

Module Formulaire

Variables générales

Variables relatives aux pages :

Les variables relatives aux pages sont de la forme {page:ref:type}:

- **ref** : **Identifiant** ou **nom de code** de la page à laquelle faire référence, peut être un identifiant numérique d'une page, un nom de code alphanumérique, 'self' pour faire référence à la page courante ou bien 'father' pour faire référence à la page père de la page courante.
Attention, désigner une page par un nom de code, par self ou par father n'a de sens que si l'on se trouve dans une page (donc dans un modèle de page ou de rangée). Dans un flux RSS ou dans un plugin Wysiwyg, la page n'a aucun sens et vous ne pouvez donc pas employer le nom de code, self ou father pour désigner une page, vous devez dans ce cas employer un identifiant.
- **type** : type de donnée souhaité pour la page parmi les suivants :
 - url : adresse de la page,
 - id : identifiant de la page,
 - title : nom de la page,
 - exists : la page existe t'elle et est elle publique ?
 - link, linktitle : titre du lien de la page
 - website : identifiant du site de la page
 - codename : nom de code de la page si il existe,
 - language : langue de la page,
 - printurl : adresse de la page d'impression,
 - keywords, description, category, author, replyto, copyright, robots, pragma, refresh, metas : Métas données de la page.

Variables relatives aux données envoyées (via une adresse ou un formulaire) :

Ces variables correspondent à une variable provenant de la soumission d'un formulaire ou bien d'un paramètre du lien ayant amené à la page en cours.

Elles sont de la forme {request:type:name} :

- **type** : Correspond au type de variable attendu, parmi les suivants :
 - int : nombre entier,
 - string : chaîne de caractère,
 - safestring : chaîne de caractère sans code HTML,
 - bool : booléen,
 - array : tableau de valeurs,
 - email : email valide,
 - date : date valide sans heure (retourne une date au format MySQL : YYYY-MM-DD),
 - datetime : date valide avec heure (retourne une date au format MySQL : YYYY-MM-DD HH:MM:SS),

- **localisedDate** : date valide sans heure (retourne une date au format de votre langue actuelle : jj/mm/aaaa).
- **name** : Correspond au nom de la variable souhaité (nom du paramètre dans l'url ou bien nom du champ du formulaire).

Variables de session

Ces variables sont disponibles tout au long de la navigation du visiteur sur le site.

Elles sont de la forme **{session:type:name}** :

- **type** : Correspond au type de variable attendu, parmi les suivants :
 - **int** : nombre entier,
 - **string** : chaîne de caractère,
 - **bool** : booléen,
 - **array** : tableau de valeurs,
 - **email** : email valide,
 - **date** : date valide sans heure (retourne une date au format MySQL : YYYY-MM-DD),
 - **datetime** : date valide avec heure (retourne une date au format MySQL : YYYY-MM-DD HH:MM:SS),
 - **localisedDate** : date valide sans heure (retourne une date au format de votre langue actuelle : jj/mm/aaaa).
- **name** : Correspond au nom de la variable souhaité (nom de la variable de session).

Variables

Ces variables correspondent à des variables PHP classiques.

Elles sont de la forme **{var:type:name}** :

- **type** : Correspond au type de variable attendu, parmi les suivants :
 - **int** : nombre entier,
 - **string** : chaîne de caractère,
 - **bool** : booléen,
 - **array** : tableau de valeurs,
 - **email** : email valide
 - **date** : date valide sans heure (retourne une date au format MySQL : YYYY-MM-DD),
 - **datetime** : date valide avec heure (retourne une date au format MySQL : YYYY-MM-DD HH:MM:SS),
 - **localisedDate** : date valide sans heure (retourne une date au format de votre langue actuelle : jj/mm/aaaa).
- **name** : Correspond au nom de la variable souhaité (nom de la variable PHP).

Constantes

Ces variables correspondent à des constantes PHP classiques.

Elles sont de la forme **{constant:type:name}** :

- **type** : Correspond au type de constante attendu, parmi les suivants :

- **int** : nombre entier,
- **string** : chaîne de caractère,
- **bool** : booléen,
- **email** : email valide
- **date** : date valide sans heure (retourne une date au format MySQL : YYYY-MM-DD),
- **datetime** : date valide avec heure (retourne une date au format MySQL : YYYY-MM-DD HH:MM:SS),
- **localisedDate** : date valide sans heure (retourne une date au format de votre langue actuelle).
- **name** : Correspond au nom de la constante souhaité (nom de la constante PHP).

Variables serveur

Ces variables correspondent aux variables `$_SERVER` de PHP.

