

Extranet France Signalisation | SPECIFICATIONS techniques**Spécifications techniques****EMETTEUR**

Manuel Grosrenaud

DESTINATAIRESAkram Amouri
Fadhel Abid
Mouna Hammami
Slim Masmoudi**HISTORIQUE**

| Date | Version | Motifs |
|------------|---------|--------------------------------------|
| 06/11/2013 | 1.0 | Configuration générale et homepage |
| 07/11/2013 | 1.1 | Lots « Demande de congé » et « GED » |
| 30/11/2013 | 2.0 | Lot « Chantier » |
| | | |
| | | |

TABLE DES MATIERES

| | |
|---|-----------|
| 1. Configuration générale..... | 4 |
| 1.1. Configuration serveur | 4 |
| 1.1.1. Php/Apache/MySQL | 4 |
| 1.1.2. Cron..... | 4 |
| 1.2. Site multi-langues | 4 |
| 1.3. Site multi-site | 4 |
| 1.4. Types de contenu | 4 |
| 1.5. Menus | 4 |
| 1.6. Thème et blocs | 5 |
| 1.6.1. Thème | 5 |
| 1.6.2. Blocs | 6 |
| 1.7. Utilisateurs, rôles et permissions | 6 |
| 2. Lot Homepage..... | 7 |
| 2.1. Types de contenu..... | 7 |
| 2.2. Vues | 7 |
| 2.3. Modules à utiliser | 8 |
| 2.4. Modules custom / features | 8 |
| 2.5. Custom javascript | 8 |
| 2.6. Divers | 8 |
| 3. Lot Demande de congé..... | 8 |
| 3.1. Formulaires | 8 |
| 3.2. Vues | 9 |
| 3.3. Modules à utiliser | 9 |
| 3.4. Modules custom / features | 9 |
| 3.5. Custom javascript | 9 |
| 3.6. Divers | 9 |
| 4. Lot Gestion Electronique de Documents (GED) | 10 |
| 4.1. Types de contenu, styles d'images et taxonomies | 10 |
| 4.2. Vues | 10 |
| 4.3. Détail d'une page..... | 12 |
| 4.4. Modules à utiliser | 12 |
| 4.5. Modules custom / features | 12 |
| 4.6. Custom javascript | 12 |
| 4.7. Divers | 12 |

| | |
|---|-----------|
| 5. Lot Chantiers..... | 13 |
| 5.1. Types de contenu..... | 13 |
| 5.2. Vues | 13 |
| 5.3. Détail d'une page..... | 14 |
| 5.3.1. Types de contenu | 14 |
| 5.3.2. Taxonomies | 15 |
| 5.3.3. Affichages | 15 |
| 5.3.4. Eléments et vues..... | 15 |
| 5.3.5. Principe de fonctionnement | 16 |
| 5.3.6. Maestro | 18 |
| 5.3.7. Custom | 18 |
| 5.3.8. Automatisation..... | 19 |
| 5.4. Modules à utiliser | 19 |
| 5.5. Modules custom / features | 19 |
| 5.6. Custom javascript | 19 |
| 5.7. Divers | 19 |
| 6. Lot standard / Autres pages | 19 |
| 6.1. Contact | 19 |
| 6.2. Mentions légales..... | 19 |

1 CONFIGURATION GENERALE

1.1 Configuration serveur

1.1.1 Php/Apache/MySQL

L'ensemble de ces configurations répond aux standards actuellement conseillés pour les projets Drupal 7 (consultables sur la page <https://drupal.org/requirements>).

La version de php est la 5.3.2-1

1.1.2 Cron

En raison des nombreux workflows et des nombreuses tâches sur ce dossier, on abandonnera la méthode par défaut du Poormanscron au profit d'une entrée dans la crontab du serveur qui déclenchera son exécution toutes les 5 minutes.

1.2 Site multi-langues

Site non multi-langue.

1.3 Site multi-site

Site non multi-site.

1.4 Types de contenu

Les types de contenus seront évoqués dans leurs parties respectives et détaillés dans les annexes ci-après, et on ajoutera donc quatre content-types à savoir « Chantier », « Document », « Tâche Modèle » et « Tâche » à ceux déjà existants.

1.5 Menus

Tous les menus listés ci-après doivent être administrables par les utilisateurs ayant les autorisations adéquates.

