ou` end` peuvent être définie en valeurs par défaut

---

3. Liste à Choix Multiple

```
{
  "type_activite": {
    "type": "array",
    "enum": ["Contrôle", "Conseil", "Contentieux"],
    "view": {
      "type": "input-select-multiple"
    },
    "paths": ["attributes.saisine.contenu_saisine.type_activite"],
    "title": "Type d'activité",
    "position": 3
  }
}
```

Avec valeur(s) par défaut :

```
{
  "type_activite": {
    "type": "array",
    "default": ["Conseil"],
    "enum": ["Contrôle", "Conseil", "Contentieux"],
    "view": {
      "type": "input-select-multiple"
    },
    "paths": ["attributes.saisine.contenu_saisine.type_activite"],
    "title": "Type d'activité",
```


Copyright © 2026 Maarch

```
    "position": 3
  }
}
```

CHAMPS SPÉCIAUX (CHARGEMENT DYNAMIQUE)

```
{
  "entityType": {
    "type": "array",
    "enum": [ "entry","binaryContent" ],
    "enumNames": [
      "Dossier",
      "Document"
    ],
    "view": {
      "type": "input-select-multiple"
    },
    "title": "Type",
    "position": 1
  }
}
```

```
{
  "subType.identifier": {
    "enum": [],
    "type": "array",
    "view": {
      "type": "input-select-multiple"
    },
    "title": "Sous-type",
    "position": 2
  }
}
```

1 avril 2025

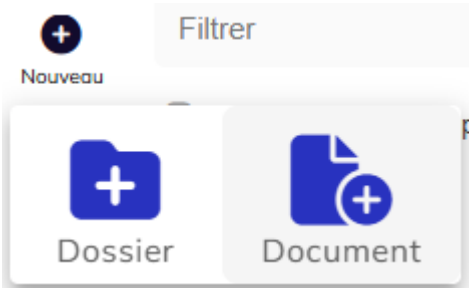
13 mars 2025

Copyright © 2026 Maarch

2.2.2 Personnalisation des interfaces

Désactiver dans l'interface le menu de création de document hors contexte de relation

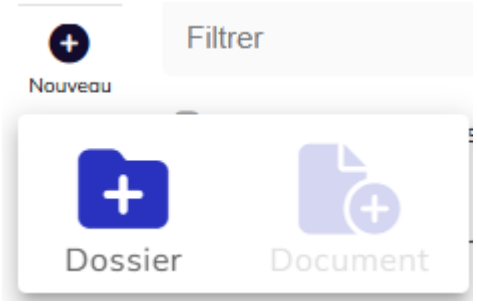
Il peut être nécessaire de désactiver pour certains utilisateurs, le menu accessible dans l'interface de création de nouveau document et ne permettre cette création qui via la mécanique des relations.



Une propriété des groupes est paramétrable pour cela :

Propriété	Type	Fonction
`canAddDocumentFromMenu`	booléen	désactive le menu principal d'ajout de nouveau document

```
{
    "settings": {"canAddDocumentFromMenu": false}
}
```



Non exposée dans l'API, cette modification n'est possible qui via une requête SQL.

3 avril 2025

3 avril 2025

Copyright © 2026 Maarch

Comment modifier un template

IDENTIFIER LE TEMPLATE À MODIFIER

Si vous ne connaissez pas l'id du Template que vous souhaitez modifier, repérer le Template à modifier en appelant la route `GET /templates`

MODIFICATION DU TEMPLATE

Pour modifier un Template, il faut avoir le droit `update`

Envoyer une requête `PUT /templates/{id}` avec en body :

```
{
  "templateFormat": "twig",
  "template": "<p>Le dossier sous la référence citée en objet a été mis en attente d'instruction par nos services pour le motif suivant <br>{{ lastComment }}
<br>--<br>MDF</p>",
  "id": "e8a32eb4-0503-4535-aef2-006dc0e626a8",
  "displayName": "Template test pour la doc",
  "description": "Template Twig à utiliser pour l'envoi de mails",
  "identifier": "template_test",
  "attributes": {
    "subject": "Objet du template modifié"
  }
}
```

Il n'y a que les champs `templateFormat`, `template`, `displayName`, `description` et `attributes` qui sont modifiables.

- `template` contient le contenu du Template
- `attributes.subject` contient l'objet du mail envoyé, si le Template est utilisé pour l'envoi de mail

VARIABLES UTILISABLES DANS LE CONTENU D'UN TEMPLATE

Template de mail utilisé dans une Task

Champs disponibles :

Champ	Type	Description	Exempla	Type de Task
lastComment	string	Contenu de la dernière annotation publique ajoutée	"Une annotation"	`acknowledgeStandBy` et `acknowledgeAdvice`
entity	object	Entité sur laquelle la Task est executé. Tous les champs de l'entité sont accessible (ex: `entity.displayName`)	Objet Entry, objet User	`acknowledgeStandBy`, `acknowledgeAdvice` et `acknowledgeReceipt`
recipients	string[]	Liste des destinataires de l'email	`["bblier@maarch.com"]`	`acknowledgeStandBy` et `acknowledgeAdvice`

Copyright © 2026 Maarch

Templates initialisation / reset du mot de passe utilisateur

Champs disponibles :

Champ	Type	Description	Exempla
firstName	string	Prénom de l'utilisateur. Utilise le champ `attributes.ModelAgent.prenom`	`"Bernard"`
lastName	string	Nom de l'utilisateur. Utilise le champ `attributes.ModelAgent.nom`	`"Blier"`
hash	string	Token d'authentification nécessaire pour appeler l'API de changement de mot de passe	
expires	int	Durée de validité du token, en secondes. La durée est paramétrable avec la variable d'environnement `RESET_PASSWORD_TOKEN_LIFETIME`	
duration	string	Durée formaté en secondes ou minutes	`"60 minutes"`
frontUrl	string	Url de l'application front. À utiliser pour formatter un lien vers la page front.	`"https://mdf.maarch.fr/#/"`

Template PDF

Les templates utilisés pour la génération de PDF sont au format HTML Twig. Le code HTML sera converti en PDF par le module d'export.

