CONCEVOIR LES DÉMARCHES EN LIGNE AVEC PUBLIK

Formation administrateur fonctionnel

AIDE ET DOCUMENTATION

- Ce support
- Guide de l'administrateur fonctionnel (https://doc-publik.entrouvert.com/admin-fonctionnel)
- Questions sous forme de tickets sur Redmine
- Catalogue des démarches (https://catalogue.publik.love/)
- Site communautaire Tracim (https://publik.tracim.fr/ui) pour des échanges directs entre utilsateurs de Publik

TERMINOLOGIE

- **Portail citoyen** = Front-Office
- **Portail agents** = Back-Office
- **Utilisateur** = une personne qui dispose d'un compte d'accès à Publik, usager ou agent
- **Une demande** = le contenu d'un formulaire saisi par un usager
- **Une démarche** = le formulaire + le circuit de traitement (le workflow)
- Une demande est traitée par une ou des **fonctions**
- Un rôle est imputé à une fonction
- Glossaire : https://doc-publik.entrouvert.com/glossaire/

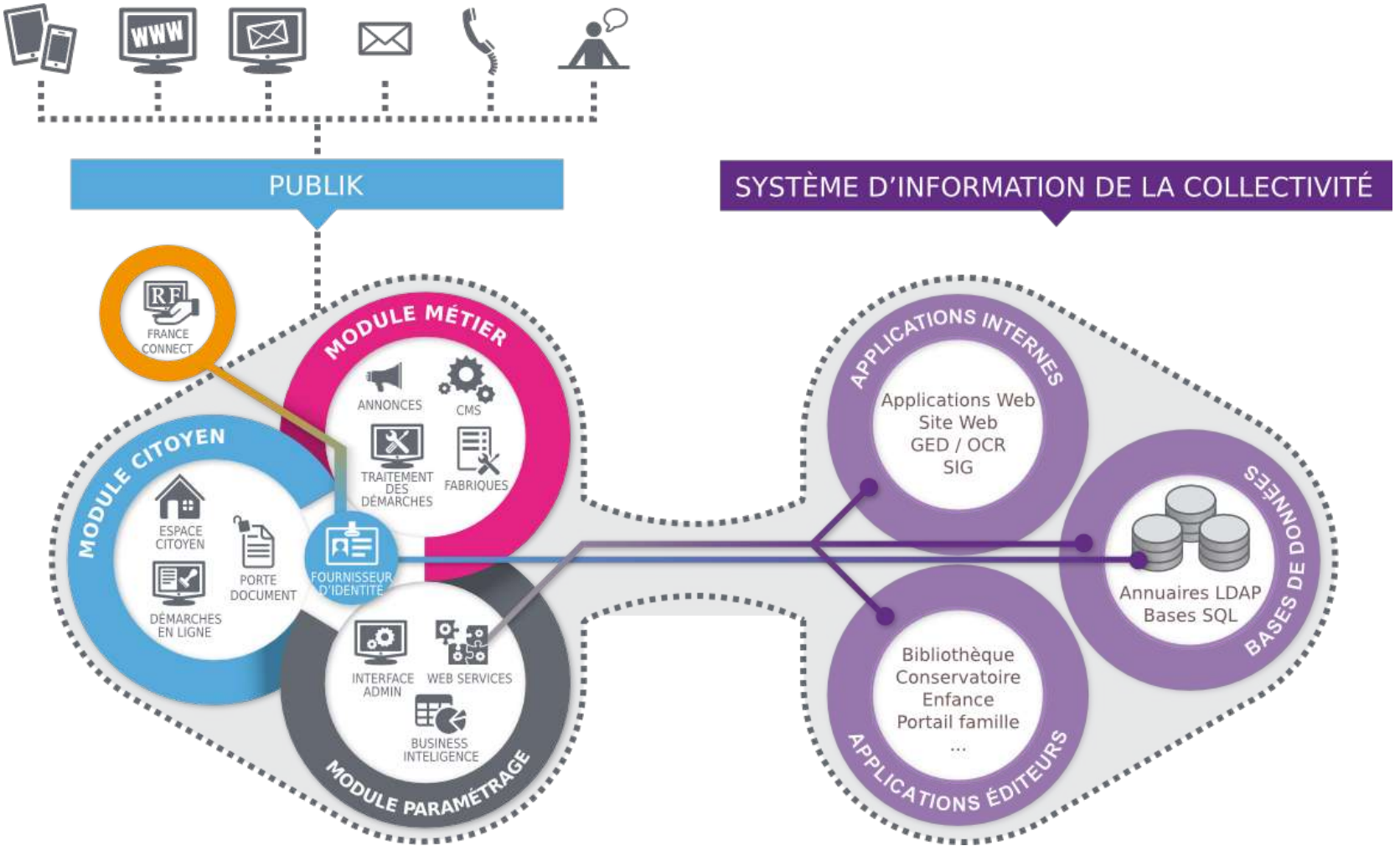
IDÉES FORCES

- Simplification pour les citoyens et les agents. C’est plus compliqué mais indispensable.
- Transversalité, pour offrir un point d’entrée unique aux usagers plutôt qu’une juxtaposition de portails sectoriels.
- Utilisation pour l’accueil, les téléopérateurs, le multi-canal.
- Parfois plus coûteux de faire des connecteurs avec des applications dépassées que de faire le traitement dans Publik.
- Application.

DÉMATÉRIALISATION

1. L'existant doit-il être conservé à l'identique ? Non, il faut simplifier.
2. Conception d'un formulaire structuré, avec un vocabulaire adapté.
3. Identification des intervenants (fonctions) dans le traitement.
4. Définition des étapes du traitement.
5. Définition des actions à chaque étape du traitement, avec fonctions associés (permet de définir les possibilités de chaque type d’intervenant à chaque étape).

PANORAMA DE PUBLIK



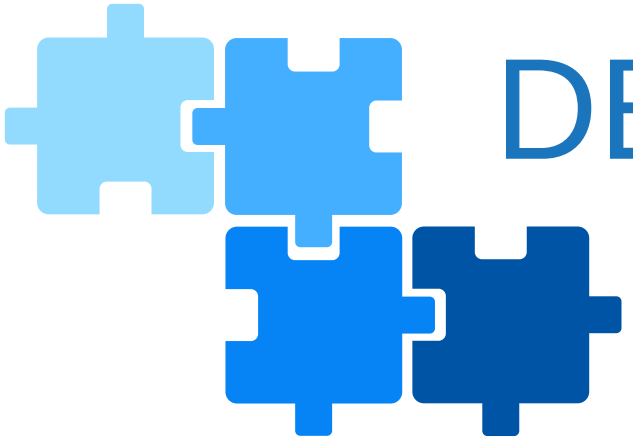
entr'ouvert

L'INTERFACE DE TRAITEMENT DES DÉMARCHES

INTERFACE DE TRAITEMENT

- **C’est l’endroit où les agents peuvent voir l’ensemble des demandes et les traiter**
- Vue globale
- Vue par formulaire, avec la possibilité d’enregistrer des vues
- Vue sur une carte (si action géolocalisation dans le workflow)
- Recherche d’une demande précise, possibilité de filtrer
- Export tableur ou csv

entr'ouvert



LA FABRIQUE DE FORMULAIRE

FORMULAIRE

- **Créer un nouveau formulaire**
  - ex-nihilo
  - en important les champs depuis un autre formulaire
  - en important un xml (catalogue, Tracim, échange)
- **Notions à balayer :**
  - informations (titre, catégorie, mots-clés)
  - workflows (choix du workflow, rôles attribués aux fonctions)
  - options
  - catégories
  - sauvegarde / historique

FORMULAIRE ET WORKFLOW

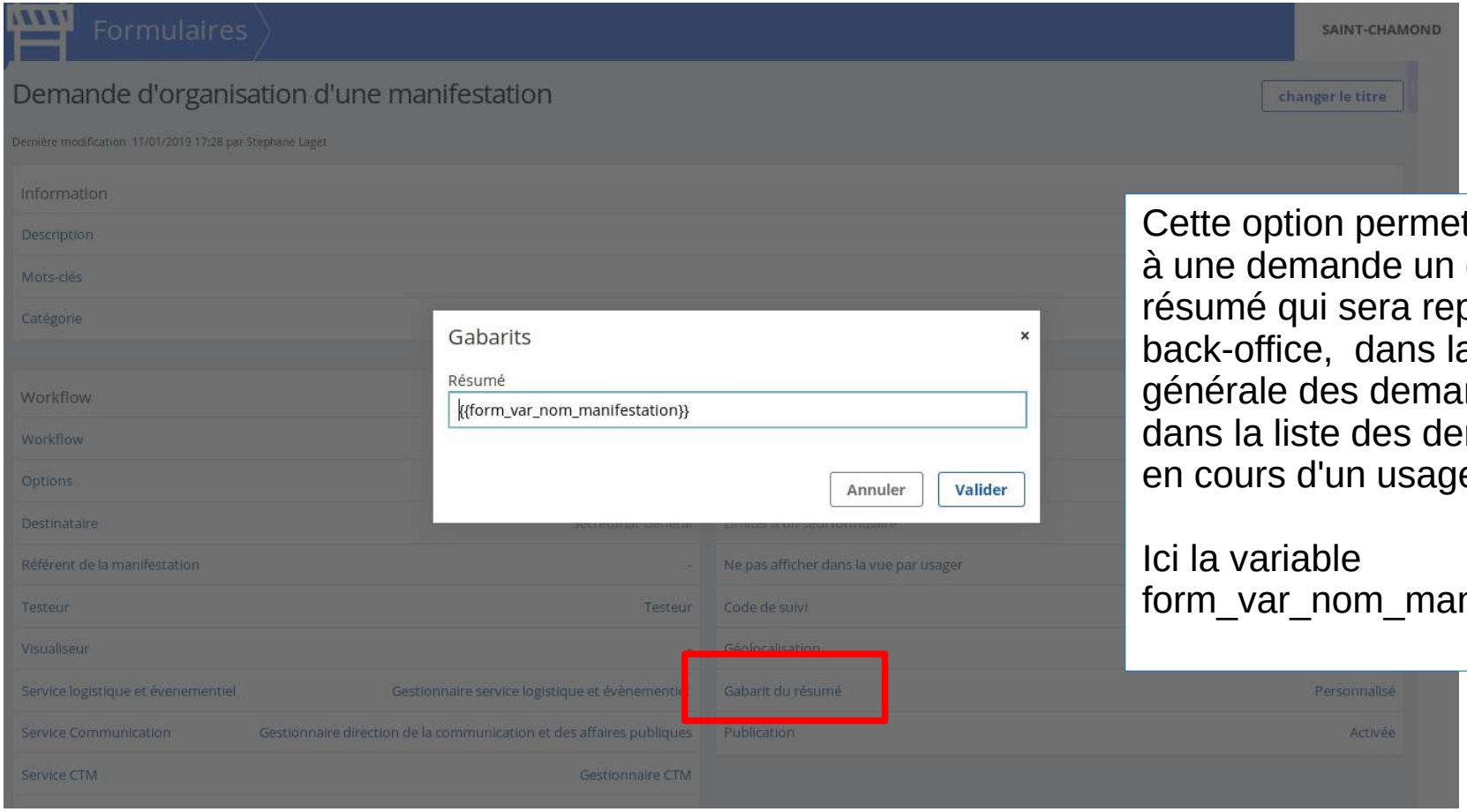
- C’est sur le formulaire que se fait l’association entre un formulaire et son workflow.
- Au départ workflow par défaut imposé (un workflow est obligatoire)
- Attribution d’un rôle aux "acteurs" (autrement dit « fonctions »)
  - Destinataire et autres fonctions définies dans le workflow
  - Utilisateur : qui peut remplir le formulaire (si aucun rôle spécifié, accessible à tous)
  - Saisie back-office : qui (quel rôle agent) peut saisir une demande depuis le back-office

PARAMÈTRES DU FORMULAIRE

- **Page de confirmation** : non éditable, permet de se relire et revenir sur la page du champ à corriger
- **Limiter à un seul formulaire**
  Lorsqu'on souhaite que les utilisateurs répondent une seule fois (sondage par exemple).
  Valable jusqu'à l'anonymisation.
- **Traitement**
  Télécharger tous les fichiers
  Ne pas afficher dans la vue usager : recherche des formulaires depuis l'usager (exemple)
- **Code de suivi** : générer un code de suivi unique pour permettre le suivi des demandes sans compte usager. Gestion de la durée de vie des brouillons.
- **Géolocalisation** : active la potentialité d'enregistrer des coordonnées x,y ; puis valeurs à attribuer par action dans le workflow
- **Gabarits**
- **Publication** :
  - Activation du formulaire ou pas (directement visble en front-office si catégorie d'appartenance publiée)
  - Possible de rediriger un formulaire désactivé vers une autre URL : le formulaire est affiché et un clic sur ce dernier redirige vers l'URL saisie.
  - Programmation par dates

OPTIONS DU FORMULAIRE

- **Le gabarit de résumé**

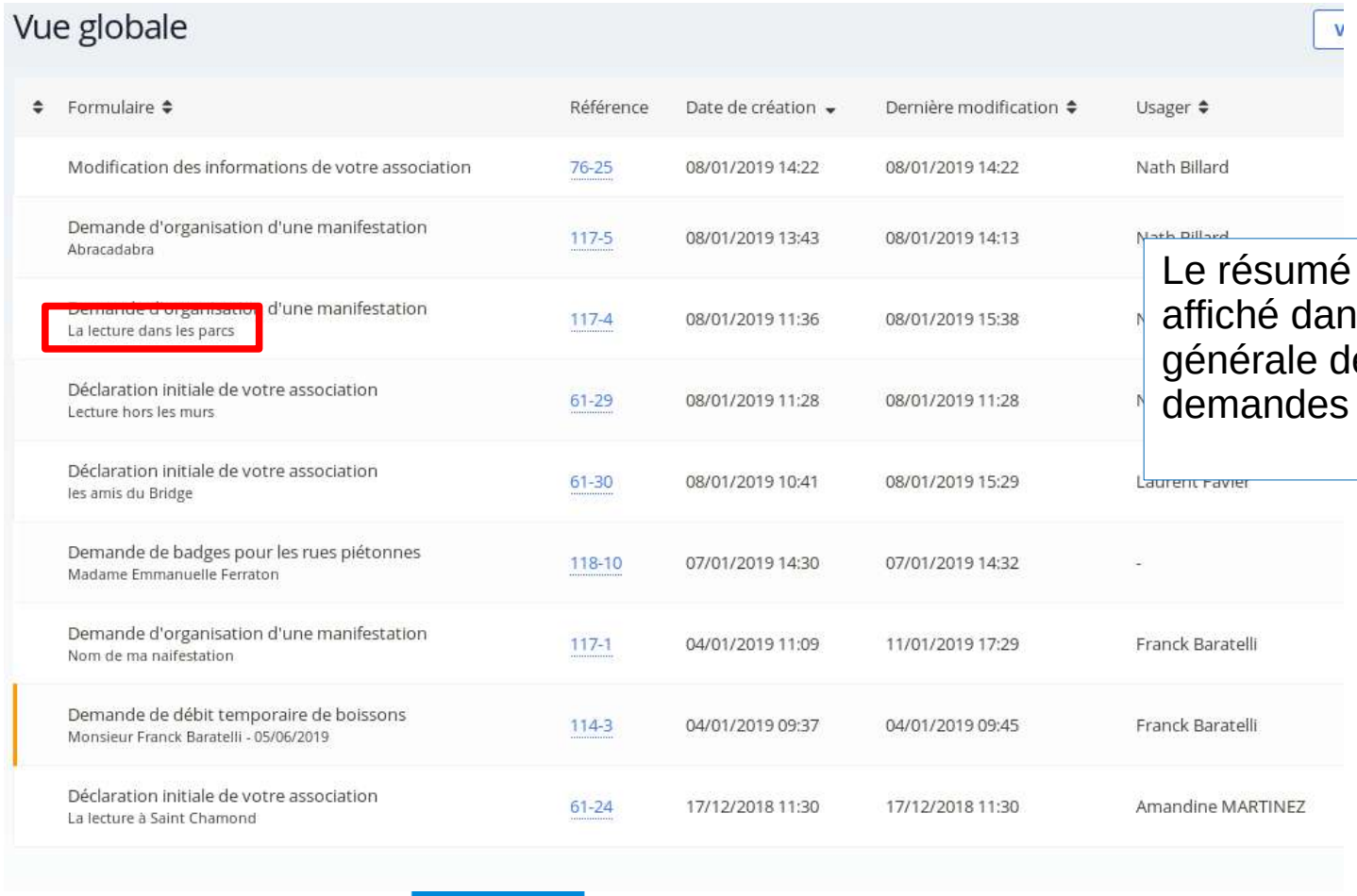


Cette option permet d'ajouter à une demande un court résumé qui sera repris, en back-office, dans la liste générale des demandes et dans la liste des demandes en cours d'un usager.