Main menu (main_menu)

4 liens principaux + 3 liens d'actions

- Accueil : pointe vers la homepage « <front> »
- Chantiers : pointe vers la page dédiée « /chantiers »
- Documents : pointe vers la page dédiée « /documents »
- Demande de congé : pointe vers la page dédiée « /demande-de-congé »
- Ajouter un chantier : déclenche le process d'ajout d'un contenu de type « Chantier »
- Ajouter un document : déclenche le process d'ajout d'un contenu « Document »
- Ajouter un utilisateur : déclenche le process d'ajout d'un utilisateur

Seuls les utilisateurs ayant les droits nécessaires seront autorisés à cliquer sur ces deux derniers liens (cf. partie 1.7)

Footer menu (footer_menu)

Quatre colonnes : les trois premières disposent de sous-éléments pointant vers différentes pages du site internet (donc externes à cet extranet) :

- « Entreprise »
 - Présentation
 - Partenaires
 - Equipe
 - Vidéos
 - Recrutement
- « Prestations »
 - Sécurisation du personnel
 - Sécurisation des circulations
 - Expertise ferroviaire
 - Coordination de sécurité
 - Travaux urbains
 - Travaux caténaires
 - Travaux de signalisation
- « Engagement Sécurité »
 - Méthodologie
 - Formations internes
 - Management de sécurité
 - Certification ISO 9001
- La dernière colonne regroupe différents items non rubriqués comme
 - « Chantiers » : pointe vers la page dédiée « /chantiers »
 - « Actualités » : pointe vers la page dédiée « /actualités »
 - « Contact » : pointe vers le formulaire de contact « /contact »
 -

1.6 Thème et blocs

1.6.1 Choix du thème

Comme le design doit être harmonisé avec le site internet, on optera donc pour le même choix de thème afin de faciliter les intégrations graphiques, à savoir « framework » : <https://drupal.org/project/framework>

Ce thème servira à créer notre starterkit que l'on nommera « fsextranet ».

| | | |
|--|--|-------------|
|  | Extranet France Signalisation SPECIFICATIONS TECHNIQUES | Version 2.0 |
| | | 30/11/2013 |

1.6.2 Blocs

Trois blocs sont repris sur toutes les pages de l'extranet et leur contenu doit rester administrable par les utilisateurs ayant les autorisations adéquates :

- le header : il contient le logo cliquable « France Signalisation » qui permet de revenir à l'accueil suivi du slogan « Notre priorité Votre sécurité » et du numéro de téléphone du standard.
- Le bloc de contact : il est situé dans le footer juste à droite du Footer menu et contient les coordonnées de France Signalisation ainsi qu'un lien cliquable vers le formulaire de contact (url « contact »)
- Le sub footer : il est situé en dessous du footer et contient une phrase de copyright, un lien vers la page des mentions légales (url « mentions-légales »), un lien vers la page de connexion ainsi que la mention « Réalisation : ACTENCY » qui pointe vers notre portail www.actency.fr

1.7 Utilisateurs, rôles et permissions

Même si le projet nécessite de créer 7 rôles d'utilisateurs, à savoir

- PDG
- CDT
- AD
- RQSE
- CG
- RC
- Agent

Leur fonctionnement reste assez simple pour que la gestion native de Drupal des rôles d'utilisateurs (url « /admin/people/permissions/roles ») réponde au besoin.

Les différents droits nécessaires seront alimentés par les modules contrib et custom et directement gérés depuis l'admin (url « /admin/people/permissions »).

2 LOT HOMEPAGE

La Homepage est « bloquée » sur le formulaire de connexion tant que l'utilisateur ne s'est pas authentifié, toutes les pages de l'extranet sont inaccessibles et redirigents sur cette page d'accueil.

Mais lorsque l'utilisateur est connecté, il trouve alors sur sa page d'accueil un tableau de bord qui le renseigne sur ses chantiers en cours, les derniers documents ajoutés dans l'extranet ainsi que des « flash info » (correspondants à des actualités).

2.1 Types de contenu

Trois types de contenu sont utilisés sur la page d'accueil :

- « Chantier » : détaillé dans sa partie dédiée au chapitre 5 et annexe 1.
- « Document » : détaillé dans sa partie dédiée au chapitre 4 et annexe 2.
- « Flash info » : correspondant au type de contenu « Article » natif à Drupal à renommer en « Flash info ». Les URLs de ce type de contenu suivront le motif : « flash-info/[node:title] » à définir dans pathauto.

