Fonctions spécifiques

Comment utiliser l'interface et l'assistant ?

Fonctions disponibles

Concaténation d'éléments

```
@concat
```

Permet de concaténer les valeurs de chacun des paramètres qui lui sont fournis.

Il est également possible de spécifier un séparateur.

Elle vous permet de réaliser des opérations de concaténation directement dans votre champ de recherche variable sans passer par un second champ de type spécifique.

Options de capture

Deux syntaxes sont possibles.

Sans spécifier de séparateur:

```
@concat(valeur1,valeur2,...,valeurN)
```

En spécifiant un séparateur:

```
@concat('@sep:separateur',valeur1,valeur2,...,valeurN)
```

Où *separateur* peut être le caractère (ou chaîne de caractères) de votre choix à l'exception du caractère quote simple.

Chaque valeur peut être :

- Une chaîne de caractères encadrée par des quotes simples.
- Un numérique.
- @self : une référence au champ lui-même.
- @<groupname> : une référence à une capture dans le filtre de champ. (effectuée dans le filtre de champ).

Date d'échéance

```
@dateecheance(@fldX,jours)
```

une macro qui fait une chose bien pratique de façon simple.

Options de capture

@fldX La référence au champ date.

jours Le nombre de jours que l'on souhaite ajouter à la date

Concaténation de Date

@concatdate permet de concaténer les valeurs de chacun des paramètres qui lui sont fournis et en plus leur applique la macro @Smartdate.

```
@concatdate(valeur1,valeur2,...,valeurN)
```

ou

```
@concat('@sep:separateur',valeur1,valeur2,...,valeurN)
```

Pour plus de détails se référer à la macro @concat

Concaténation d'éléments et conversion en Date

@concatdate permet de concaténer les valeurs de chacun des paramètres qui lui sont fournis et en plus leur applique la macro **@Smartdate**.

```
@concatdate(valeur1,valeur2,...,valeurN)
```

ou

```
@concat('@sep:separateur',valeur1,valeur2,...,valeurN)
```

Pour plus de détails se référer à la macro @concat

Valeurs alternatives

@alternative renvoi la première valeur non vide parmi une liste de valeurs possibles.

```
@alternative(@FLD4,@FLD3,'Non renseignée')
```

La macro **@alternative** prend au minimum 2 paramètres.

Chaque paramètre alternatif peut être :

- Liste à puceUne chaîne de caractère (encadrée ou non par des simples quotes ou des doubles quotes)
- Liste à puce Une référence à un champ (**@FLD***xxx*).

Exemple

ID	Description du champ	Valeur retrouvée
3	Adresse de facturation	2 Rue Pégoud, 90130 PETIT-CROIX
4	Recherche Adresse de livraison	1 Rue du Général de Gaulle, 90130 Montreux-Château

Nous avons deux champs. L'un nous retrouve l'adresse de facturation. Le second l'adresse de livraison. Nous voudrions, si l'adresse de livraison n'est pas spécifiée, choisir l'adresse de facturation.

Créons un champ supplémentaire de type spécifique et utilisons la macro @alternative

```
@alternative(@FLD4,@FLD3)
```

Le premier choix se porte sur le champ 4 (Recherche Adresse de livraison). S'il n'est pas trouvé on prendra la valeur du champ 3 (Adresse de facturation).

Et si aucun n'est trouvé et que l'on souhaite indexer une valeur par défaut :

```
@alternative(@FLD4,@FLD3,'Non renseignée')
```

From:
https://wiki.ezdev.fr/ - **EzGED Wiki**

Permanent link:
https://wiki.ezdev.fr/doku.php?id=doc:v3:acquisition:apprentissage:fonctionsspecifiques&rev=1514382161

Last update: **2023/03/17 09:56**