Ici la variable form_var_nom_manifestation

OPTIONS DU FORMULAIRE

- **Le gabarit de résumé (suite)**



Le résumé est affiché dans la vue générale des demandes

ÉDITION DES CHAMPS

- Cliquer sur « **Édition** »
- Choisir un libellé (qui sera le texte affiché à l'utilisateur) et un type.
- Types de champs
  - de saisie de données : Texte (ligne), Texte long, Courriel, Case à cocher (choix unique), Fichier, Date , Liste , Liste à choix multiple, Carte, Mot de passe
  - de mise en forme du formulaire : Titre, Sous-titre, Commentaire, Page
  - donnée calculée
  - les blocs de champs (si existants)
- Après création on peut
  - éditer, supprimer ou dupliquer un champ en le survolant
  - ordonner les champs par cliquer/déplacer (« drag and drop »)

CHAMPS, PARAMÉTRAGE « GÉNÉRAL »

- **Identifiant** :
  - Pour identifier le champ en question au sein du formulaire. Les variables de formulaire, intégrant cet identifiant, permettent de faire des formulaires conditionnels et autres opérations dans les workflows, modèles…
  - Exemple :
    - Identifiant : type
    - Le nom de la variable sera form_var_type
  - Déterminer une **charte de nommage** pour les identifiants et s'y tenir
    - Exemple : un champ « Type de signalement » qui peut se retrouver sur plusieurs formulaires
    - Nommer le « signalement_type » ou « typesignalement », mais garder toujours la même logique de nommage
  - Choisir un **identifiant différent pour chaque champ**, même si les champs sont sur des pages différentes.

CHAMPS, PARAMÉTRAGE « GÉNÉRAL »

- Champ « **Obligatoire** » ou pas (par défaut coché i.e. obligatoire)
- « **Remarque** » : permet d'apporter une aide au répondant (texte affiché sous le champ)
- « **Données / Source de données** » (pour les champs de type listes) :
  - Pour utiliser une même liste dans plusieurs formulaires, ou plusieurs fois, dans le même formulaire
  - Construit au sein de Publik
    - avec des fiches
    - automatiquement depuis un agenda
  - ou correpond à un référentiel proposé en webservice par un logiciel tiers

TP « FICHES COMME SOURCES DE DONNÉES »

TP Fiches

- Créez un modèle de fiche simple avec quelques champs
- Donnez vous les droits nécessaires pour pouvoir remplir des fiches (une nouvelle entrée « Fiches » sera disponible dans le menu pour saisir des fiches)
- Pensez à bien configurer le gabarit du résumé pour générer automatiquement une source de données
- Saisissez une ou deux fiche
- Dans votre formulaire, créez un champ de type liste qui va exploiter la source de donnée créée.

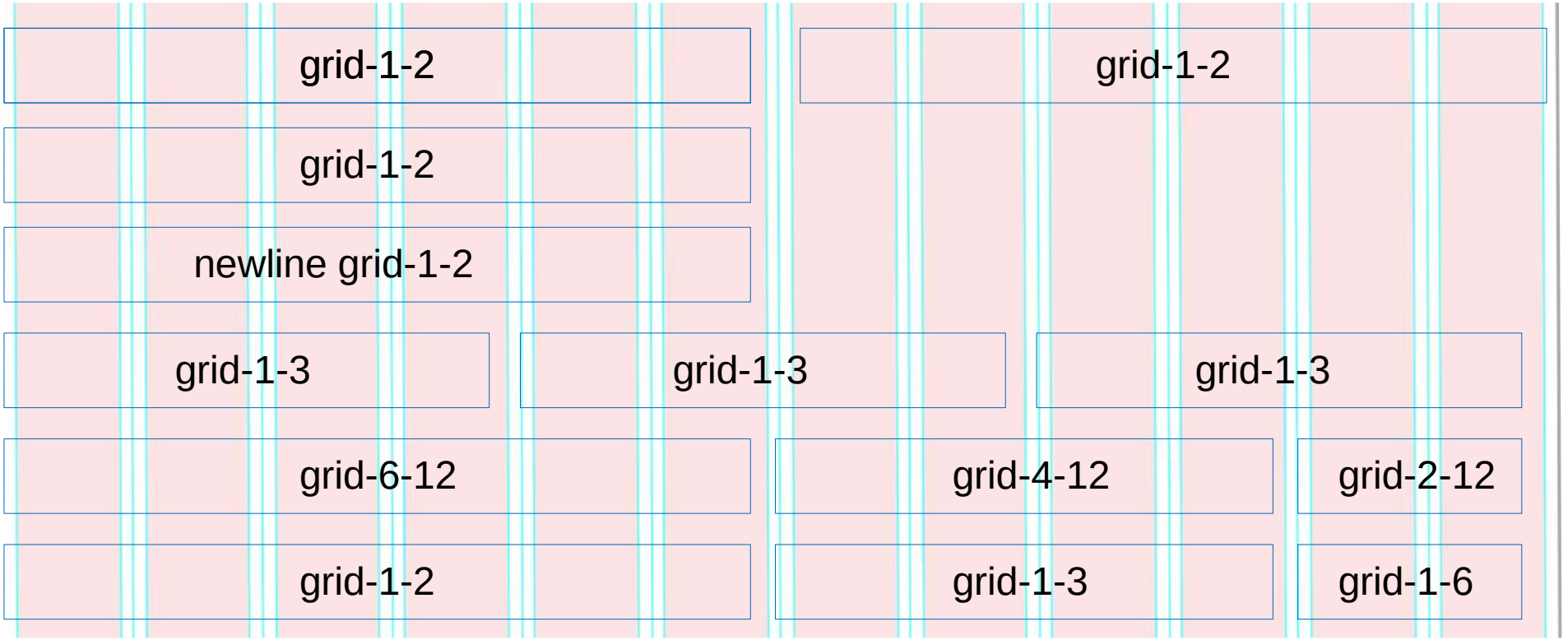
CHAMPS, PARAMÉTRAGE « AFFICHAGE »

- « **Affichage** », dont, décochés par défaut :
  - « **Tableaux de traitement** » : détermine si le champ concerné doit figurer, par défaut, dans les tableaux back-office (une colonne du listing) des agents traitant les demandes
  - « **Statistiques** », pour les champs liste : fait apparaître ce champ dans les cellules « Visualisation de données »
- « **Classe supplémentaire pour les styles CSS** » : permet d'indiquer une classe CSS pour le champ. Cette classe doit évidemment exister dans la feuille de style du thème choisi pour être opérante.
  - 4 classes de mise en forme disponibles par défaut : pk-information, pk-attention, pk-success, pk-error
    https://doc-publik.entrouvert.com/admin-fonctionnel/modifier-le-contenu-des-portails/classes-css/

LES GRIDS

Positionnement possible avec "grid" qui fonctionne sur 12 colonnes

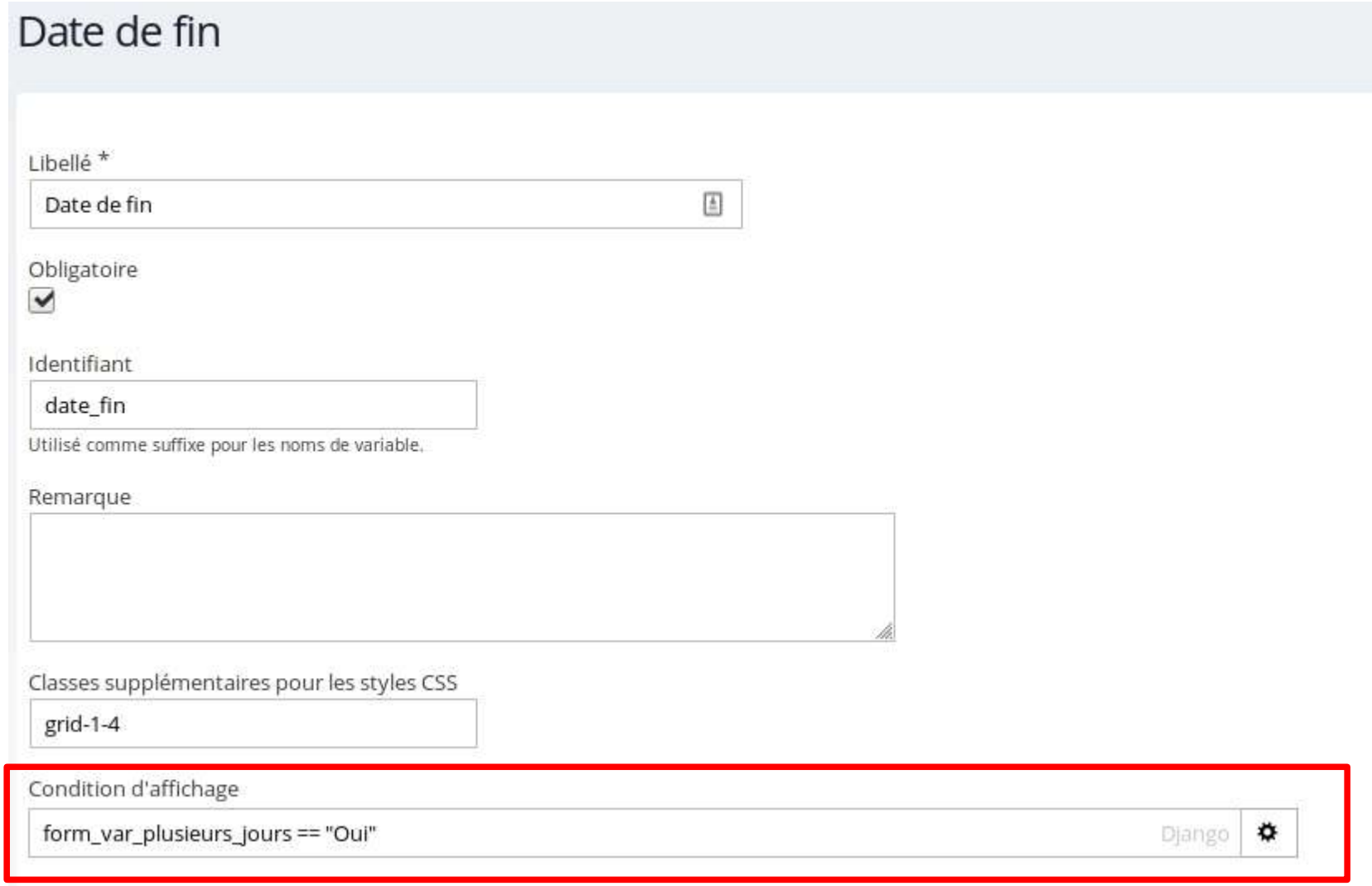
- grid-1-2 : la ligne comprend 2 colonnes au total et le champ en occupera une (la moitié de la ligne donc)
- on peut forcer la mise à la ligne avec newline : newline grid-1-2



LES CHAMPS CONDITIONNELS

- **Publik permet d'inclure des champs conditionnels, qui ne vont s'afficher que si une condition est remplie.**
- Cette « condition d'affichage » est définie au format gabarit Django
- Ajouter une condition :
  - form_var_nom_de_la_variable == « La réponse qui déclenche l'affichage du champ »
  - Attention à la casse (majuscules/minuscules), aux espaces, aux tirets, aux accents…

LES CHAMPS CONDITIONNELS



CHAMPS, PARAMÉTRAGE « AVANCÉ »

- **Pré-remplissage**
  - avec les données utilisateur du compte citoyen, le plus souvent.
  - on peut également pré-remplir avec du texte, une variable…

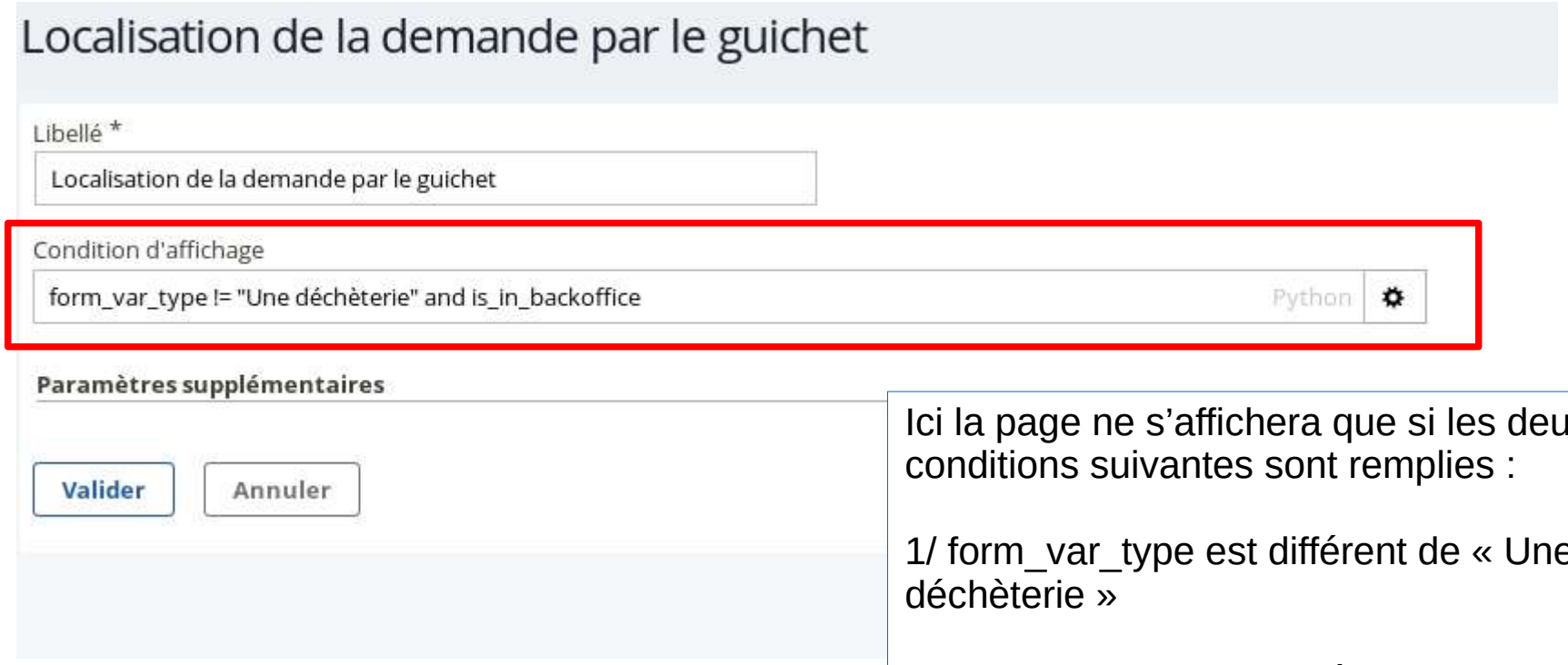
- **Validation** : pour contrôler ce qui est saisi
  - types pré-définis (code postal, siren…)
  - talidation django
  - expression rationnelle : voir https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-formulaires/form-champs/champ_texte-ligne/

- **Paramètres spécifiques à certains champs**
  - Date minimale, maximale, date dans le futur, date du jour
  - Nombre maximal de choix (liste à choix multiple)
  - Condition de sortie de page (champs page)

LES PAGES CONDITIONNELLES

- **Les pages conditionnelles**
  - Les champs obligatoires le sont au sein d'une page (si une page non affichée à l'usager contient des champs marqués obligatoires, ils ne lèveront pas d'alerte)
  - Créer au minimum deux pages.
  - Éditer le champs page pour ajouter une condition sous la forme :
    - form_var_nom_de_la_variable == « La réponse qui déclenche l'affichage de la page »
    - Attention à la casse (majuscules/minuscules), aux espaces, aux tirets, aux accents…

