



epicraft

Feuilles de référence

Informations concernant ce document

Editeur :	Epiconcept, 47 rue de Charenton, 75012 Paris, France - www.epiconcept.fr
Titre :	Epicraft Feuilles de Référence (pour la version 1.22)
Version actuelle :	Version 3.3 – 12 Août 2021
Versions précédentes :	v1.01 – 21 juillet 2016 ; v1.1 – 4 novembre 2016 ; v1.2 – 24 novembre 2016 ; v2 – 16 juin 2017 ; v2.1 – 10 novembre 2017 ; v2.2 – 11 décembre 2017 ; v2.3 – 4 janvier 2018 ; v2.4 – 26 avril 2018 ; v2.5 – 12 septembre 2018 ; v2.6 – 18 janvier 2019 ; v2.61 – 21 mars 2019 ; v2.7 – 16 avril 2019 ; v2.71 – 25 avril 2019 ; v2.8 - 15 novembre 2019 ; v2.9 - 17 janvier 2020 ; Version 3.0 – 5 juin 2020 ; Version 3.1 – 24 septembre 2020 ; Version 3.2 – 26 Janvier 2021
Concerne :	Editeur Epicraft version 1.22
Licence :	Copyright © Epiconcept 2021
Auteur(s) :	Curtis Broderick
Distribution :	Publique
Autres langues :	Disponible en anglais
Documents corollaires :	Fiches de référence Voozanoo 4, Les tutoriaux vidéo Voozanoo (Chaine Epiconcept sur YouTube®)
Autres informations :	
Commentaires :	Merci de bien vouloir adresser vos remarques concernant ce document à c.broderick@epiconcept.fr

Table des matières

Informations concernant ce document	2	Élément : Onglets	9
Table des matières	3	Élément : Matrice	9
Glossaire	8	Élément : Bouton (simple)	9
Organiser des pages avec des dossiers	9	Composante : Bouton (avec passage de données)	9
Propriétés de la page	9	Pages d'accueil	9
Coloration des varsets	9	Affichage conditionnel : instructions JavaScript	9
L'espace de travail	9	Importer des variables (1 de 2)	9
Propriétés générales	9	Importer des variables (2 de 2)	9
Propriétés d'affichage	9	Importer des dictionnaires	9
Propriétés de contrôle	9	Export des pages et des dictionnaires	9
Propriétés Aide et Structure	9	Avancé : Création de liens entre les tables	9
Type de variable : Nombre	9	Avancé : Requêtes de données (1 de 3)	9
Type de variable : Date	9	Avancé : Requêtes de données (2 de 3)	9
Type de variable : Booléen	9	Avancé : Requêtes de données (3 de 3)	9
Type de variable : Choix	9	Avancé : Variables calculées (1 de 3)	9
Élément : Texte	9	Avancé : Variables calculées (2 de 3)	9
Élément : Groupe	9	Avancé : Variables calculées (3 de 3)	9
Élément : Colonnes	9	Avancé : listing des données sur le formulaire	9
		Publier un projet	9

ATTENTION : Nouveau depuis la version 1.20, les options de votre interface dépendent maintenant de votre niveau d'utilisateur. Il se peut que vous ne puissiez pas voir ou utiliser toutes les options qui sont indiquées dans cette documentation utilisateur. Veuillez consulter la section "Rôles des utilisateurs et statut du cycle de vie" pour comprendre pourquoi vos options d'édition pourraient être réduites.

A propos de ce document

Ce document décrit l'organisation et l'utilisation de l'éditeur Epicraft destiné à la création des pages d'une application fondée sur la base de données Voozanoo. La création d'une application Voozanoo s'appuie à la fois sur l'application de base (voir les fiches de référence Voozanoo 4) et sur cet éditeur. D'une manière générale, le noyau gère les données et les utilisateurs alors que l'éditeur permet de créer les différents écrans de l'application.

Ce document utilise le signe > pour indiquer « puis aller sur » pour enseigner au lecteur comment accéder à un écran ou une fonction donnée. Par exemple : Accueil> Rôles> Ajouter.