Elles sont de la forme `{server:type:name}` :

- **type** : Correspond au type de variable attendu. Dans le cas des variables serveur, elles sont usuellement de type **string**.
- **name** : Correspond au nom de la variable (nom de la variable serveur désirée).

Variables relatives aux utilisateurs

Les variables relatives aux utilisateurs sont de la forme `{user:id:type}`:

- **id** : Identifiant de l'utilisateur auquel faire référence, peut être un identifiant numérique d'un utilisateur ou bien **'self'** pour faire référence à l'utilisateur courant (uniquement si la vérification des droits côté client est activé dans les paramètres Automne).
- **type** : Correspond au type de variable attendu, parmi les suivants :
 - **id** : identifiant numérique de l'utilisateur,
 - **login** : identifiant de l'utilisateur,
 - **firstName** : prénom,
 - **lastName** : nom,
 - **fullName** : prénom et nom,
 - **email**: email,
 - **language** : retourne un objet CMS_language correspondant à la langue du compte utilisateur,
 - **active** : booléen, **true** pour actif, **false** pour inactif,
 - **deleted** : booléen, **true** pour supprimé, **false** pour existant.

Traitement des variables - Helpers

Pour retraiter les variables ci-dessus ainsi que les différentes variables des éléments des modules Polymod, vous avez à votre disposition un ensemble de helpers capable de

transformer le contenu de ces variables. Vous pouvez ainsi les tronquer, en changer la casse, en nettoyer le contenu (supprimer le HTML par exemple), etc.

Le format de ces helpers est la forme `{helper:function:content}` :

- **function** : [Fonction PHP](#) à employer pour transformer le contenu de votre variable. Il s'agit du nom de la fonction à employer, par exemple :
 - [substr](#) : Retourne un segment de chaîne.
 - [strtoupper](#) : Renvoie une chaîne en majuscule.
 - [strtolower](#) : Renvoie une chaîne en minuscule.
 - [htmlspecialchars](#) : Convertit les caractères spéciaux en entités HTML.
 - [urlencode](#) : Encode une chaîne en URL.
 - [strip_tags](#) : Supprime les balises HTML et PHP d'une chaîne.
 - [strpos](#) : Trouve la position d'un caractère dans une chaîne.
 - [str_replace](#) : Remplace toutes les occurrences dans une chaîne.
 - ...
- En plus des fonctions PHP classique, vous pouvez aussi employer toute fonction PHP ou méthode statique de classe [définie dans Automne](#) ou que vous avez vous même défini dans votre propre code PHP. Par exemple :
 - [io::ellipsis](#) : Tronque une chaîne et ajoute une ellipse.
 - [io::isPositiveInteger](#) : Vérifie si un nombre est un entier positif.
 - ...
- Si une erreur survient lors du traitement de la fonction demandée (la fonction n'existe pas ou son usage est incorrect), le helper retournera la valeur d'origine sans la modifier. Consultez le log d'erreur d'Automne pour obtenir plus d'informations sur la nature de l'erreur.
- **content** : Contenu à faire modifier par le helper. Il peut s'agir d'un contenu fixe ou d'une variable. Exemples :
 - `{helper:strtoupper:abc}` : Passe en majuscule la chaîne "abc".
 - `{helper:strtoupper:{request:string:text}}` : Passe en majuscule le contenu de la variable {request:string:text}.

Paramètres

Un certain nombre de fonctions nécessitent le passage de paramètres pour fonctionner. Exemple, la fonction `substr` nécessite deux paramètres pour découper une chaîne : un pour déterminer où commencer la coupe, un autre pour déterminer la longueur de la coupe. Vous pouvez spécifier autant de paramètres que vous le souhaitez en les séparant par des caractères "pipes" : | .

Exemples :

- `{helper:substr:abcdef|1|3}` : Coupe la chaîne "abcdef" à partir du premier caractère et conserve 3 caractères de long.
- `{helper:strtoupper:{request:string:text}|0|{var:int:length}}` : Coupe le contenu de la variable {request:string:text} à {var:int:length} de longueur.

Imbrication des helpers

Il est tout à fait possible d'imbriquer les helpers les uns dans les autres pour réaliser une chaîne de traitements.

Exemples :

- **{helper:ucfirst:{helper:strtolower:{var:string:text}}}** : Passe en minuscules le contenu de la variable {var:string:text} puis passe en majuscule la première lettre du contenu résultant.
- **{helper:io::ellipsis:{helper:strip_tags:{var:string:text}}|10}** : Supprime les tags HTML du contenu de la variable {var:string:text} puis fait une ellipse à 10 caractères du contenu résultant.