LES PAGES CONDITIONNELLES



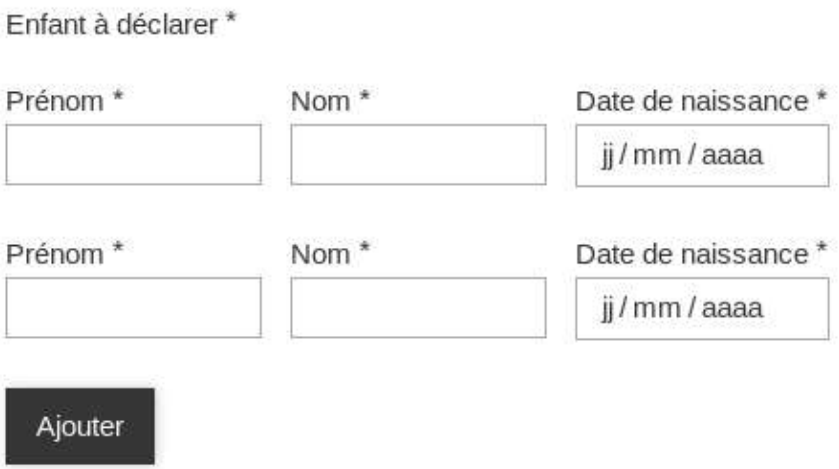
Ici la page ne s'affichera que si les deux conditions suivantes sont remplies :

1/ form_var_type est différent de « Une déchèterie »

2/ le formulaire est affiché en back-office

LES BLOCS DE CHAMPS

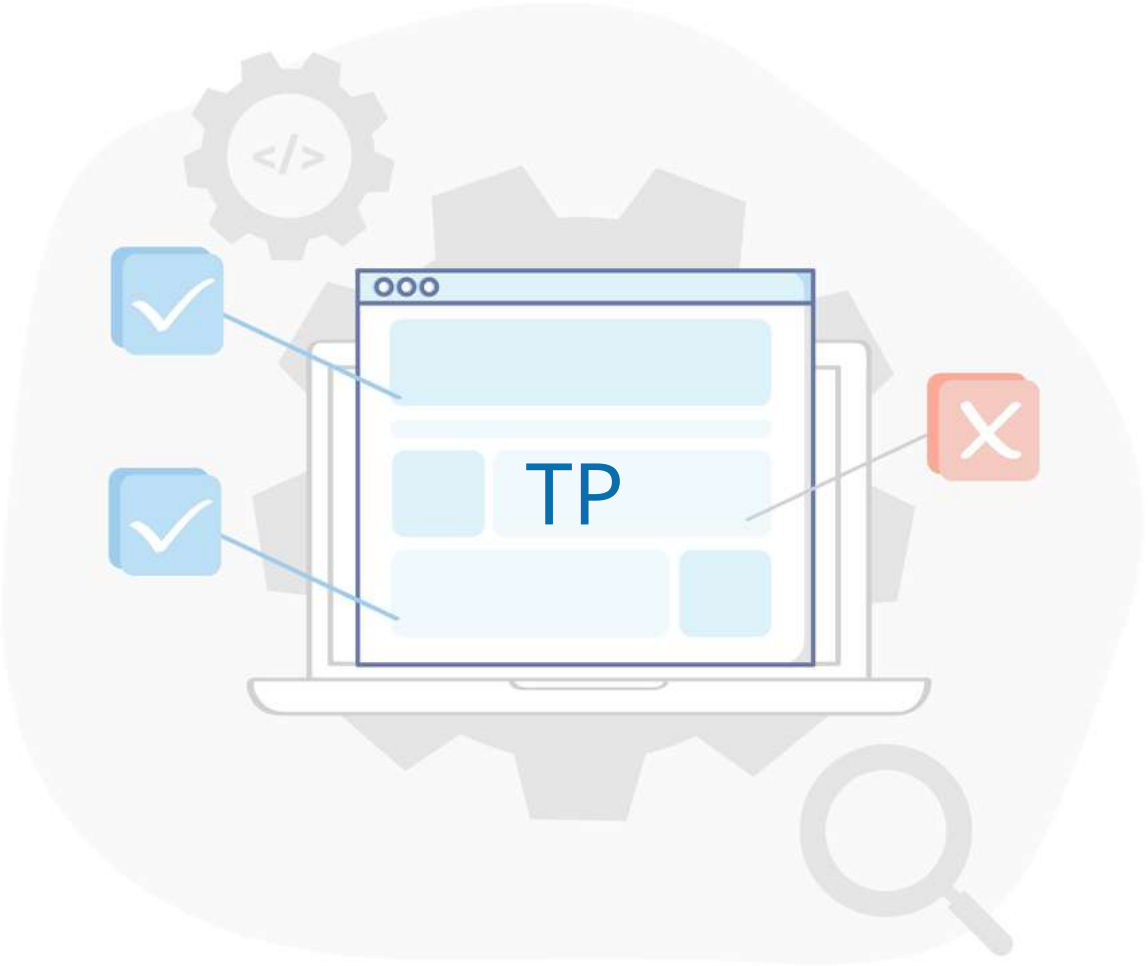
- Les blocs de champs permettent
  - d'afficher plusieurs fois le même « morceau » de formulaire et l'usager peut en rajouter
  - de dupliquer sur plusieurs formulaires, la même séquence de champs
- Par exemple : un bloc de champ « enfant » (prénom, nom, date de naissance), peut être affiché deux fois dans le formulaire à l'ouverture mais l'usager peut aussi rajouter de tels blocs



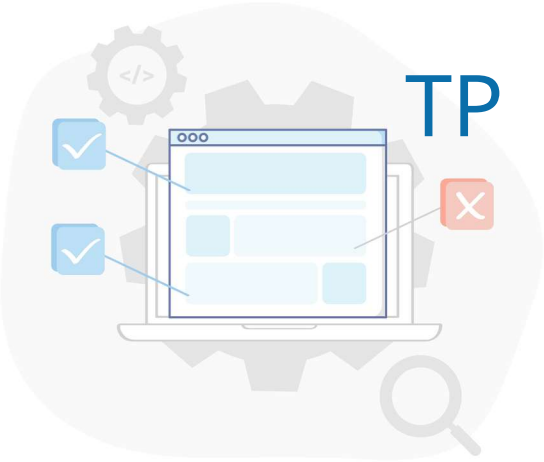
LES BLOCS DE CHAMPS

- Permettent de ne pas se répéter. Par exemple avec un bloc de champ coordonnées qui sera utilisé dans de nombreux formulaires
- La mise à jour du bloc de champ entraîne la mise à jour de tous les formulaires qui l'utilisent.
- Permet de répéter plusieurs fois une série d'informations structurées d'une façon beaucoup plus efficace qu'un tableau
- block_var_IDENTIFIANT-DU-CHAMPS permet de désigner des variables dans l'environnement du champ
- Utiliser la vue inspect pour être sûr de ne pas se tromper sur le nom des variables dans les blocs, exemple {{ form_var_enfant_scolarise_0_prenom }} désigne le champ prénom de la première occurrence du bloc enfant_scolarise

entr'ouvert



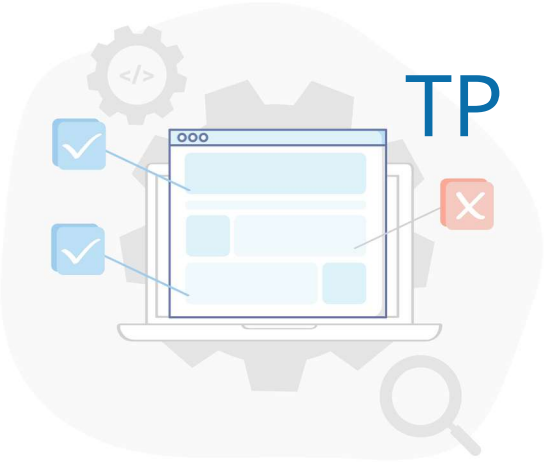
entr'ouvert



TP formulaire 1

- Créez un formulaire ex nihilo :
  - mettez votre prénom dans le titre : «TP1 - Théodule »,
  - branchez-le sur le workflow par défaut
  - mettez-le dans la catégorie « Formation »
- Ajoutez des champs signalétiques : prénom, nom, courriel, ville, téléphone
- Pré-remplissez les champs utilisateurs avec les données du compte usager s'il existe
- Déterminez les champs qui apparaîtront dans le listing des demandes en backoffice
- Testez ! Remplissez le formulaire et allez voir sur le listing

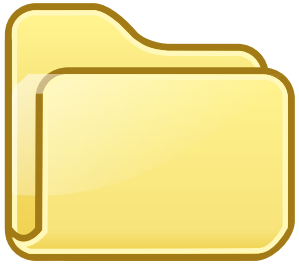
entr'ouvert



TP formulaire 2

- Reprenez le formulaire précédent
- Ajoutez un champ de type liste « type de signalement », ajoutez 3 éléments : « Problème de stationnement », « Nid de poule » et « Autre »
- Ajoutez un champ texte « Autre, précisez »
- En utilisant les conditions d'affichage, affichez le champ « Autre, précisez », seulement si « Autre » est sélectionné dans le champ « type de signalement »

entr'ouvert



LES CATÉGORIES

CATÉGORIES

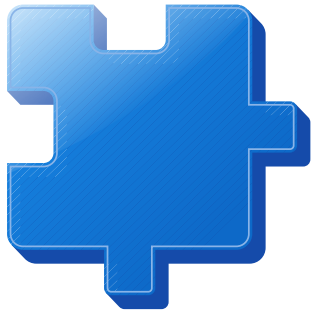
- **Existent pour les formulaires, workflow, fiches, agendas, sources de données…**
  - classement en backoffice
  - droits délégués pour fabriquer workflows et formulaires (Studio)
  - droits particuliers pour les demandes (exports et statistiques)
- **Les catégories dans le portail citoyen**
  - cellule pour afficher tous les formulaires d'une catégorie
  - description : affichée en front-office
  - image : la catégorie doit être utilisée dans combo (ajout d'une cellule « démarches d'une catégorie ») ; une image vide se crée automatiquement dans les ressources, on peut la remplacer par l'image souhaitée.

CATEGORIES

- **Servent à ranger les formulaires**
  - Dans le BO
  - Sur le portail citoyen

- **Paramètres**
  - Description : affichée en front-office
  - Rôles pour les exports, la gestion et les statistiques

entr'ouvert



LES VARIABLES

LES VARIABLES

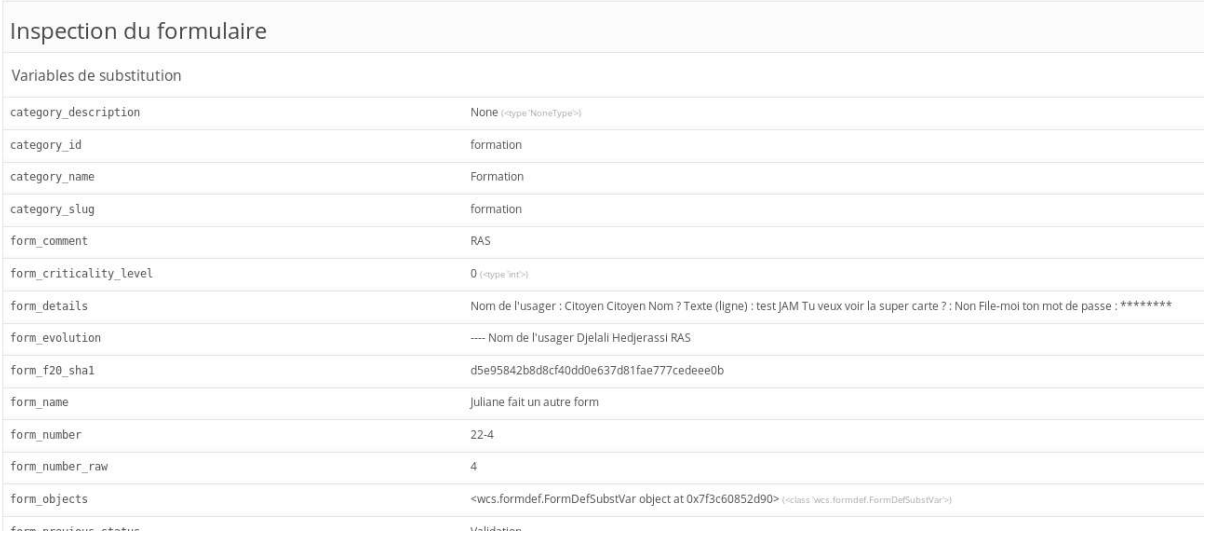
- Les variables sont essentielles pour construire les modèles de courriels et de documents
- Variables liées à la demande de l'utilisateur : {{form_var_identifiant_du_champ}}
- Variables liées au formulaire : {{form_details}}, {{form_name}}, {{form_url}}, {{ form_url_backoffice }}, {{ form_tracking_code }}
- Variables systèmes liées au site, aux catégories ou à l'utilisateur {{ site_name }}, {{site_url}}, {{session_user_var_nom}}
- Variable d'un formulaire backoffice (paramétré dans le workflow) {{form_workflow_form_var_identifiant_du_formulaire_var_identifiant_du_champ}}
  - Exemple : formulaire en back-office avec un identifiant «contact», on obtiendra le contenu du champ téléphone (dont le nom de variable est telephone) en mettant dans le modèle de message {{form_workflow_form_var_contact_var_telephone}}

Dans la documentation :

- https://doc-publik.entrouvert.com/admin-fonctionnel/utiliser-les-variables/

L'INSPECTION D'UNE DEMANDE

- Dans l'interface de « Traitement » d'une demande, il y a un lien vers l'inspection de la demande, on y trouve :
  - la liste de toutes les variables disponibles **dans le contexte de la demande**.
  - Les rôles associés à chaque fonction
  - Les actions qui ont été exécutées pour la demande en question
- C'est le premier écran à regarder en cas de difficulté (cela implique qu'il faut avoir complété une demande).

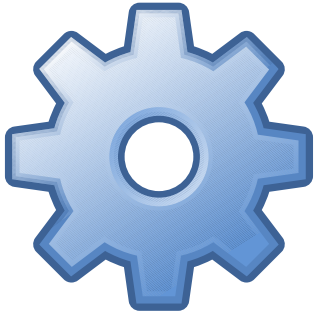


Inspection du formulaire

Variables de substitution

category_description	None (<type 'NoneType'>)
category_id	formation
category_name	Formation
category_slug	formation
form_comment	RAS
form_criticality_level	0 (<type 'int'>)
form_details	Nom de l'usager : Citoyen Citoyen Nom ? Texte (ligne) : test JAM Tu veux voir la super carte ? : Non File-moi ton mot de passe : ********
form_evolution	---- Nom de l'usager DJelali Hedjerassi RAS
form_f20_sha1	d5e95842b8d8cf40dd0e637d81fae777cedeee0b
form_name	Juliane fait un autre form
form_number	22-4
form_number_raw	4
form_objects	<wcs.formdef.FormDefSubstVar object at 0x7f3c60852d90> (<class 'wcs.formdef.FormDefSubstVar'>)
form_previous_status	Validation

entr'ouvert



LA FABRIQUE DE WORKFLOWS

WORKFLOWS

- **Workflow = le circuit que suit une demande dans son traitement**
- Une série d'étapes appelées statuts
- À chaque statut on définit ce qui peut se passer pour la demande via une série d'actions disponibles
- Les actions sont parfois déclenchées par des utilisateurs ayant une fonction explicite, elle sont parfois automatiques

GÉRER UN WORKFLOW

- From scratch
- En dupliquant le workflow par défaut ou autre (workflow de base que vous auriez défini)
- En important un workflow existant (fichier .wcs).
- Liens à droite pour la suppression, la copie, l'export et l’historique

LES STATUTS

- Les statuts sont les différentes étapes pouvant être parcourues par une demande.
- Créer / Supprimer un statut
- Options :
  - **Changer le nom du statut.** Il faut noter que ce nom apparaît pour l'usager (citoyen) d'une part et pour l'agent traitant le formulaire (le destinataire) d'autre part.
  - **Changer la visibilité du statut :** pour le cacher à l'usager.
  - **Changer le caractère final du statut** : affirmer le caractère final d'un statut même s'il peut déboucher sur d'autres statuts (permet de faire figurer la demande comme « terminée » dans les listings de l'agent et de l'usager).
  - Utile de définir les couleurs et les aides contextuelles pour faciliter le travail des agents.