Le concept de Voozanoo 4 et les différences avec Voozanoo 3

Voozanoo 4 est une mise à jour majeure du framework de développement Voozanoo. Pour résumer, si Voozanoo 3 était un environnement un peu monolithique de création de base de données, Voozanoo 4 adopte, lui, une approche résolument modulaire laissant le choix d'inclure ou non certains modules. Comme avec toutes les évolutions, les nouveaux avantages ont aussi leurs inconvénients. Il faut se familiariser avec les modules, comprendre leur fonction, savoir lesquels inspecter lorsqu'on souhaite apporter des modifications à l'application.

Mais Voozanoo 4 propose un avantage majeur : il repose sur une couche très structurée du logiciel Zend™. C'est un framework MVC (model-view-controller). Autrement dit, il sépare le Modèle logique de la Visualisation et utilise des Contrôleurs. Voozanoo est, de ce fait, beaucoup plus fiable et flexible. Autre avantage, certains éléments autrefois exclusivement réservés au back-end, sont désormais accessibles en front-end pour certains types d'utilisateurs. L'exemple le plus courant est celui de l'attribution d'un rôle utilisateur spécifique (« catégories » d'utilisateurs dans Voozanoo 3) pour gérer et créer d'autres utilisateurs. Les fonctionnalités back-end peuvent effectivement être transportées vers le front-end.

Pour créer une application, vous devez au minimum utiliser le noyau Voozanoo et l'Editeur Epicraft.

L'éditeur Epicraft et ses fonctions clé

L'éditeur est utilisé pour créer et modifier ce que l'on voit sur les écrans tandis que l'application (voir les *fiches de référence Voozanoo 4*) permet de modifier le fonctionnement de l'application, de définir les droits des utilisateurs, et de saisir et voir ses données.

Editeur Epicraft	Application Voozanoo 4
Créer et modifier une application <ul style="list-style-type: none"> • pages • navigation • champs de saisie • jeux de variables (varsets) • fichiers ressources • éléments customisés • requêtes des données 	Créer et gérer les droits utilisateurs Créer, gérer et importer des utilisateurs Accorder ou révoquer les droits utilisateurs Créer et gérer des groupes Créer et gérer des listings Créer et gérer des exports de données Créer et gérer des filtres Gérer l'utilisation sur app mobile Gérer les accès entre modules - OAuth Gérer les politiques de données personnelles
Importer des pages Importer des dictionnaires	

L'éditeur Epicraft et les fichiers ressources

Une application Voozanoo 4 est pilotée par des fichiers ressources lisibles par l'homme, principalement écrits en XML. Ils représentent la couche entre l'aspect visible de « l'application » et l'exécution bas niveau de la base MySQL et du code PHP. L'éditeur Epicraft fournit une interface graphique pour créer les fichiers ressources des pages visibles par l'utilisateur final (fiches de saisie de données). L'éditeur Epicraft dispose d'un champ spécial pour personnaliser chaque élément de même qu'un champ général (Bulk) pour qu'un développeur expérimenté puisse ajouter du code XML spécifique aux fichiers ressources. L'ajout de code XML aux fichiers ressources est généralement confié aux seuls développeurs d'Epiconcept car cela pourrait facilement provoquer le crash de votre application.

Les pages d'une application sont construites en utilisant l'éditeur Epicraft et écrites ensuite dans le projet via une des commandes du menu Déploiement. Le processus est similaire à celui de Voozаноо 3 où l'on passe des écrans roses de l'éditeur aux écrans bleus de l'application réelle.

Editeur multi-utilisateur

Un projet est souvent le fruit de l'effort conjoint des membres d'une équipe. L'éditeur Epicraft permet d'attribuer à autant de personnes que nécessaires des privilèges de création sur un projet (application). Il est donc important de comprendre que les actions d'une personne peuvent changer ce que les autres voient. En haut à gauche de l'affichage de la page, des icônes vous informent quand d'autres travaillent sur le même projet que vous.  Ici, deux autres utilisateurs avec les initiales R et V y travaillent en même temps que vous.