2.2 Vues

Trois vues seront donc associées à ces types de contenu :

- « Chantiers en cours » : Filtre et liste dans un tableau les contenus de type « Chantier » auxquels l'utilisateur courant à accès. L'ordre de tri par défaut s'effectuera sur le statut du chantier (workflow le plus récent en premier). Ce tableau affichera trois colonnes :
 - Nom Chantier
 - Type
 - Statut
- « Nouveaux documents » : Filtre et liste dans un tableau les contenus de type « Document » auxquels l'utilisateur courant à accès. L'ordre de tri par défaut s'effectuera sur la date d'ajout du document (derniers ajouts en premier). Ce tableau affichera quatre colonnes :
 - Chantier
 - Type
 - Date
 - Catégorie

Ces deux premières vues seront l'une en dessus de l'autre dans la zone de contenu. On limitera l'affichage à X résultats avec une pagination en dessous de chaque vue.

- La troisième quant à elle sera dans la zone de sidebar right et affichera les teasers des contenus de type « Flash info », en filtrant bien de manière à assurer que ces contenus sont reliés à des chantiers auxquels l'utilisateur courant à accès. On n'utilisera pas de pagination et le nombre de résultats sera limité à Y.

2.3 Modules à utiliser

- Views (<https://drupal.org/project/views>)
- Pathauto (<https://drupal.org/project/pathauto>)

2.4 Modules custom / features

Il n'est pas nécessaire de créer un module custom pour cette homepage, par contre on créera une nouvelle feature « EFS | Homepage » (efs_homepage) qui contiendra les menus et les vues de ce chapitre, ainsi que la structure du type de contenu « Flash info » et sa configuration pathauto.

2.5 Custom javascript

Pas de javascript custom dans cette partie.

2.6 Divers

RAS.

3 LOT DEMANDE DE CONGE

La page « Demande de congé » (url : demande-de-congé) reprendra la même disposition que la homepage, à savoir : contenu plus une colonne à droite. Dans la zone de contenu on affichera un texte explicatif suivi (field « body » de la page : wysiwyg au format « filtered html ») de deux vues (détaillées ci-après en 3.2), et on placera dans la sidebar le formulaire de demande de congé.

3.1 Formulaires

Il y aura donc le formulaire de demande de congé dans la colonne de droite comprenant les champs suivants :

- Type de congé : **obligatoire**, référence à un terme de la taxonomie « Type de congé » (les termes précis seront communiqués ultérieurement), un seul choix possible.
- Date de début : **obligatoire**, champ date avec datepicker prérempli à la date du jour, ne peut pas être ultérieur à la date de fin.
- Matin/après-midi : pas d'étiquette sur ce champ, liste (select) de trois valeurs (matin/après-midi, matin, après-midi), un seul choix possible, présélection sur la première valeur « matin/après-midi ».
- Date de fin : **obligatoire**, champ date avec datepicker, ne peut pas être antérieur à la date de début.
- Matin/après-midi : idem que le troisième tiret.
- Commentaire : textarea de cinq lignes, obligatoire si type de congé « autres ».
- Bouton de soumission.

Ce formulaire comprendra également un champ « Statut » correspondant au workflow de validation : « En cours de validation » comme statut par défaut puis bascule en « Validé » ou

« Refusé » lorsqu'un utilisateur ayant les droits requis (AD ou Administrator) valide ou refuse la demande. Ce champ est inaccessible pour tous les autres rôles que AD ou Administrator.

3.2 Vues

Deux vues très similaires (on utilisera deux occurrences du même master en changeant simplement les filtres) seront affichées dans la zone de contenu :

- « Demandes en cours » : Filtre et liste dans un tableau les demandes de congé soumises par l'utilisateur courant qui sont toujours en cours de validation. L'ordre de tri par défaut s'effectuera sur la date de début (la plus récente en premier). Ce tableau affichera quatre colonnes :
 - Type de congé (nom du terme de taxonomie relié)
 - Date de début
 - Date de fin
 - Statut
- « Demandes passées » : C'est la même vue mais elle filtre les demandes qui ne sont plus en cours de validation.

Ces deux vues seront l'une en dessus de l'autre dans la zone de contenu. On limitera l'affichage à X résultats avec une pagination en dessous de chaque vue.