LES ACTIONS

- Définissent ce qui se passe lorsqu'une demande se trouve dans un statut donné.
- Ajouter une action avec la liste déroulante
- Supprimer / éditer en cliquant dessus
- Il y a des actions pour :
  - faire évoluer la demande dans le workflow (changement de statut)
  - échanger avec l'usager ou l'agent (e-mails, SMS, commentaires…) ou un logiciel (webservice)
  - gérer la demande (éditer, anonymiser, supprimer…)
  - agir sur le demandeur (attribution de rôle)
- Attention les actions sont ordonnées au sein de chaque statut
  - Par exemple, si vous avez une action automatique de changement de statut sans condition, toutes les actions qui suivent ne seront pas exécutées.
- L'entrée dans le workflow se fait par le premier statut de la liste

LES ACTIONS

- Changer de statut
  - Saut manuel
  - Saut automatique

- Échanger
  - Alerte
  - Commentaire
  - Courriel
  - Création de document
  - Fichier joint
  - Formulaire
  - Message dans l’historique
  - Notification au demandeur
  - SMS (optionnel)
  - Webservice

- Agir sur la demande
  - Anonymisation
  - Création d’une demande
  - Création d’une fiche
  - Données de traitement
  - Édition
  - Géolocalisation
  - Liaison fonction / rôle
  - Modification d’une fiche
  - Rattachement d’un usafer à une fiche
  - Redirection web
  - Resoumission
  - Suppression
  - Suppresion du code de suivi
  - Workflow externe

- Agir sur l'usager
  - Ajout d’un rôle
  - Modification d’un profil
  - Retrait d’un rôle

https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-de-workflows/les-actions-de-workflow/

FONCTIONS, RÔLES ET UTILISATEURS

- Les actions dans un workflow sont associées à une fonction (exemple : "Destinataire")
  - les fonctions sont attribuées à des rôles, soit via les formulaires (alors pré-determiné), soit via l'action "Liaison fonction/rôle" (alors dynamique)
- Les fonctions vont ainsi déterminer ce qu'un utilisateur, ayant le rôle associé, peut faire en terme de traitement des demandes.
- Tous les utilisateurs qui ont un rôle associé à une fonction pourront voir les demandes dans les listings, même quand ils n'ont aucun traitement à y faire
- On peut ajouter plusieurs rôles à une fonction

FORMULAIRE ET WORKFLOW, LE RETOUR

- **Dans l'encart workflow du formulaire.**
  - « Destinataire » indique le rôle que doit avoir les agents pour traiter les demandes (à priori indispensable). Ce rôle peut être modifié en cours de traitement de la demande.
  - « Rôles du demandeur » : pour indiquer le rôle que doit avoir l'utilisateur pour qu'il puisse remplir le formulaire (optionnel).
  - « Rôle pour la saisie backoffice » : pour indiquer le rôle que doit avoir un agents pour être autorisé à saisir des demandes pour au guichet i.e. via le back-office (optionnel)
- Fonctions supplémentaires.
  - Toutes ces fonctions peuvent ne pas avoir de rôle attribué via le formulaire et se voir un rôle assigné seulement au cours du traitement (i.e. dans le workflow).
  - Dès qu'un rôle est associé à une fonction, les utilisateurs qui ont ce rôle pourront voir le contenu de la demande.

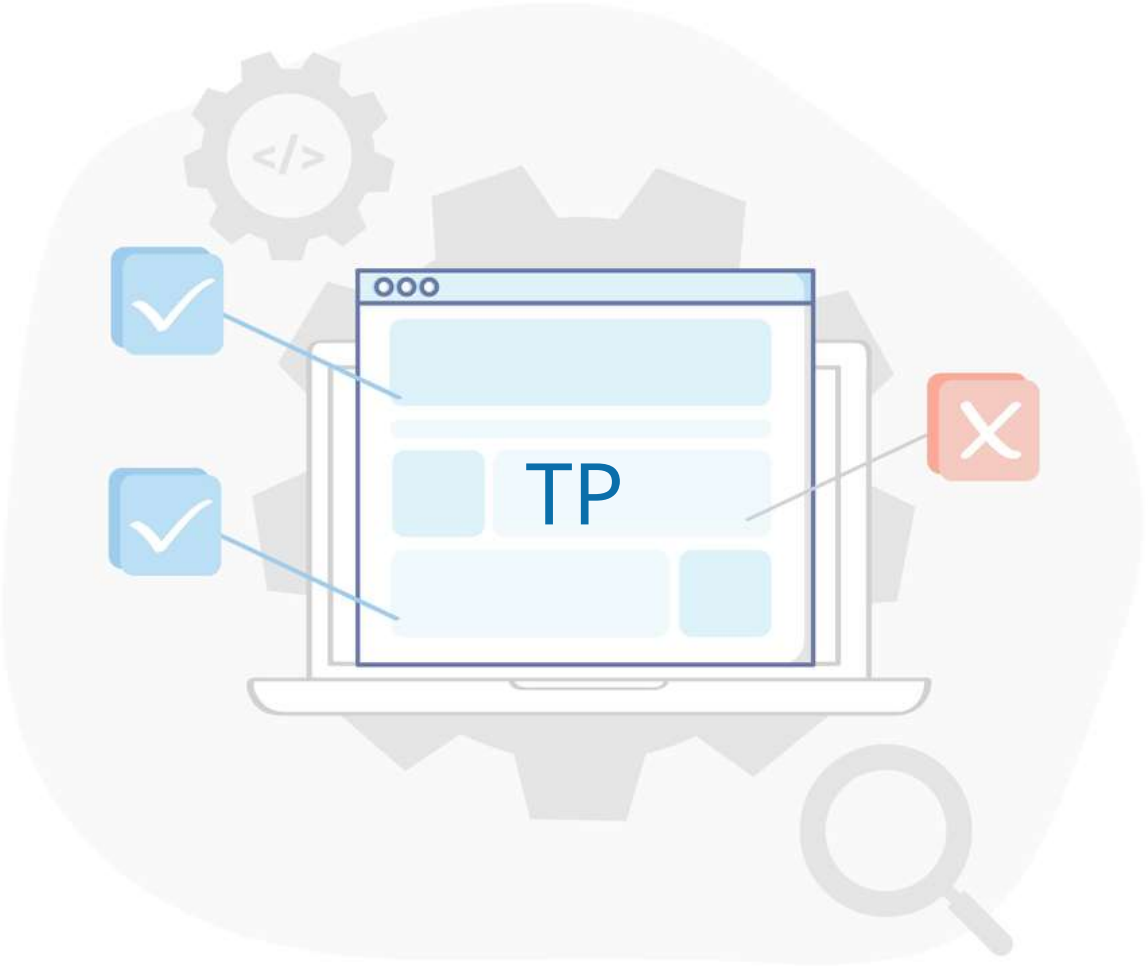
OPTIONS AVANCÉES

- **Criticité** : niveaux définis librement + action pour la faire évoluer. On peut créer des actions changement de statut « criticité + » / « criticité - »
- **Variables de workflow** : leur valeur sera définie au niveau de chaque formulaire
  - Très pratique pour faire varier des éléments d'un formulaire à l'autre tout en gardant un WF générique (par exemple, signature du service) ou ajuster le workflow d'année en année (par exemple, date de clôture)
- **Actions globales** : des actions de workflows automatiquement disponibles pour tous les statuts.
  - Permettent aussi d'agir sur le listing des demandes
- **Données de traitement** : champs qui viennent se greffer aux demandes. Par exemple : quartier, choisi par un agent ou valeur calculée sur la base de plusieurs champs

ASSOCIER WORKFLOWS ET FORMULAIRES

- Depuis le formulaire, cliquer sur « Changer » en face de « workflow » et sélectionner le nouveau workflow
- S'il y a déjà des demandes remplies pour ce formulaire, il faut faire correspondre les statuts de l'ancien workflow et du nouveau

entr'ouvert



TP WORKFLOW

TP Workflow 1

- **Dupliquez** le workflow « Workflow TP1 », renommez-le avec votre prénom : « Workflow TP1 - Théodule » et mettez-le dans la catégorie « Formation »
- Envoyez un courriel à l'usager pour le prévenir que sa demande a bien été reçue.

- Pour tester : importez/réutilisez le formulaire de contact, branchez-le sur votre workflow
  - Renommez le formulaire de contact avec votre nom dedans
  - Mettez-le dans la catégorie « Formation »
  - Remplissez la démarche jusqu'au bout et validez

TP WORKFLOW

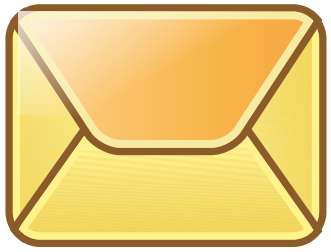
TP Workflow 2

- Reprenez le workflow précédent
- L'objectif est de pouvoir refuser (rejeter) la demande.
- Ajoutez un statut « Rejeté » dans lequel vous envoyez un courriel à l'usager pour le prévenir que sa demande est rejetée
- À partir du statut « Nouveau », permettez à l'agent de rejeter la demande
- Testez !
  - Sur le formulaire, pensez bien à renseigner le destinataire » avec un rôle que vous avez
  - Remplissez la démarche jusqu'au bout et validez
  - Allez sur le listing, traitez (rejetez) une demande

TP Workflow 3

- Reprenez le workflow précédent
- L’objectif est d’ajouter une étape de validation : l'agent accepte une demande, son chef de service doit valider le choix de l’agent avant que le traitement soit effectif. Le chef peut rejeter la demande.
- Il faut ajouter une fonction chef de service à laquelle on donnera un rôle dans le formulaire

entr'ouvert



LES COURRIELS

COURRIELS

Documentation :

https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-de-workflows/les-actions-de-workflow/elements_envoyer-un-email/

L’envoi d’un mail à un usager

Lorsque l'action est paramétrée pour envoyer un courriel à l'usager, c'est-à-dire au demandeur de la démarche, l'adresse de destination utilisée est :

- soit le courriel saisi dans le formulaire, dans le premier champ « courriel » avec pré-remplissage depuis le champ utilisateur courriel ;
- soit le courriel renseigné dans le profil du compte si le champ n'est pas renseigné.

Il est donc important de prévoir un champ « courriel » avec pré-remplissage pour les formulaires pouvant être complétés sans compte usager.

Mise en page des mails

Il ne faut pas utiliser de code HTML dans les gabarits de Mails

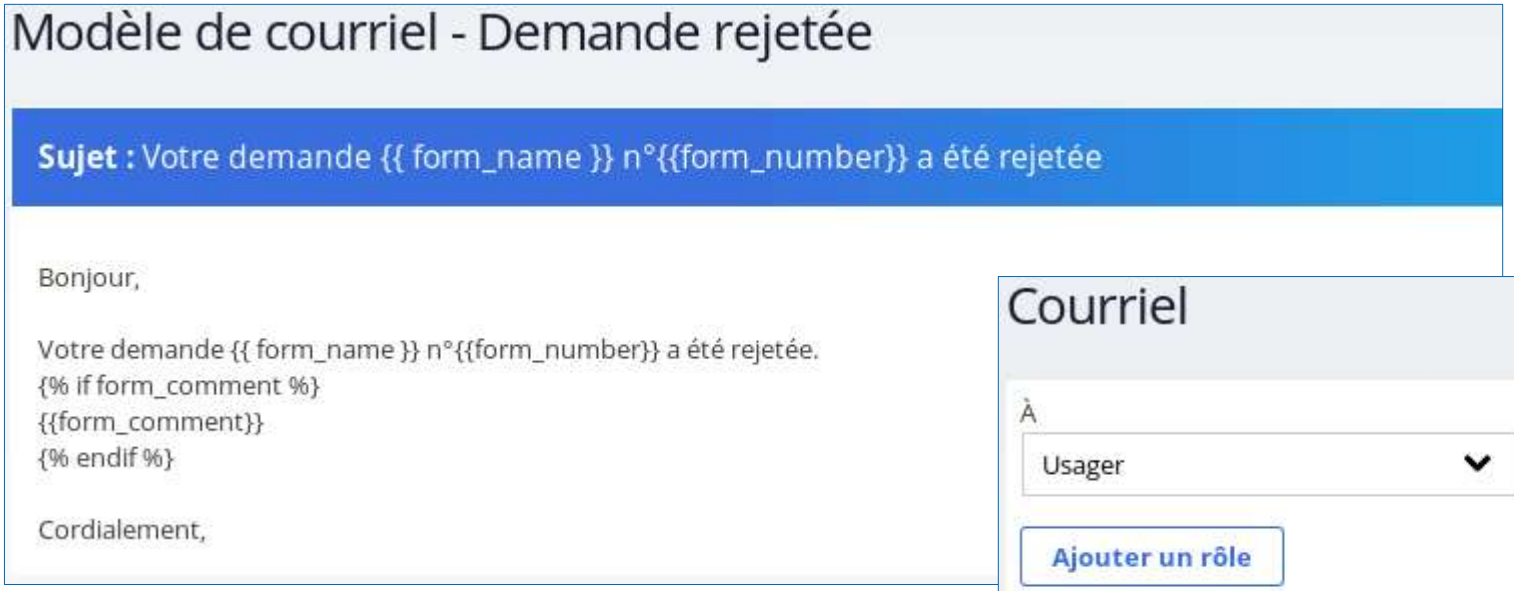
La page de documentation suivante permet de comprendre :

- comment gérer les mise en page sur les mails (gras, italique, ...)
- gérer les sauts de ligne
- comment insérer des fichiers attachés

https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-de-workflows/les-actions-de-workflow/elements_envoyer-un-email/#optimiser-la-mise-en-forme-du-message

MODÈLES DE COURRIEL

- Pour créer des emails types, utilisables dans l'action d'envoi de courriel.



TP WORKFLOW

TP Workflow 3

- L’objectif est de transmettre dans un mail, un commentaire saisi par l’agent traitant en cours de traitement
- Il faut donner un nom de variable « xxxx » à une action «commentaire »
- Utiliser cette variable sous la forme {{ comment_xxxx }} dans le template du mail envoyé
- Vérifiez le mail reçu

entr'ouvert

CRÉATION DE DOCUMENT

CRÉATION DE DOCUMENT

Documentation :

https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-de-workflows/les-actions-de-workflow/elements_creer-document/

- Créer un modèle
  - utilisation des variables de formulaire
- Importer le modèle
- Revue des options :
  - PDF
  - inclure dans l'historique
  - importer dans porte-document
  - identifiant =/nom du fichier
- Méthode : interactive / non interactive

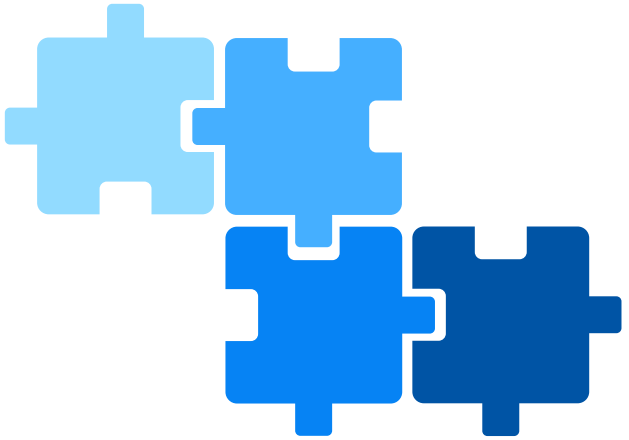
TP WORKFLOW

TP Workflow 4

- L’objectif est de générer un document reprenant les informations du formulaire
- Partir du modèle par défaut (si existant)
- Ajouter les variables du formulaire
- Générer un pdf
- Joindre à l’historique
- Vérifiez le document généré
- Bonus : envoyer par courriel

entr'ouvert

FORMULAIRE DE WORKFLOW



FORMULAIRE DE WORKFLOW

Documentation :

https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-de-workflows/les-actions-de-workflow/elements_afficher-un-formulaire/

Pour demander des informations supplémentaires ou pour compléter la demande avec des informations internes

- pointer et ajouter chaque fonction et/ou rôle d'utilisateur destiné à saisir le formulaire
- attribuer un identifiant au formulaire à afficher : les variables seront ainsi récupérables et utilisables (courriels, documents générés…)
- le formulaire est semblable à un formulaire « front », mais il n'est accessible qu'au cours du traitement, quand la demande arrive sur le statut où il est défini ; il ne peut comporter qu'une seule page

entr'ouvert

LA GESTION DES RÔLES

UTILISATEURS, RÔLES ET DROITS

- Un utilisateur correspond à un compte
  - un compte peut être créé en ligne (cas classique pour les usagers)
  - un compte peut être créé via synchronisation avec un annuaire LDAP (cas classique pour les agents)
  - un compte peut être créé directement dans le back-office, menu « Utilisateurs »
  - un compte peut être créé à la volée, en mode multi-canal
- Un compte peut avoir
  - 0 rôle (cas habituel pour les usagers)
  - 1 rôle. Exemples : « agent », « chef de service état-civil » pour un compte agent ; « parent » pour un compte usager
  - plusieurs rôles. Exemples : « agent », « agent accueil » et « agent état-civil »
- Un rôle peut inclure d'autres rôles
  - cas classique : « chef de service état-civil » a automatiquement le rôle « agent état-civil » qui lui-même hérite automatiquement du rôle « agent »

UTILISATEURS (COMPTES) ET RÔLES

La gestion des rôles

- Accessibles en gestion, lorsqu'on a les bonnes permissions (rôles de gestion des rôles et utilisateurs), depuis le menu latéral en backoffice.
- Import des utilisateurs
- Différence entre :
  - Rôles internes qui définissent des droits dans l'application.
  - Rôles créés pour chaque plate-forme (agrégation de rôles internes ou rôles purement métiers) qui définissent ce qu'un utilisateur peut faire à chaque étape du circuit de traitement mais également peut voir / accéder à ...