Tous les champs de l'entité exportée sont disponibles, par exemple pour avoir le displayName il suffit d'utiliser `displayName`

Le SubType de l'entité est accessible via le champ `type` (ex: `type.displayName`)

13 mars 2025

13 mars 2025

Copyright © 2026 Maarch

Comment personnaliser les libellés de l'application

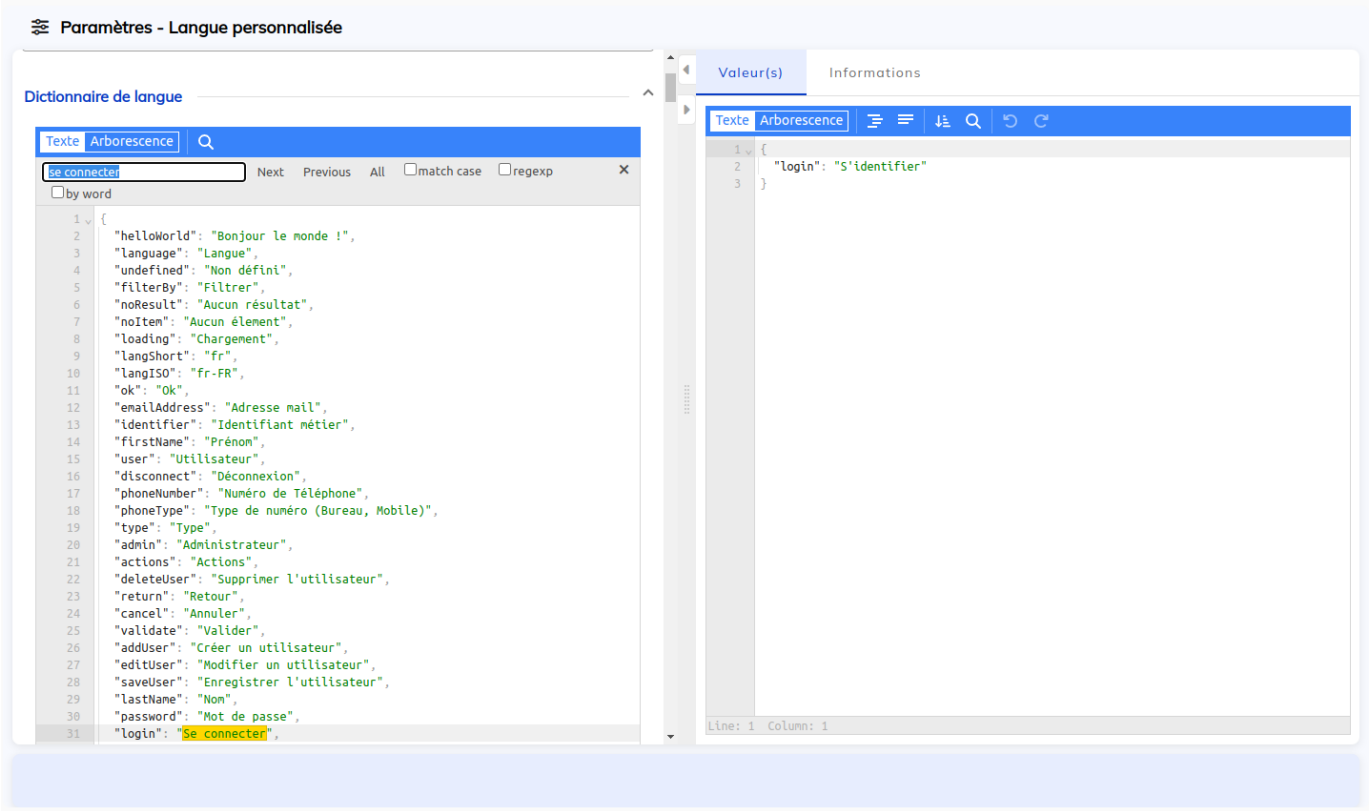
Maarch Digital Flow utilise un système de dictionnaire de langue pour la traduction des interfaces de l'application en format `.json` (situé dans `assets/lang/fr.json`)

Celles-ci peuvent être surchargé pour les re-définir :

- Accéder à l'administration des Paramètres :
- Ouvrir le paramètre `custom_lang_fr`
- Il suffit de rechercher la variable de langue à surcharger et de la redéfinir
- Recharger l'application

Le plus simple est de faire un ctrl+f dans le dictionnaire de langue et rechercher la chaîne de caractère voulue

Ex : Je souhaite modifier le bouton "Se connecter" par "S'identifier"



31 mars 2025

31 mars 2025

Copyright © 2026 Maarch