3.3 Modules à utiliser

- Views (<https://drupal.org/project/views>)
- Webform (<https://drupal.org/project/webform>)
- Workflow (<https://drupal.org/project/workflow>)
- Webform Mysql Views (https://drupal.org/project/webform_mysql_views)
- Data (<https://drupal.org/project/data>)

3.4 Modules custom / features

Il n'est pas nécessaire de créer un module custom pour cette page, par contre on créera une nouvelle feature « EFS | Demande de congé » (efs_conge) qui contiendra la structure du webform, ses vues ainsi que la taxonomie (structure + contenu) « Type de congé ».

3.5 Custom javascript

Pas de javascript custom dans cette partie.

3.6 Divers

Il faut créer la taxonomie « Type de congé », les champs par défaut suffisent.

A chaque nouvelle demande de congé, un email sera envoyé (adresse et contenu à venir – on utilisera le comportement natif à webform) pour avertir un utilisateur AD ou Administrator, qui se chargera de valider ou non la demande en mettant simplement le workflow à jour.

4 LOT GESTION ELECTRONIQUE DE DOCUMENTS (GED)

La page « Documents » (url : documents) n'utilisera pas la colonne de droite. Dans la zone de contenu on affichera un moteur de recherche de documents avec la vue des résultats juste en dessous. Pour chaque ligne de résultat, on pourra alors cliquer sur un lien « détail du doc » pour accéder à sa page dédiée (chapitre 4.3) ou un bouton « télécharger » pour récupérer le document localement.

4.1 Types de contenu, styles d'images et taxonomies

Il faudra donc paramétrer un nouveau type de contenu « Document » (document) avec les champs suivants en plus des champs par défaut (cf. ANNEXE 2) :

- « Chantier » (field_document_chantier) : référence à un noeud du type « Chantier »
- « Date » (field_document_date) : date équivalent à la date de valeur du document
- « Catégorie » (field_document_categorie) : référence à un terme de la taxonomie « Catégorie de document »
- « Sous catégorie » (field_document_souscategorie) : référence à un terme de la taxonomie « Catégorie de document »
- « Type » (field_document_type) : référence à un terme de la taxonomie « Type de document »
- « Document » (field_document_document) : fichier, limité aux extensions « .doc » et « .pdf » pour le moment
- « Version » (field_document_version) : textfield évoquant la version du document
- « Aperçu » (field_document_preview) : image dont l'affichage sera lié au style d'image « document_preview »

Ce type de contenu nécessite la création de deux nouvelles taxonomies :

- « Catégorie de document » (document_categorie
- « Type de document » (document_type)

Et on ajoutera également le type d'image « Aperçu de document » (document_preview) qui est un scal & crop de largeur 45px et de hauteur 50px.

Le titre (champ par défaut title) sera renseigné programmatically lors du node_presave() (cf. chapitre 4.5). Le champ body sera renommé en « Description ».

4.2 Vues

Nous utiliserons une vue « Documents » avec filtres exposés pour créer cette page (chemin : /documents). Le bloc de recherche exposant les filtres sera placé au dessus et les résultats correspondants seront affichés en dessous par paginations de dix éléments. Nous utiliserons l'ajax (option proposée par views) pour que les résultats soient rafraichis à chaque nouvelle saisie/modification de critère de recherche.

- moteur de recherche :

Le moteur de recherche sera découpé en trois niveaux :

- un premier niveau permettra une saisie (autocomplétion) dans un champ intitulé « Chantier » : le bouton de soumission « OK » lancera la recherche sur les documents dont les noms de chantiers correspondent.
 - Le deuxième niveau va contenir la recherche avancée sur les champs date, catégorie et type :
 - « Date » : datepicker
 - « Catégories » et « Sous-catégories » : Ces deux champs de recherche seront organisés en facettes : l'utilisateur choisit déjà une (ou plusieurs) catégorie(s) (textfield avec autocomplétion sur les termes de premier niveau de la taxonomie « Catégorie de document » séparés par une virgule), puis un select de sous-catégories lui est alors proposé (les options sont obtenues à partir des éléments de second niveau pour le(s) terme(s) sélectionnés juste au dessus, dans cette même taxonomie « Catégorie de document »)
 - « Type » : checkboxes obtenues de la taxonomie « Type de document »
 - Enfin on affichera le nombre de résultats dans un troisième niveau
- résultats :

Les résultats seront affichés sous forme de liste avec une pagination disponible au dessus et en dessous. Chaque élément reprendra :

- L'aperçu du document
- Le titre reformaté
- La date de création
- La date de modification
- L'extension (label « Format »)
- La taille
- Le résumé ou corps coupé à 600 caractères
- Un bouton permettant le téléchargement du document
- Un lien vers la page dédiée au document

4.3 Détail d'une page

La page dédiée à un document aura pour motif d'url « /document/[node :id] » et affichera exactement les mêmes informations que sur la liste des résultats de la page /documents en remplaçant simplement :

- le résumé par le corps (body) du nœud
- le lien « Détail du doc » par un lien « Retour à la liste »

Le positionnement des éléments sur la zone de contenu sera quant à lui un peu différent.