LES TYPES DE RÔLES

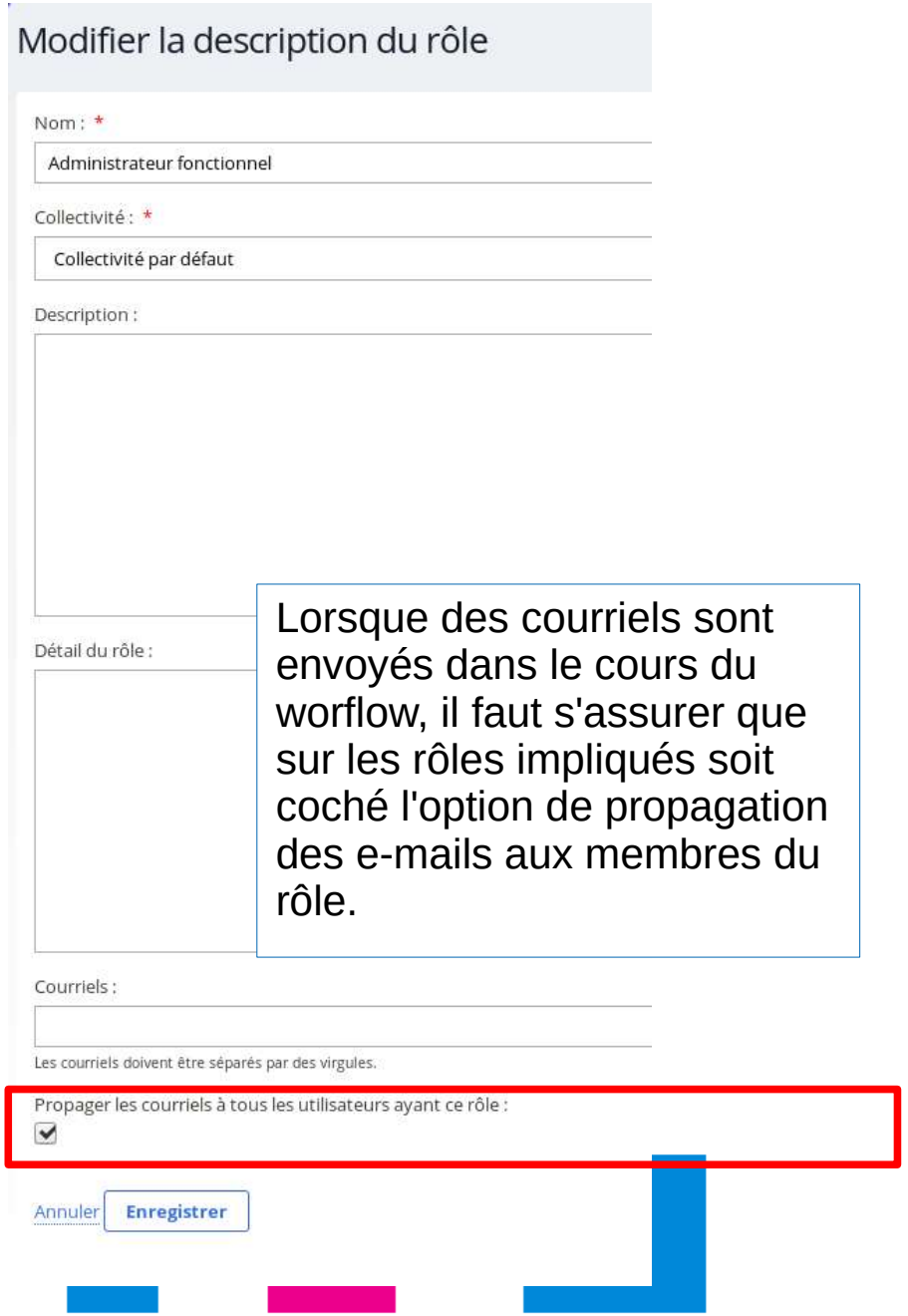
Les rôles internes Publik

- Administrateur des rôles
- Administrateur des utilisateurs
- Administrateur de Démarches
- Administrateur de "Compte citoyen"
- Administrateur de "Portail agent"
- Administrateur de Passerelle (services web)
- ...

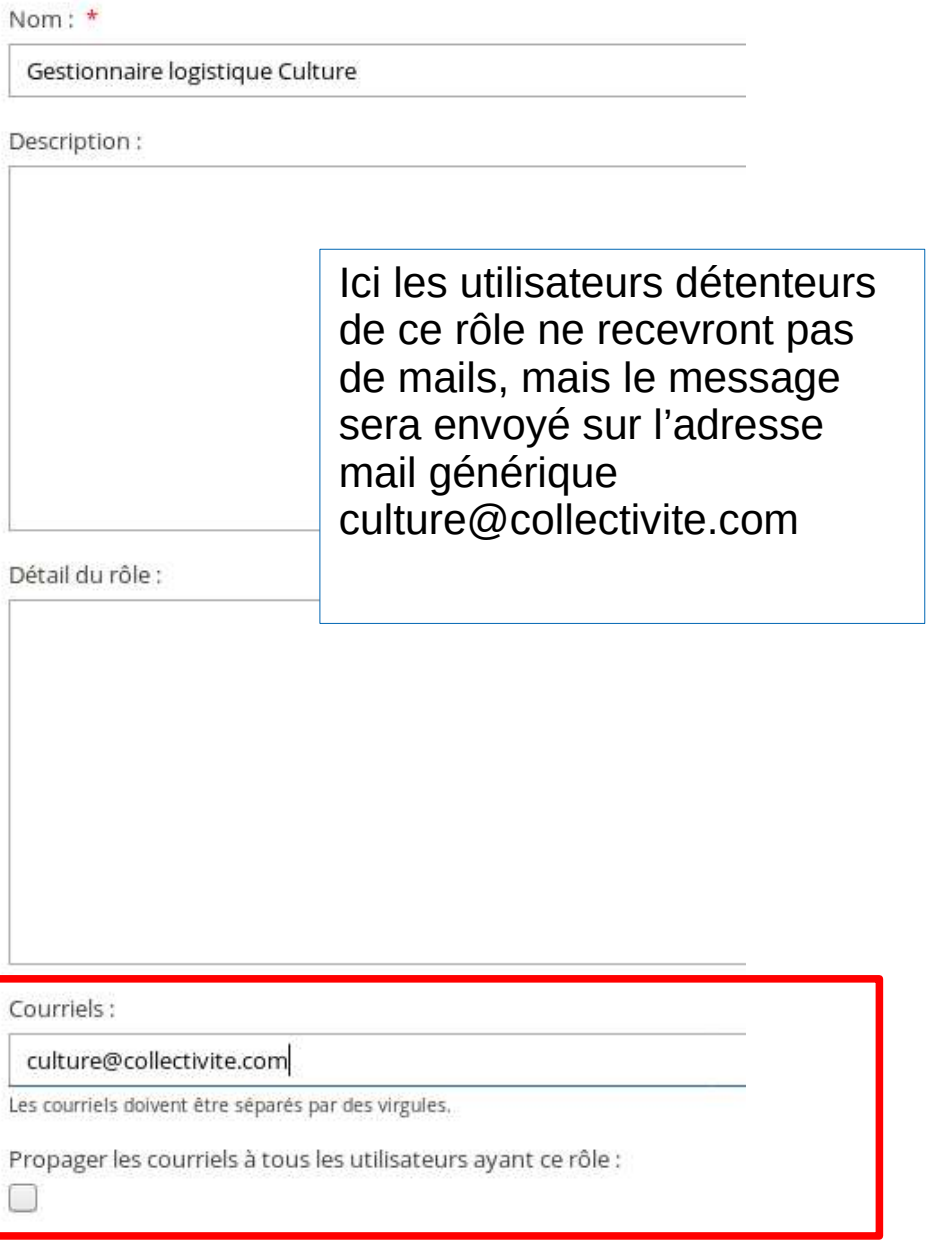
Les rôles métiers (construits)

- "Agent" : donne accès au portail agent
- "Administrateur fonctionnel" : rôle pour administrer la plate-forme au quotidien (utilisateurs, rôles, démarches, portails)
- Les rôles « service », par exemple : "Gestionnaire voirie" qui pourra traiter les demandes du domaine
- Il est possible de créer des rôles composés uniquement d'agrégation d'autres rôles, sans droits spécifiques à ce dernier ; exemple : agent traitant de toutes les demandes pouvant être complétées au guichet

FONCTIONS, RÔLES ET UTILISATEURS



Lorsque des courriels sont envoyés dans le cours du worflow, il faut s'assurer que sur les rôles impliqués soit coché l'option de propagation des e-mails aux membres du rôle.

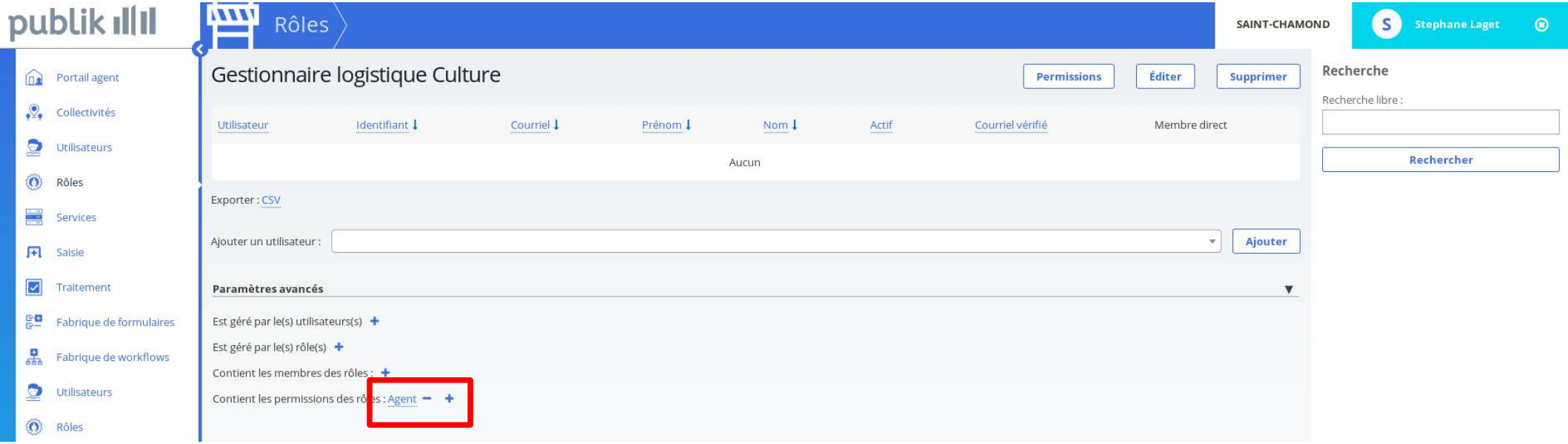


Ici les utilisateurs détenteurs de ce rôle ne recevront pas de mails, mais le message sera envoyé sur l'adresse mail générique culture@collectivite.com

UTILISATEURS (COMPTES) ET RÔLES

Le rôle agent

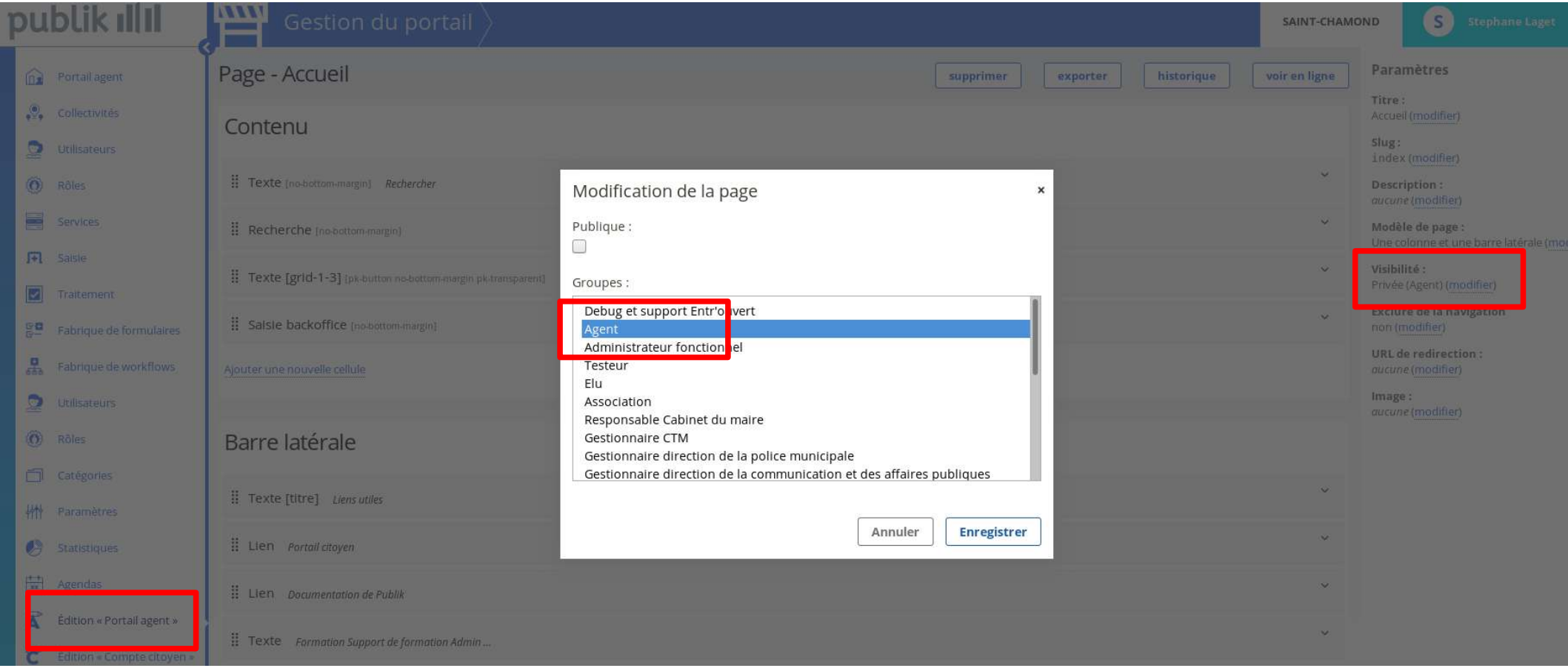
- En pratique, il convient de limiter l'accès au portail agent aux utilisateurs identifiés qui ont un rôle « Agent »
  - Créer un rôle Agent s'il n'existe pas
  - A chaque fois que l'on crée un nouveau rôle ayant vocation à aller sur le BO, lui attribuer le rôle agent