4.4 Modules à utiliser

- Views (<https://drupal.org/project/views>)
- Pathauto (<https://drupal.org/project/pathauto>)
- Entity Reference (<https://drupal.org/project/entityreference>)
- FacetAPI (<https://drupal.org/project/facetapi>)

4.5 Modules custom / features

Comme pour les chapitres précédents, on utilisera une nouvelle feature « EFS | Documents » (efs_documents) qui contiendra la structure du type de contenu « Document » ainsi que ses taxonomies, vues et styles d'images associés (cf. chapitre 4.1).

On développera également un module custom « EFS Document title format » (efs_doctitle_format) qui implémentera un hook sur le node_presave() des documents pour renseigner leur champ « Titre » (title) en utilisant le motif suivant :

[chantier:title]-[document:type]-[document:date]-[document:categorie:name]-
[document:sous-categorie:name]-[document:sous-categorie:increment]- [document:version]

Si les documents sont des modèles (identifiés par leur type) alors on utilise le même motif en remplaçant le titre du chantier par MODELE en toutes lettres.

4.6 Custom javascript

Pour lien « Retour à la liste » d'une page document, on utilisera la méthode javascript history.back() pour retrouver la même liste de résultats et les mêmes critères de filtres que ceux précédemment saisis. Ce bouton sera à placer dans un template node—document.tpl.php.

4.7 Divers

On placera les documents uploadés dans sites/default/files/GED/documents/ et leurs aperçus dans sites/default/files/GED/previews/

Le motif d'URL pour les documents à configurer dans pathauto sera le suivant :
document/[node :nid]

5 LOT CHANTIER

La page « Chantiers » (url : chantiers) n'utilisera pas la colonne de droite. Dans la zone de contenu on affichera les filtres de recherche de chantiers avec la vue des résultats juste en dessous.

5.1 Types de contenu

Il faudra donc paramétrer un nouveau type de contenu « Chantier » (chantier) avec les champs suivants en plus des champs par défaut (cf. ANNEXE 1) :

- « Nom chantier » (renommer le champ par défaut title)
- « Type de chantier » (field_chantier_type) : référence à un terme de la taxonomie « Type de chantier »
- « Statut » (field_chantier_status) : workflow à trois étapes : « Préparation de chantier » (par défaut), « Suivi de chantier » et « Fin de chantier »
- « Date de début » (field_chantier_startdate) : date équivalent à la date de démarrage du chantier (attention, n'est pas la date de création du contenu)
- « Date de fin » (field_chantier_enddate) : date équivalent à la date de fin du chantier
- « Remarque » (renommer le champ par défaut body)
- « CDT » (field_chantier_ur_cdt) : référence à un utilisateur ayant le rôle CDT
- « AD » (field_chantier_ur_ad) : référence à un utilisateur ayant le rôle AD
- « RQSE » (field_chantier_ur_rqse) : référence à un utilisateur ayant le rôle RQSE
- « CG » (field_chantier_ur_cg) : référence à un utilisateur ayant le rôle CG
- « Agents » (field_chantier_ur_agents) : référence à un utilisateur ayant le rôle Agent, plusieurs valeurs possibles

Ce type de contenu nécessite la création d'une nouvelle taxonomie :

- « Type de chantier » (chantier_type)

5.2 Vues

Nous utiliserons une vue « Chantiers » avec filtres exposés pour créer cette page (chemin : /chantiers). Le bloc de recherche exposant les filtres sera placé au dessus et les résultats correspondants seront affichés en dessous par paginations de dix éléments.

- Recherche : Le bloc de recherche va contenir les filtres suivants (sans labels), on soumet avec un bouton « Rechercher » :
 - « Nom chantier » : autocomplete field sur le champ « Nom Chantier » avec comme placeholder « Tapez le nom d'une ville »
 - « Sélectionnez un type » : select sur le champ « Type de chantier »
 - « Choisissez un statut » : select sur le statut du chantier

- Résultats : Les résultats seront affichés sous forme de tableau avec une pagination disponible en dessous. Chaque colonne aura pour header :
 - Nom Chantier
 - Type
 - Statut
 - Un lien vers le contenu « Voir »

5.3 Détail d'une page

Vu la complexité des tâches et workflows de ce lot, la solution proposée ci-après vise à constituer un modèle de processus normalisé qui permettra de gérer chaque tâche de la même façon, quel que soit son niveau de complexité ou sa position dans le cycle de vie d'un chantier.

Le détail d'une page chantier va donc se découper sur un template trois colonnes (cf. partie 5.3.3), positionnées sous un bandeau comprenant le nom du chantier et son workflow courant (à la manière du tunnel de commande de Drupal Commerce par exemple).

Alors que les deux colonnes de droites seront les mêmes pour chaque affichage, celle de gauche sera directement reliée à l'état actuel du chantier.

5.3.1 Types de contenu

Nous aurons besoin de deux nouveaux types de contenu pour cette partie, à savoir « Tâche modèle » (tache_modele) et « Tâche » (tache).

Une tâche modèle suivra le pattern d'URL tâche-modèle/[node:title] et sera structurée comme suit (cf. ANNEXE 3) :

- « Intitulé de la tâche » (title)
- « Position » (field_tachem_position) : référence à un terme de la taxonomie « Tâches workflow » détaillée dans la partie 5.3.2
- « Périodicité » (field_tachem_periode) : référence à un terme de la taxonomie « Période » détaillée dans la partie 5.3.2
- « Initiateurs » (field_tachem_roles) : référence à un rôle

Une tâche (cf. ANNEXE 4) suivra le pattern d'URL tâche/[node:title] et sera structurée comme suit :

- « Intitulé de la tâche » (title)
- « Remarques » (body)
- « Date d'effet » (field_tache_date) : popup date prérempli à la date du jour
- « Documents » (field_tache_documents) : référence à un ou plusieurs Documents (maximum 5), widget d'autocomplétion
- « Chantier » (field_tache_chantier) : référence à un chantier, widget d'autocomplétion
- « Tâche modèle » (field_tache_modele) : référence à la tâche modèle ayant servi à la création de la tâche (cf. 5.3.5 pour le principe de fonctionnement)

5.3.2 Taxonomies

Une première taxonomie « Tâches workflow » (taches_workflow) sera créée afin d'ordonner les différentes tâches à réaliser dans le cycle de vie d'un chantier. Si les tâches n'ont pas d'ordre spécifique, elles n'apparaissent pas dans cette taxonomie.

Et une seconde taxonomie « Période » (periode) sera mise en place afin de gérer les éventuelles périodicités des tâches. Les termes possibles seront soit une fréquence du style « hebdomadaire » ou encore « bimensuelle », soit « aléatoire » pour la gestion des tâches qui n'auront pas de périodicité fixe, soit « unique » pour la gestion des tâches qui ne s'effectuent qu'une seule fois par chantier. Cette taxonomie utilisera un champ « Nombre de jours » qui renseignera de manière plus précise l'intervalle de cette période en nombre de jours. On indiquera 0 pour une tâche aléatoire et -1 pour une tâche qui ne s'effectue qu'une seule fois.

5.3.3 Affichages

Comme pour une commande sur un site d'e-commerce, on affichera la page du chantier en fonction de l'état actuel de son workflow « Statut ». Dans les faits seule la première colonne sera adaptée, les deux autres ainsi que le reste des éléments de la page resteront toujours identiques.

Le contenu de ces pages étant généré par du code custom, il n'est pas nécessaire de pousser des modules comme display suite ou context, mais le développeur sera libre de les utiliser s'il estime que cela lui facilite le travail.