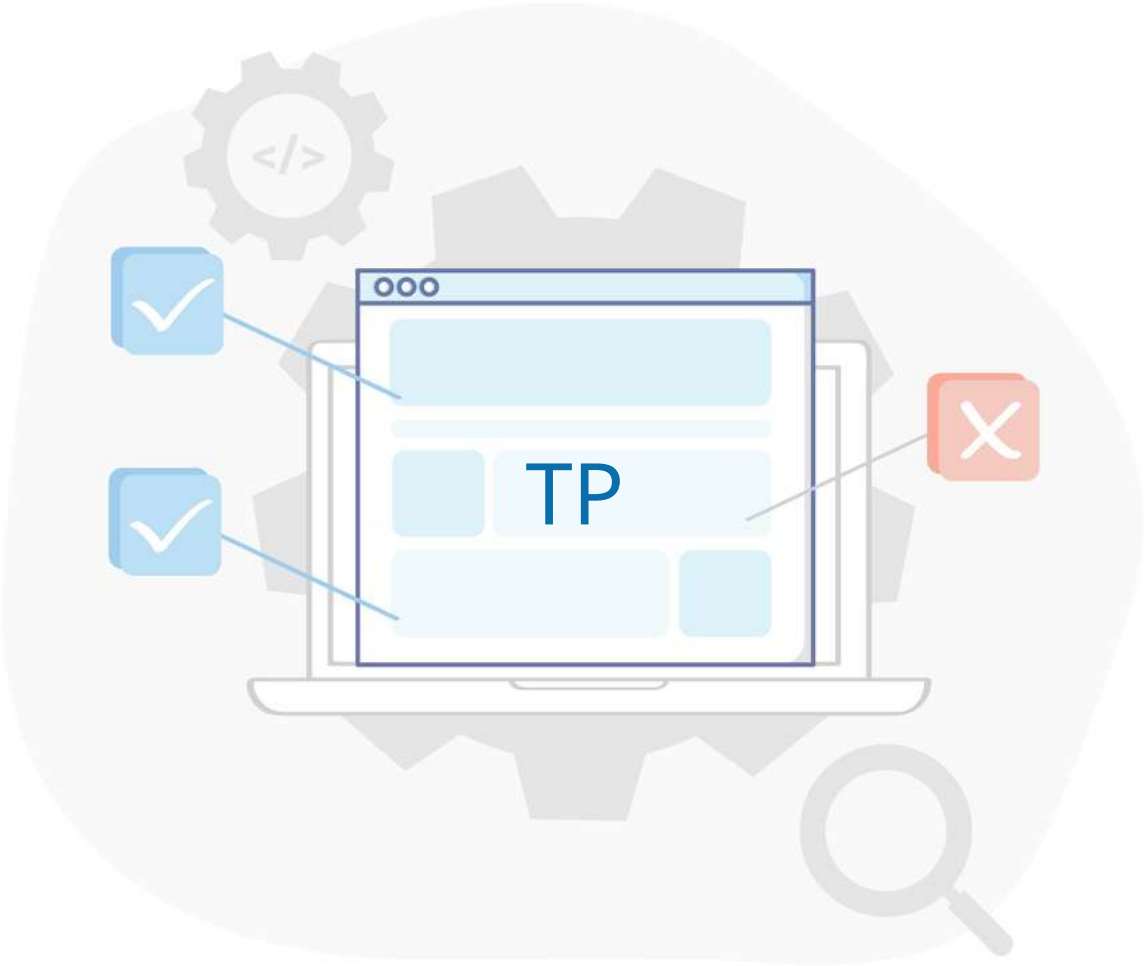
UTILISATEURS (COMPTES) ET RÔLES

Le rôle agent

- Et dans l'édition du portail agent, restreindre l'accès de la page d'accueil aux utilisateurs identifiés qui ont un rôle agent



entr'ouvert



UTILISATEURS (COMPTES) ET RÔLES

TP 1

- Créer les rôles qui seront utilisés :
  - Héritage du rôle Agent
  - Paramétrage des permissions

UTILISATEURS (COMPTES) ET RÔLES

TP 2

- Créer un utilisateur suivant votre charte de nommage (LDAP interne) puis la respecter
- Générer un mot de passe automatiquement.
- Donner à cet agent les droits de gérer les utilisateurs, les rôles et les téléservices. Donner à cet agent l'accès au portail.
- Tester dans un autre navigateur ou en navigation privée.
- Supprimer le compte

entr'ouvert

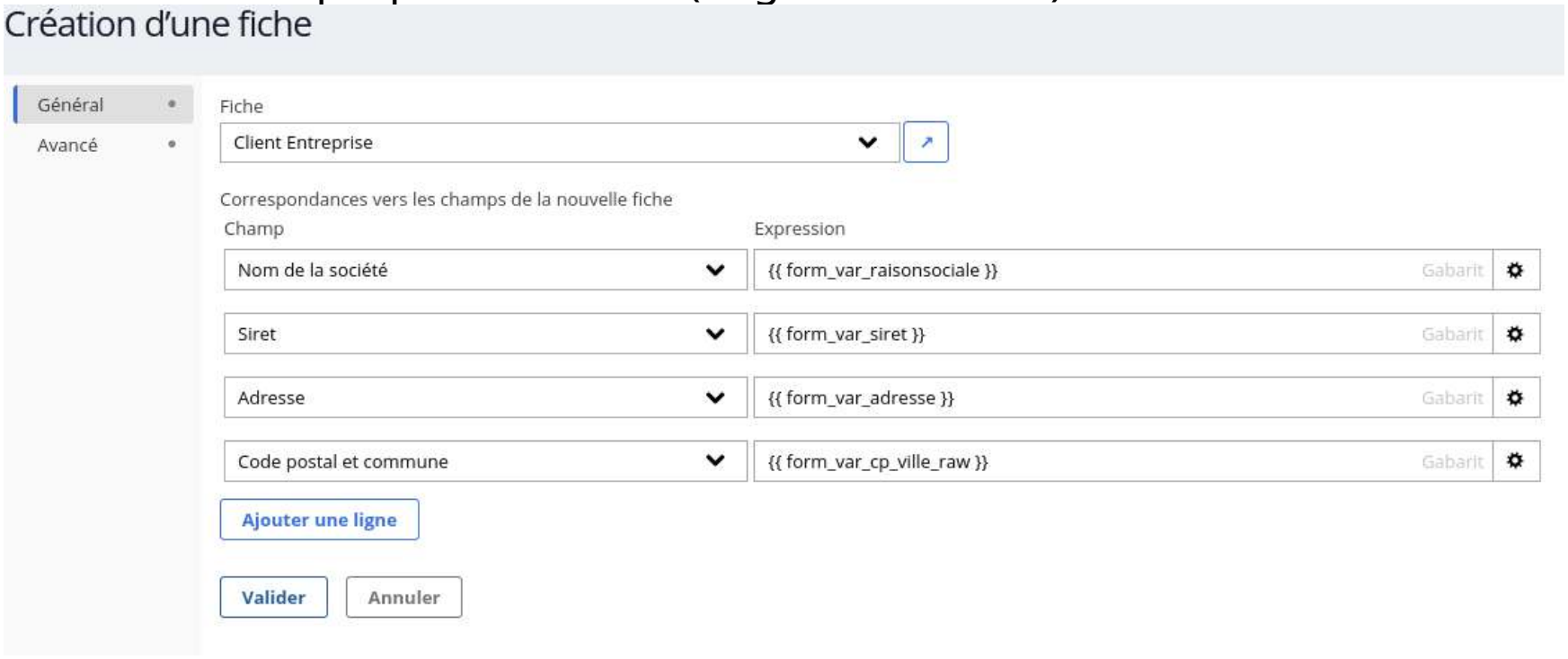
STUDIO : LES FICHES

LES FICHES

- Les fiches permettent d'organiser les données de manière structurée.
  - Par exemple : établissements scolaires, ressources…
- Les fiches peuvent aussi servir à « étendre » les informations stockées sur l'usager (enfants, association…)
- La conception d'un modèle de fiche est très proche de celle d'un formulaire : ajout de champs, organisation…
- Une fiche suit un workflow
  - Notion de fonctions/rôles
  - Actions de workflow
- Ajout de contenu :
  - dans le backoffice, ajout de fiches une à une
  - dans le backoffice, par import
  - via une démarche, avec l'action de workflow « création d'une fiche »

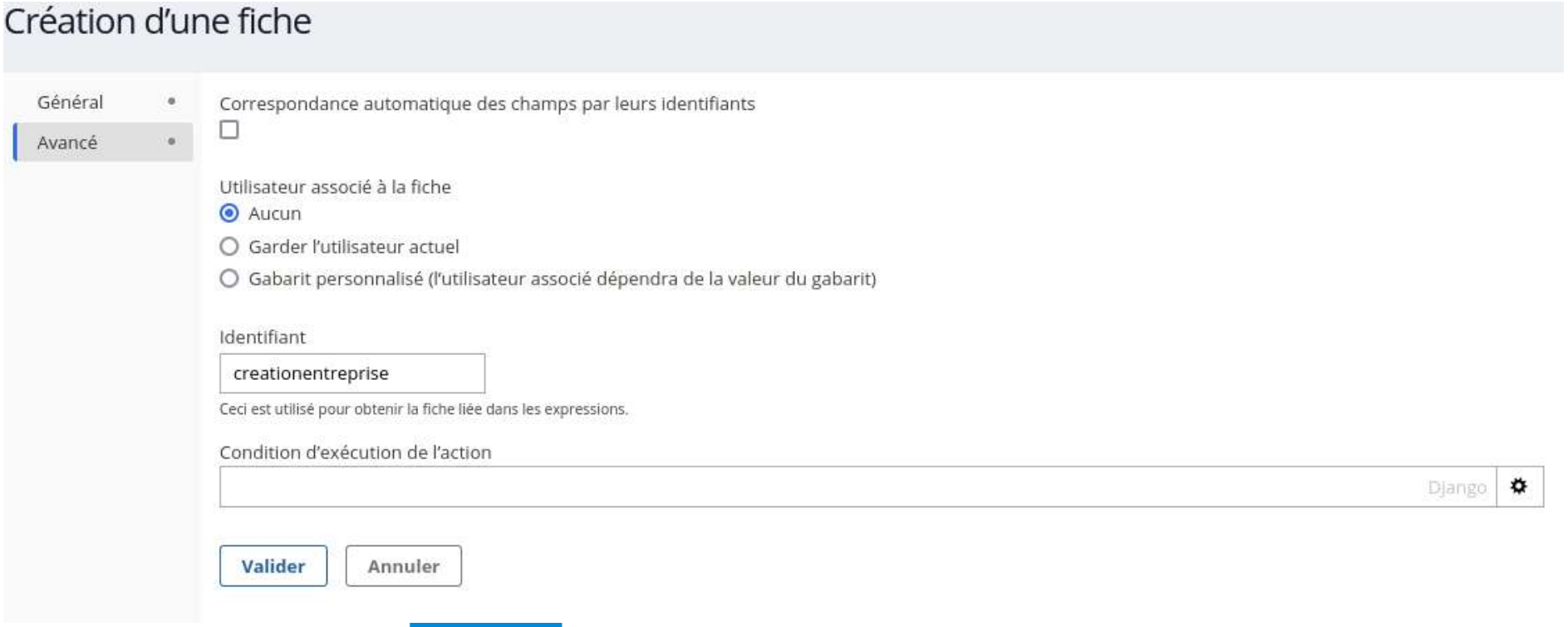
LES FICHES : CRÉER, MODIFIER

- Action de workflow « création d'une fiche »
  - Choix du modèle de fiche
  - Puis : correspondance des champs un par un ou correspondance automatique par identifiant (onglet « avancé »)



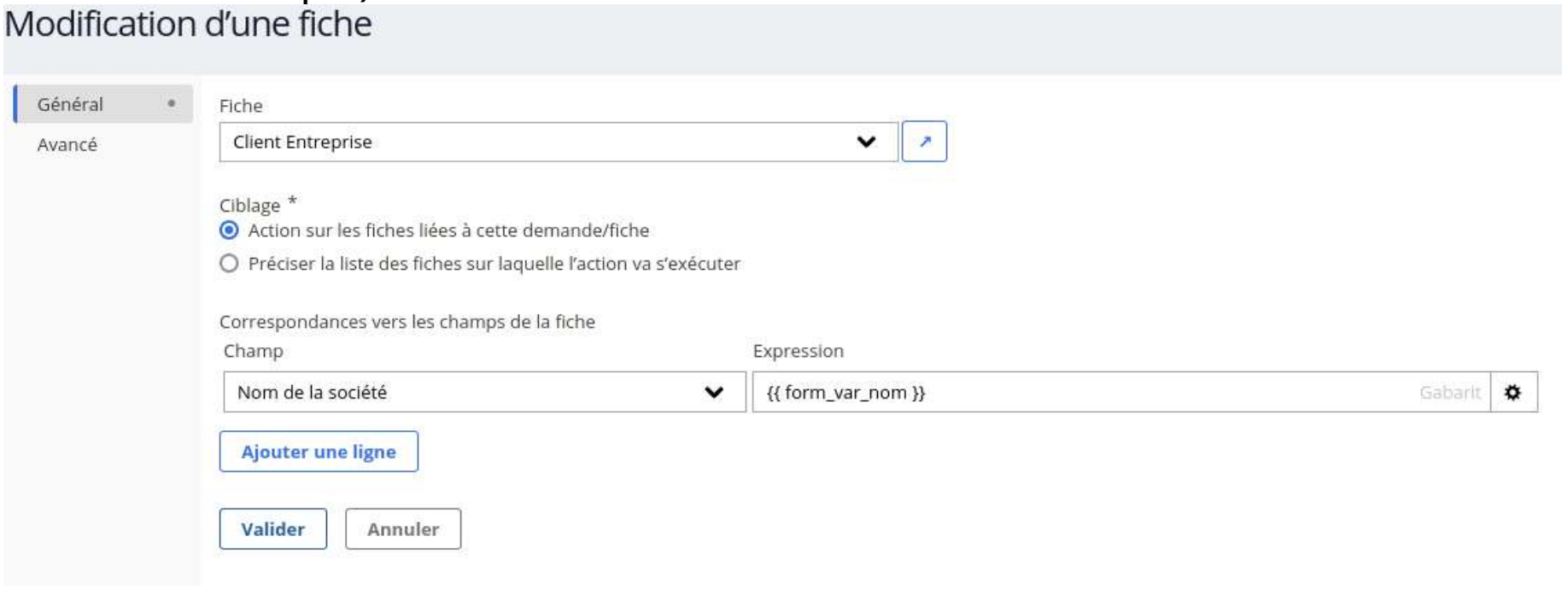
LES FICHES : CRÉER, MODIFIER

- Possibilité de :
  - Lier la fiche à l'utilisateur lié à la demande
  - Lier la fiche à un utilisateur autre
  - Créer une fiche indépendante de tout utilisateur



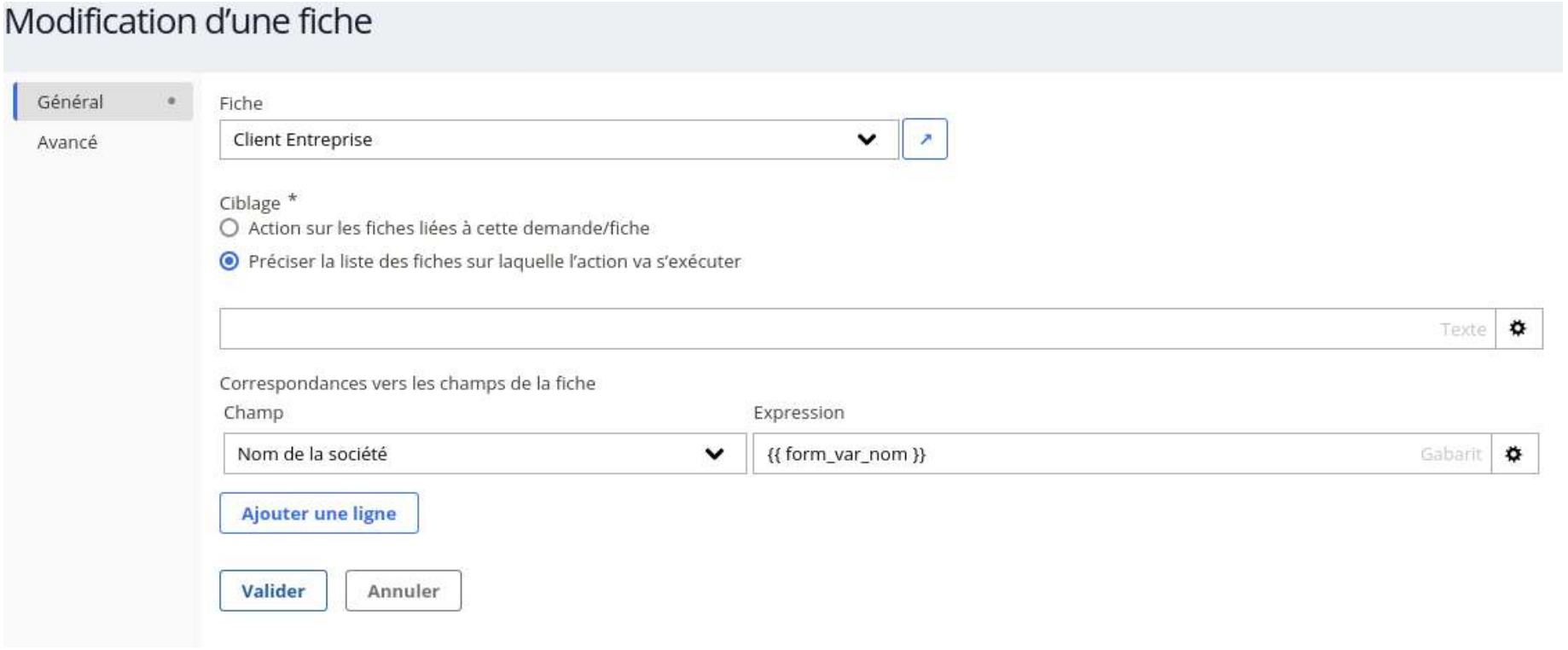
LES FICHES : CRÉER, MODIFIER

- Action de workflow « modification d'une fiche »
  - Choix du modèle de fiche
  - Puis : correspondance des champs un par un (pas de correspondance automatique)



LES FICHES : CRÉER, MODIFIER

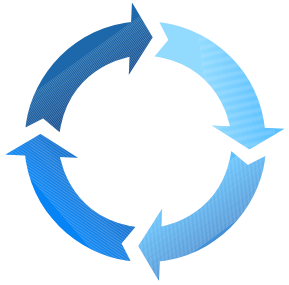
- Modification d'une fiche : possibilité d'indiquer une liste des fiches (identifiants à saisir « en dur » ou récupérés via un filtre de requête)



LES FICHES

- Utilisation d'une fiche comme référentiel/source de données
  - renseigner l'option « Gabarit du résumé » dans le modèle de fiche concerné : la fiche devient automatiquement une source de données disponible dans les formulaires
  - possibilité de créer des vues personnalisées dans le listing et de les enregistrer comme sources de données
- Utilisation des fiches dans le portail Agents
  - Liste des fiches
  - Recherche sur les fiches
  - Renvoi vers la fiche
- Utilisation des fiches dans le portail Citoyens
  - Liste des fiches
  - Recherche sur les fiches
  - Contenu d'une fiche

entr'ouvert



COMMENT TESTER ?