5.3.4 Eléments et Vues

Ce template en trois colonnes affichera respectivement les éléments suivants :

- **Colonne de gauche**, statut « Préparation de chantier » : empilement de vues listant les tâches à venir, les tâche à faire et les tâches effectuées pour le statut actuel du chantier. Si une vue est vide on ne l'affiche pas.
- **Colonne de gauche**, statut « Suivi de chantier » : idem que pour le statut précédent, sauf qu'on enlève les tâches à venir et qu'on génère autant d'empilements qu'il n'y a de périodicités. On aura donc tâches hebdomadaires à faire / tâches hebdomadaires effectuées, tâche mensuelles à faire / tâches mensuelles effectuées, etc.
Attention : l'affichage de la page pour ce statut prend un paramètre, qui est le numéro de semaine (numéro de la semaine courante par défaut). On peut choisir d'afficher cette page pour une semaine antérieure (le minimum étant la date de bascule du chantier au statut « Suivi de chantier ») à l'aide d'un select qui sera positionné à côté du titre de la page (ex. « Chantier Strasbourg – semaine 31). On verra alors les vues des tâches effectuées cette semaine là, pour chaque période.
- **Colonne de gauche**, statut « Fin de chantier » : idem que le premier statut.

En face de chaque ligne affichant une tâche « à faire », on placera un bouton « Effectuer la tâche » qui déclenchera la création/validation de la tâche (cf. parties 5.3.5 à 5.3.7). Ce bouton ne sera accessible qu'aux rôles autorisés (contrôle des permissions sur la tâche modèle).

Il serait fastidieux de gérer ces vues via views et on préférera créer une méthode générique (cf. partie 5.3.7) qui sera appelée autant de fois que nécessaire en adaptant simplement ses paramètres d'entrée (chantier, période, etc.), et qui restituera son contenu via un fichier .tpl.php pour faciliter le travail de l'intégrateur et l'évolution éventuelle des affichages.

- **Colonne du milieu « Documents liés »** : On liste dans cette colonne tous les documents reliés au chantier via son champ « Documents ». On ne liste pas ceux qui lui sont reliés via une tâche (cf. partie 5.3.1) pour éviter d'avoir trop d'instances à afficher. Chaque item renvoie à la page dédiée du document correspondant dans la GED.
- **Colonne de droite** : On empile trois boutons déclenchant leurs workflows respectifs (cf. partie 5.3.6) « Ajouter un agent », « Ajouter une facture » et « Ajouter un relevé ». Et juste en dessous, on affiche les affectations sur ce chantier (cf. les champs de la partie 5.1 commençant par « field_chantier_ur ») dans un encadré « Equipe ».
- **Bandeau supérieur** : il est situé au dessus de ces trois colonnes et affiche le titre de la page (Nom du chantier (title)) et juste en dessous la frise en trois étapes représentant le workflow d'un chantier (statut).
- **Bandeau inférieur** : il est situé en dessous de ces trois colonnes et permet aux rôles autorisés de faire passer le chantier au statut suivant en cliquant sur un bouton.

5.3.5 Principe de fonctionnement

Pour expliquer le principe de fonctionnement, nous partons du postulat que les tâches modèles sont créées et correctement reliées à leurs taxonomies associées. Nous rappelons également que ce principe a été conçu pour que toutes les tâches soient traitables de la même manière, quel que soit le statut du chantier ou la complexité de la tâche.

a/ Un utilisateur ayant les permissions requises crée un nouveau chantier

b/ Sur la page du chantier, nous en serons donc à l'étape « Préparation de chantier », le nom sera renseigné et on pourra déjà lister les éventuels documents liés et personnes affectées, le cas échéant.

c/ Pour afficher la colonne de gauche (les tâches) il suffit de faire les trois requêtes suivantes :

- « Tâches effectuées » : Pour ce chantier, y a-t-il des contenus existants de type tâches, dont les tâches modèles font référence à un terme enfant du terme « Préparation de chantier » de la taxonomie tâches workflow ?
- « Tâche à faire » : Pour ce chantier, quelle est la première tâche modèle, faisant référence à un terme enfant du terme « Préparation de chantier » de la taxonomie tâches workflow depuis laquelle aucun contenu de type tâche n'a encore été créé ? On n'oublie pas d'afficher à côté le bouton « Effectuer cette tâche », dont le comportement va être expliqué juste après.
- « Tâches à venir » : Pour ce chantier, quelles sont les tâches modèles qui font référence à un terme enfant du terme « Préparation de chantier » de la taxonomie tâches workflow situé en dessous de celui de la « tâche à faire » (requêtée juste avant) dans le classement de cette taxonomie tâches workflow ?

d/ Un utilisateur ayant les permissions requises clique sur un bouton « Effectuer cette tâche » :

- On crée automatiquement un nouveau type de contenu tâche en préremplissant le titre avec celui de la tâche modèle, le chantier avec le chantier courant et bien sur la tâche modèle avec la tâche modèle ayant servi à cette création (cf. partie 5.3.7).
- Si la tâche le nécessite, un workflow Maestro sera automatiquement déclenché (cf. partie 5.3.6).
- Si la tâche le nécessite, l'utilisateur pourra relier à cette tâche jusqu'à cinq documents de la GED. Si ces documents n'ont pas encore été créés dans la GED, l'utilisateur trouvera un lien « Ajouter un document » dans le formulaire d'édition de la tâche (en dessous du champ « Documents »), qui permettra d'enregistrer un nouveau document dans la GED puis de revenir à son formulaire initial d'édition de la tâche (cf. partie 5.3.7).

e/ Un utilisateur ayant les permissions requises bascule le statut d'un chantier à l'état suivant : on retourne alors au point b/ (en adaptant le statut).

Remarque complémentaire : Pour le point c/ lors du statut « Suivi de chantier », on effectue autant de fois cette série de trois requêtes qu'il n'y a de périodes dans la taxonomie « Période », avec le filtre de périodicité correspondant, ainsi que le paramètre sur le numéro de semaine expliqué dans la partie 5.3.4.

5.3.6 Maestro

On va utiliser Maestro pour gérer les tâches plus complexes nécessitant un workflow particulier (se référer au dossier commercial pour les workflows en question). On normalisera les process de la manière suivante :

- 1 seul workflow maestro par tâche
- 1 seule tâche par workflow maestro
- les workflows maestro suivent tous le schéma suivant :
 - start : trigger = création d'une tâche
 - IF : test sur la tâche modèle pour identifier les spécificités
 - Workflow spécifique
 - end

De plus, on utilisera trois workflows Maestro pour les trois actions suivantes (cf. pages 12 et 13 du dossier commercial) :

- Ajouter un agent
- Ajouter une facture
- Ajouter un relevé

5.3.7 Custom

Cette partie 5.3 va nécessiter le développement de deux nouveaux modules custom : « FSE Forms Alter » (fse_forms_alter) et « FSE Chantier » (fse_chantier), dans le package existant « FSE ».

Le premier va, comme son nom l'indique, hooker le form_alter du type de contenu « Tâche » afin de préremplir les champs title, chantier et tâche modèle par les valeurs par défaut, comme expliqué dans la partie 5.3.5.

Il va également ajouter le lien « Ajouter un document » en dessous du champ « Documents », qui permettra à l'utilisateur d'ajouter un document dans la GED directement depuis le process de validation de sa tâche s'il ne l'a pas fait avant, comme expliqué dans la partie 5.3.5, puis de revenir à l'édition de sa tâche en pouvant cette fois-ci lui lier le document qu'il vient juste d'enregistrer.

On utilisera ce même procédé de lien « Ajouter un document » dans le form_alter du chantier.

Le second va implémenter une méthode générique du style chantier_tasks_view() qui va être appelée avec les paramètres désirés pour chaque affichage d'un ensemble « tâches à venir » « tâches à faire » et « tâches effectuées », comme décrit dans la partie 5.3.4.

Cette méthode poussera son résultat dans un template chantier_tasks_view.tpl.php qui pourra facilement être exploité par l'intégrateur et/ou adapté pour d'éventuelles évolutions ultérieures.

5.3.8 Automatisation

La solution retenue d'utiliser des tâches modèles reliées à des taxonomies a l'avantage d'implémenter un affichage dynamique des pages qui permet de se passer du concept d'automatisation, qui consistait à vérifier/créer chaque semaine les tâches à réaliser.

5.4 Modules à utiliser

- Views (<https://drupal.org/project/views>)
- Pathauto (<https://drupal.org/project/pathauto>)
- Workflow (<https://drupal.org/project/workflow>)
- Maestro (<https://drupal.org/project/maestro>)

5.5 Modules custom / features

Comme pour les chapitres précédents, on utilisera une nouvelle feature « EFS | Chantiers » (efs_chantiers) qui contiendra la structure du type de contenu « Chantier » ainsi que ses taxonomies et vues.

5.6 Custom javascript

Pas de javascript custom dans cette partie.

5.7 Divers

Le motif d'URL pour les documents à configurer dans pathauto sera le suivant :
chantier/[node :title]

6 LOT STANDARD / AUTRES PAGES

6.1 Contact

On utilisera le module « Contact » du Drupal Core.

6.2 Mentions légales

On ajoutera une page dédiée de type « basic page » pour les mentions légales.