COMMENT TESTER ?

- Avoir autant de comptes que de rôles intervenants dans le workflow, chaque compte doit avoir uniquement un des rôles du workflow
- Créer les rôles et attribuer les rôles aux comptes.
- Utiliser plusieurs navigateurs (ou une session privée du navigateur) pour voir simultanément les écrans des différents utilisateurs.
- À chaque étape, vérifier ce que chaque utilisateur est en mesure de faire et les mails qui sont envoyés.
- Si ça ne marche pas :
  - l'inspect
  - la page globale des erreurs

entr'ouvert

BONNES PRATIQUES ET ÉVOLUTIONS

- Pratiques visant à assurer que votre démarche en ligne soit facile à maintenir et à faire évoluer.
  - faire des WF simples.
  - utiliser le bon type de champs (liste à choix multiple plutôt que plusieurs cases à cocher successives par exemple).
  - ordonner les actions des statuts pour faciliter la compréhension.
  - mettre une description soignée dès que cela est possible.
  - donner des identifiants de champs explicites, cohérence (intitulés composés) pour avoir des variables compréhensibles et avec une charte de nommage cohérente
    - OK : commune, id_procedure, date_naissance… qui sont lisibles et mémorisables (form_var_commune, form_var_id_procedure, form_var_date_naissance…)
    - KO : dst, g23…

entr'ouvert

AGENDA

AGENDA

- 3 types d'agendas : « **Événements** », « R**endez-vous** » et « **Virtuel** »
  - on crée un agenda en choisissant son type, avec un rôle pour l'édition (« qui » peut modifier l'agenda ?) et un rôle pour la visualisation (« qui » peut voir l'agenda ?)
  - ajustement des délais de réservation (nombre de jours minimal, borne incluse, et un nombre de jours maximal, borne exclue, pour encadrer la période de réservation)
  - possibilité de ranger dans des catégories
  - on utilise ensuite les agendas dans les formulaires et les workflows
- « **Rappels de réservation** » (rappel avant le rendez-vous ou l'événement)
  - user_email et/ou user_phone_number contenant respectivement le courriel / le numéro de mobile de l'usager, ainsi que form_url contenant l'URL de la demande, typiquement {{ form_url }}
- Documentation complète : https://doc-publik.entrouvert.com/admin-fonctionnel/prises-de-rendez-vous/

AGENDA

- Type d’agendas : rendez-vous
  - Notion de guichet
  - Notion de type de rendez-vous : durée d’un rendez-vous
  - On crée une ou plusieurs durées de rendez-vous
  - On crée une ou plusieurs plages horaires pour les rendez-vous

  - Le paramétrage d’un guichet peut être dupliqué sur d’autres guichets
  - Les exceptions peuvent être définies pour tous les guichets
  - Possibilité de mettre des couleurs

AGENDA

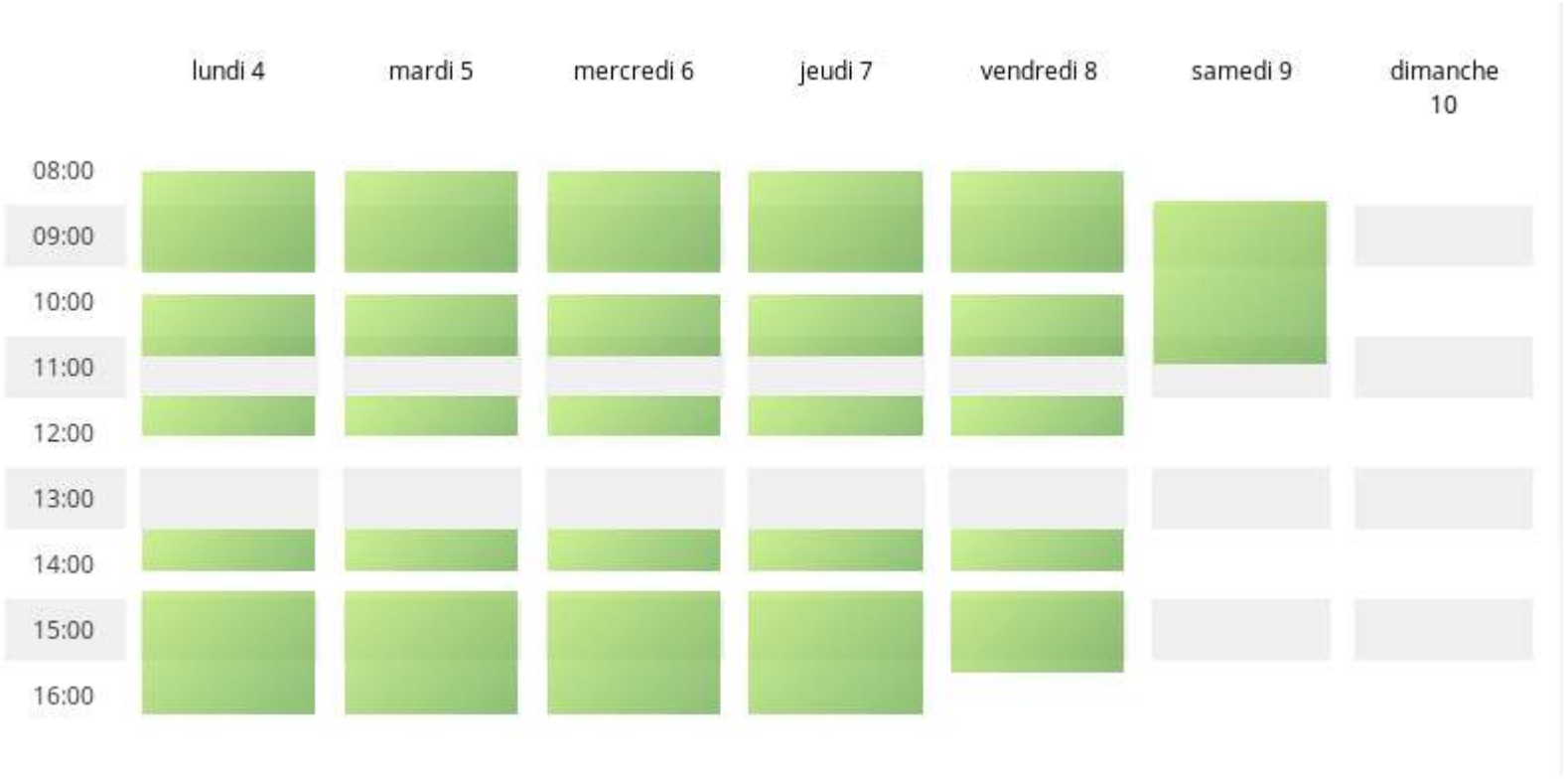
- Visualisation front-office

Date du RV *

<	mercredi 20 mars 2019	jeudi 21 mars 2019	vendredi 22 mars 2019	lundi 25 mars 2019	mardi 26 mars 2019	>
	09:00	10:00	10:00	09:00	09:00	
	09:20	10:20	10:20	09:20	09:20	
	09:40	10:40	10:40	09:40	09:40	
	10:00	11:00	11:00	10:00	10:00	
	10:20	11:20	11:20	10:20	10:20	
	10:40	11:40	11:40	10:40	10:40	
	11:00	14:00	14:00	11:00	11:00	
	11:20	14:20	14:20	11:20	11:20	
	11:40	14:40	14:40	11:40	11:40	
	12:00	15:00	15:00	14:00	14:00	
	12:20	15:20	15:20	14:20	14:20	
	12:40	15:40	15:40	14:40	14:40	
	13:00	16:00	16:00	15:00	15:00	
	13:20	16:20	16:20	15:20	15:20	
	13:40	16:40	16:40	15:40	15:40	
	14:00	17:00	17:00	16:00	16:00	

AGENDA

- Visualisation back-office



AGENDA

- Réservation de ressources
  - avec le type « rendez-vous »
  - permet de réserver une ressource (voiture, vidéoprojecteur, salle de réunion…) sur une durée paramétrée

AGENDA VIRTUEL

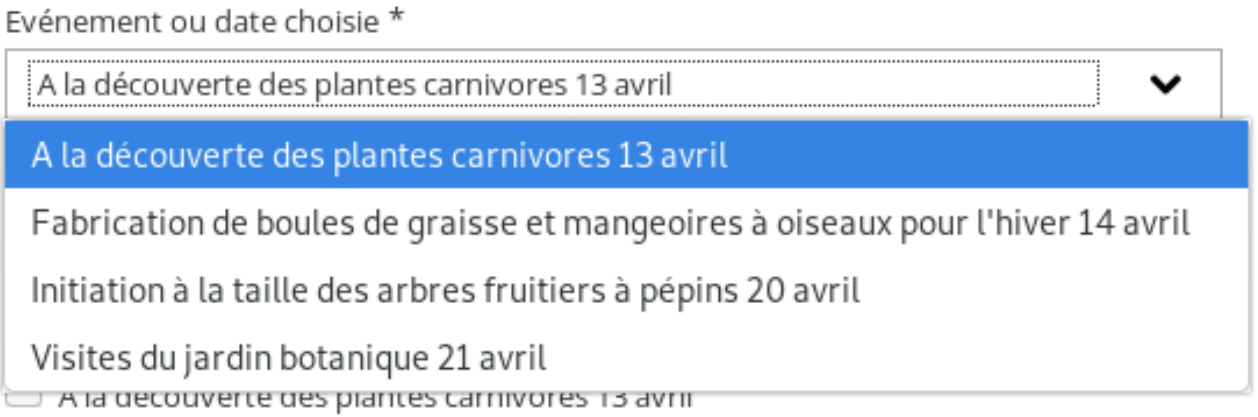
- Agrège plusieurs agendas RdV
  - les agendas agrégés doivent tous avoir les mêmes types de rendez-vous i.e. même durée, même identifiant et même libellé.
- Cas d'usages
  - Proposer les premiers créneaux disponibles parmi l'ensemble des agendas inclus dans l'agenda virtuel : si vous avez plusieurs agendas gérant chacun les rendez-vous d'un(e) assistant(e) sociale, passer par un agenda virtuel qui inclut tous ces agendas, permettra à l'usager de prendre rendez-vous avec n'importe quel assistant(e) sociale, sur le créneau qui l'arrange
  - Réservez des créneaux uniquement pour enregistrement par les agents. Sur un agenda « rendez-vous état civil », vous pouvez créer un agenda virtuel qui inclut cet agenda et y définir des périodes d'exclusion ; si vous exposez l'agenda initial « rendez-vous état civil » aux agents, ceux-ci auront accès à l'intégralité des créneaux et en exposant l'agenda virtuel dans un formulaire en front office i.e. pour les usagers, ceux-ci ne pourront pas réserver sur les créneaux dans les périodes d'exclusion.

AGENDA

- Type d’agendas : évènements
  - Un ou plusieurs événements
  - Nombre de places
  - Liste d’attente
  - Possibilité de créer des événements récurrents, hybridation entre les rendez-vous et les événements. La récurrence se configure dans les options de l'événement et offre la possibilité de définir des événements répétés tous les jours, ou tous les jours ouvrés, ou toutes les semaines, ou toutes les deux semaines.
  - Écran de pointage
    - affiche la liste des inscrits et permet d'indiquer, pour chacun d'eux, s'il a finalement participé ou non
    - sur la page paramètre de l’agenda : gabarit pour configurer le listing de pointage
    - des filtres pour limiter l’affichage (nécessite des catégories créées au moment de l’appel webservice (exemple régime : vegetarien, régime: allergique)

AGENDA

- Visualisation front-office



AGENDA

- Visualisation back-office

AGENDA

- Utilisation dans les formulaires : sources de données automatiquement créées et disponibles dans les champs liste
- Utilisation dans les workflows, appel post sur {{form_var_event_api_fillslot_url}}
  - label pour le libellé qui sera affiché dans les plages réservées dans les agendas, vue back-office (par exemple pourrait être {{form_name}},
  - user_first_name et user_last_name pour le prénom et le nom de l'usager, si défini viendra s'afficher dans la plage réservée à la suite de label,
  - backoffice_url pour la demande vers laquelle créer un lien activé sur le libellé précédent (usuellement doit être {{form_url_backoffice}}) (uniquement pour fichier ics)
  - user_display_label pour le libellé qui sera présenté à l'usager (par exemple « Rendez-vous passeport ») (uniquement pour fichier ics)
  - cancel_callback_url, optionnel https://doc-publik.entrouvert.com/admin-fonctionnel/prises-de-rendez-vous/annulation-dans-un-agenda/
  - use_color_for
  - Couple clé : valeur (exemple regime)

AGENDA

- Appels webservices supplémentaires pour un agenda de type événement
  - Nombre de places disponible : {{agendas_url}}api/agenda/REUNIONS-D-INFORMATION/status/{{form_var_event_raw}}/
    - si reservations est l'identifiant de l'appel. Places dispos : {{webservice.RESERVATIONS.places.available}}
  - Liste d'attente : le retour d'une réservation peut contenir un attribut in_waiting_list à True. Pour changer de statut vers un statut d'attente, une condition sur reservation_response_in_waiting_list peut être posée.
  - Réserver plusieurs places : {{agendas_url}}api/agenda/REUNIONS-D-INFORMATION/fillslot/{{form_var_event_raw}}/?count={{form_var_nb_places}}
  - Réservation de plusieurs événements possible mais complexe :
    - dans le formulaire, utiliser un champ de type liste à choix multiple plutôt qu'un champ de type liste
    - https://doc-publik.entrouvert.com/admin-fonctionnel/prises-de-rendez-vous/enregistrement-dans-agenda/#reservation-multiple-agendas-de-type-evenements

GESTION DU PORTAIL AGENT ET USAGER (CMS COMBO)

CMS COMBO

- Conçu spécifiquement pour fonctionner avec les formulaires et catégories, les démarches, le code de suivi....
- Pas de chaîne éditoriale : validez, c'est publié
- Templates (modèles) de pages permettent d'avoir une ou plusieurs colonnes.
  - possibilité de créer des modèles de page
- Cellules : blocs d'information couvrant les besoins classiques pour un portail de démarches en ligne
- Possibilité de n'afficher certaines cellules/pages qu'à certains rôles
- On hiérarchise les pages avec un simple glisser/déplacer

LES BASES

- Ajouter une page
- Exporter un site ou une page
- Importer et les slugs
- Historique de modification d’une page
- Voir en ligne
- Renommer une page
- Changer son slug
- Template de page
- Visibilité & menu
- Redirection de page

STRUCTURE

- Le menu racine : 2 premiers niveaux
  - Option exclure de la navigation
- Les slugs des pages et les URL
  - La racine : page avec l'URL index
- La hiérarchie des pages :
  - Construction des URL
  - Héritage d'une « zone »
  - Construction de menus
- Dupliquer une page / ajouter une sous-page
- Navigation : page précédente/suivante/parent

CMS COMBO

- Différentes cellules
  - Texte
  - Lien / Liste de liens
  - Menu
  - Carte
  - Fiches : liste et contenu d'une fiche
  - Une démarche (Lien vers une démarche)
  - Mes démarches (en cours, brouillon, terminées)
  - Toutes les démarches d'une catégorie
  - Liste des catégories
  - Saisie back-office
  - Recherche
  - Notifications à l'usager
  - Demandes à traiter
  - *Identique au parent*
- On déplace les cellules dans les bonnes zones et dans le bon ordre par glisser/déplacer

CONTENU

- Ajout de cellule.
  - slug : identifiant de la cellule
  - permet de définir une ancre pour pointer avec une URL sur le contenu de la page
  - visibilité
- Cellule texte
  - wysiwyg
  - source
  - ajout classes css :
    - https://doc-publik.entrouvert.com/admin-fonctionnel/modifier-le-contenu-des-portails/classes-css/
    - Foldable : cellule texte qui commence par un h2

CELLULE RECHERCHE

- Index mis à jour toutes les 24 heures
- Contenu indexé :
  - Pages, en se basant sur le titre de la page et le texte des cellules.
  - Liens externes comme les démarches, ici aussi le titre et le texte formé de la description et des mots-clés.
  - les titres sont pondérés (1,5).
  - la recherche se fait sur sous-chaînes exactes.
  - outils utilisés :
    - Haystack : http://django-haystack.readthedocs.io
    - Whoosh : https://bitbucket.org/mchaput/whoosh/wiki/Home

CARTOGRAPHIE

Modification de la couche cartographique

Libellé :

Parkings Quimper

Identifiant :

parkings-quimper

URL GeoJSON :

https://places-parkings.quimper-communaute.bzh/WebServiceInfosGeolocFront.asmx/GetGeoJsonListDysplayItemByType?type=parkings

Couleur du marqueur :

Icône du marqueur :

---------	Bâtiment	Cloche	Empreinte	Livre	Pinceau	Taxi
Ambulance	Bus	Coche	Étoile	Maison	Poubelle	Train
Ampoule	Camion	Cube	Goutte	Marteau	Recyclage	Université
Arbre	Chaîne cassée	Douche	Hôpital	Métro	Route	Vélo
Astérisque	Chaise roulante	Drapeau	Œil	Moto	Signalisation	Voiture
Avertissement						

Couleur de l'icône :

Durée de cache :

60

Inclure l'identification de l'utilisateur dans la requête :

Propriétés :

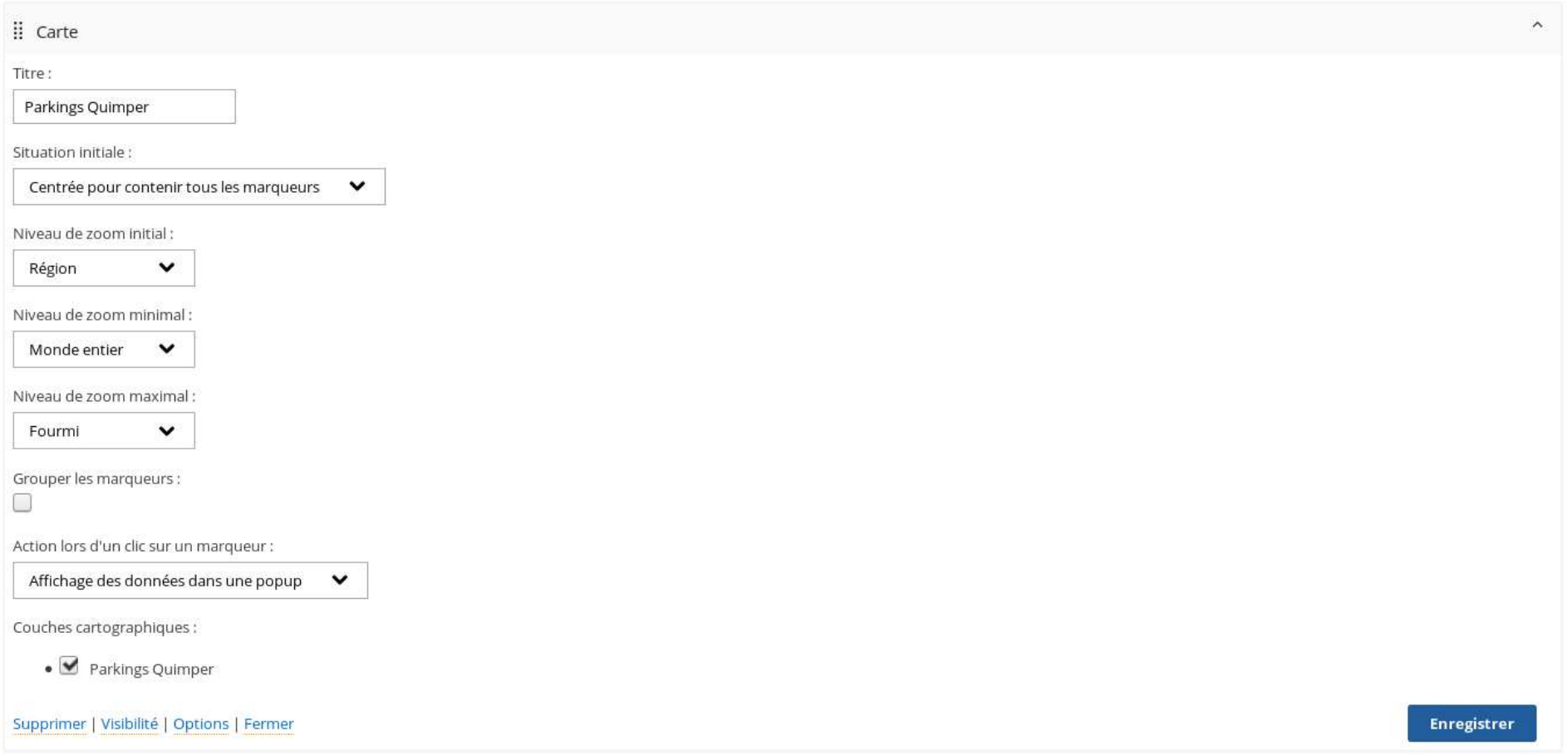
dresse, Description, Soustitre

Liste des propriétés à inclure, séparées par des virgules

Enregistrer Annuler

Supprimer

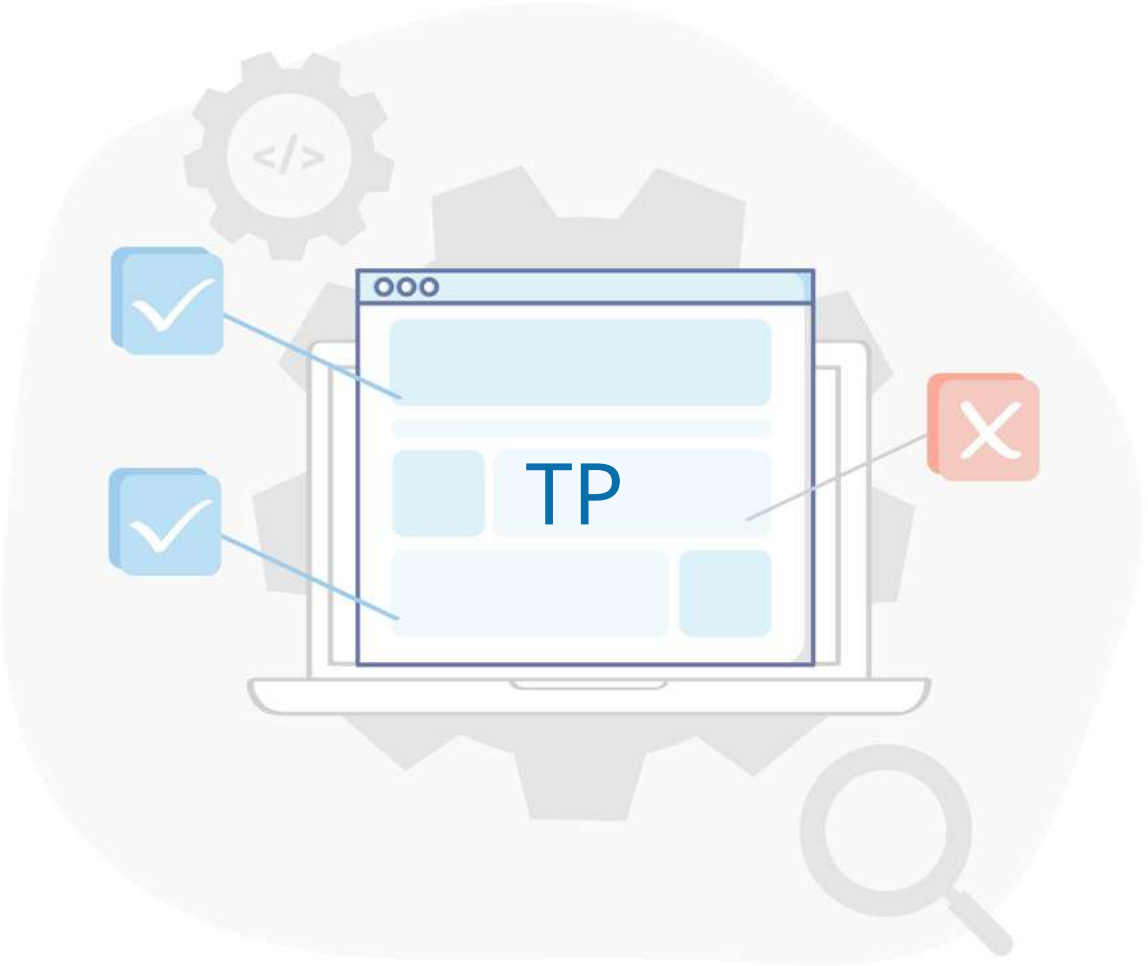
CARTOGRAPHIE



LES RESSOURCES

- Gestion depuis le kebab en page d’accueil.
- Ajout de liens ou d’images vers des ressources.
- L’ajout de ressource peut aussi se faire dans une cellule texte avec l’outil d’ajout d’une image.
- Sur le Saas Publik Entr’ouvert, la taille maximale d’une ressource est 200 Mo.

entr'ouvert



TP1

- Construire une page d’accueil
  - Mettre en avant les 5 démarches les plus utilisées d’une catégorie
  - Permettre de retrouver une démarche avec un code de suivi
  - Mettre un lien vers le site institutionnel

entr'ouvert

TP2

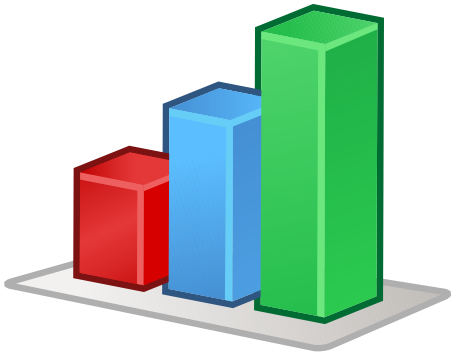
- Construire une page « Mon compte »
  - Informations utiles à l'usager

entr'ouvert

TP3

- Construire une page « Associations »
  - Y mettre toutes les informations utiles aux associations
  - Limiter l'accès aux associations
- Créer un menu
  - Avec la bonne visibilité des entrées du menu

entr'ouvert

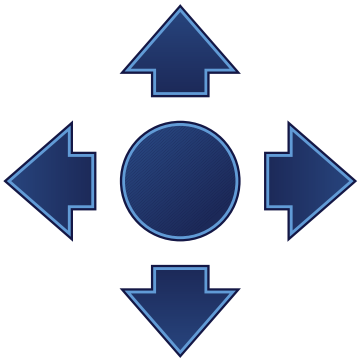


STATISTIQUES

STATISTIQUES

- Statistiques par défaut sur tous les formulaires
  - pour toutes les questions fermées (ssi case « Statistiques » / « Statistiques » cochée)
  - pour des critère pré-définis (par années, jours de la semaines, heures)
  - sur une période définie librement
- Statistiques disponibles dans la cellule graphe de combo avec type de graphe, taille et filtres pour :
  - agendas
  - connexion
  - SMS
  - démarches
- Statistiques sur mesures avec le module statistique (en voie de dépréciation)

entr'ouvert



PASSERELLE

PASSERELLE

- C’est l’endroit où l’on configure les connecteurs :
  - avec des logiciels métier
  - avec des outils transversaux (envoi de SMS,…)
  - avec les API nationales (API entreprise, BAN…)

APPROFONDISSEMENT

entr'ouvert

GABARITS ET FILTRES DJANGO

SYNTAXE

- Publik supporte la syntaxe des gabarits et filtres Django

  https://docs.djangoproject.com/fr/3.2/ref/templates/builtins/

- Mais aussi des filtres supplémentaires et spécifiquement développés pour Publik

  https://doc-publik.entrouvert.com/admin-fonctionnel/utiliser-les-variables/mecanique-de-template/

- Connaître quelques éléments de cette syntaxe est utile pour :
  - Écrire les conditions
  - Mettre en forme du texte (courriels, documents, messages dans l'historique…)
- Il faut utiliser la page inspect pour tester ses conditions ou ses gabarits

entr'ouvert

SYNTAXE

Exemple d'utilisation des gabarits :

Bienvenue {{session_user_display_name}},

Toute l'équipe de {{site_name}} vous remercie de votre inscription et vous souhaite une agréable visite.

Exemple d'une condition :

{% if form_var_regime_alimentaire %}

- Régime alimentaire : {{form_var_regime_alimentaire}}

{% else %}

- Aucun régime alimentaire précisé

{% endif %}

CONDITIONS ÉLABORÉES

- Conditions Imbriquées

```
{% if form_var_commune %}
    {% if form_var_commune == «Grenoble» %}texte pour les grenoblois
        {% else %} Texte pour les non-grenoblois
    {% endif%}
    {% else %}Texte si on ne connaît pas la commune
{% endif %}
```

- Les opérateurs :
  - L'égalité : ==
  - La différence : !=
  - Comparaison : > , < , >= , <=
  - Booléens : and , or , not

entr'ouvert

OPÉRATEURS BOOLÉENS

- {% if form_var_commune **and** form_var_quartier %} est vraie si les 2 variables existent
- {% if **not** form_var_quartier %} est vraie si la variable form_var_quartier n'existe pas
- {% if form_var_commune **or** form_var_quartier %} est vraie si la variable form_var_commune ou si la variable form_var_quartier existent
- {% if **not** form_var_commune **or** form_var_quartier %} est vraie si la variable form_var_commune n'existe pas ou si la variable form_var_quartier n'existe pas
- {% if form_var_commune **and not** form_var_quartier %} est vraie si la variable form_var_commune existe mais pas form_var_quartier

LES FILTRES

- Les filtres sont introduits avec des « pipes » (|) et s'appliquent au texte qui est à gauche du pipe
  - ils peuvent être enchaînés
  - certains prennent des paramètres
- Exemples :
  - {{ form_var_commune|upper }} : si form_var_commune contient « GrEnoBle », le résultat sera « GRENOBLE »
  - {{ form_var_commune|capfirst }} : si form_var_commune contient « grenoble », le résultat sera « Grenoble »
  - {{ form_var_fichier|slugify }} : si form_var_fichier contient « Notification 2018-05-25 », le résultat sera « notification-20180525 »
  - {{ value|add:"2" }} : si value = 4, alors cela va renvoyer 6
  - {{ form_var_commune|default:"" }} : si form_var_commune n'existe pas, cela ne va rien renvoyer (sans le filtre cela renverrait « None »)

entr'ouvert

FIRSTOF

- Affiche le premier paramètre qui ne vaut pas False. N'affiche rien si toutes les valeurs transmises valent False.
  - exemple d'utilisation : {% firstof var1 var2 var3 %}
  - c'est l'équivalent de :

{% if var1 %} {{ var1 }} {% endif %}

{% if var2 %}{{ var2 }}{% endif %}

{% if var3 %}{{ var3 }}{% endif %}

- Vous pouvez aussi utiliser une chaîne littérale comme valeur de repli dans le cas où toutes les variables spécifiées valent False :
  {% firstof var1 var2 var3 "Inconnu" %}