Editeur hors-ligne

Bien qu'Epicraft soit une application web, vous ne devez pas nécessairement être connecté à Internet pour l'utiliser. En laissant l'onglet de votre navigateur ouvert lorsque vous vous déconnectez d'Internet (quittez une zone Wifi ou débranchez le câble Ethernet de votre ordinateur portable), vous pouvez continuer à modifier vos projets (l'arrière-plan de la barre de titre devient rose). Si vous fermez l'onglet (ou fermez votre navigateur), vous devrez alors vous reconnecter à Internet pour accéder au projet. Vos modifications ne seront pas perdues. Si plusieurs personnes modifient simultanément les pages du projet hors connexion, certaines de leurs actions pourraient entrer en conflit lors de la reconnexion (quand elles sont renvoyées au serveur Epicraft). Au moment de la reconnexion au serveur d'Epicraft, toutes les modifications apportées au projet depuis la dernière connexion seront instantanément appliquées à la version actuelle présente sur le serveur. Si l'une de vos actions ne peut être appliquée (ex : vous avez modifié une variable que quelqu'un d'autre a effacé pendant que vous étiez hors ligne), alors l'action sera ignorée.

Projets multilingues et jeux de caractères étrangers

Il est possible de créer une version multilingue de votre application. Pour ce faire, votre application est d'abord créée dans une langue de base, par exemple le français. Vous devrez ensuite traduire tous les textes français vers d'autres langues cible via l'application open source PoEdit (www.poedit.net). On peut

assigner une langue à un utilisateur ou le laisser libre de choisir sa langue d'interface. L'éditeur Epicraft à proprement parler est seulement disponible en anglais et en français.

Voozаноо emploie le système d'encodage Unicode UTF-8 pour ses textes. Par conséquent, les pages gèrent les jeux de caractères étrangers tels que le chinois, le russe, le grec, l'hébreu, le cyrillique et l'arabe.

Un design réactif pour toutes les tailles d'écran

Voozаноо met en œuvre la solution open source Bootstrap (www.getbootstrap.com) pour afficher des pages Internet de façon confortable sur des écrans de différentes tailles (écran de bureau, ordinateur portable, tablette ou smartphone). L'éditeur Epicraft est paramétré par défaut sur un système d'affichage à 12 colonnes ; les résultats peuvent être prévisualisés en choisissant différentes tailles d'écran afin de vérifier la bonne disposition des champs et de leurs labels. Gardez à l'esprit que Bootstrap s'appuie



seulement sur du CSS et qu'il est facile de le modifier par un CSS personnalisé.

Les rôles, niveaux de risque et état du cycle de vie des projets

Nouveau depuis la version 1.20, le système de droits d'utilisation des différentes fonctions d'Epicraft a été repensé afin de permettre au chef de projet d'Epicconcept de disposer d'une plus grande souplesse pour assurer la sécurité des projets. Chaque projet peut être placé dans l'un des états du cycle de vie afin de rendre la modification d'un projet plus facile ou plus difficile (pour éviter des erreurs coûteuses). Lorsqu'un projet passe d'un état à un autre, un utilisateur, en fonction de son rôle au sein de l'équipe de projet, sera autorisé à modifier seulement certaines parties et à exécuter certaines actions d'un projet.

- **Ouvert** : un nouveau projet démarre initialement dans cet état du cycle de vie
- **Protégé** : un projet passe alors à cet état
- **Verrouillé** : un projet en cours est généralement placé dans cet état

Dans le but de permettre la flexibilité et la protection, chacune des plus de 300 actions d'Epicraft s'est vu attribuer un niveau de risque spécifique.

Affichage uniquement : permet de voir les valeurs et les options, mais pas de les modifier

Bénigne : la modification de ces champs ne fait que changer les informations sur l'écran de l'utilisateur mais ne nuit en rien à l'intégrité du projet ou à la structure de la base de données.

Dangereux : la modification de ces champs et actions peut modifier la structure de la base de données de votre application et/ou permettre la perte de données.

Critique : l'exécution de ces commandes peut avoir des effets négatifs irrémediables très graves si elles ne sont pas utilisées correctement.

Voir la section ci-dessous, *Rôles de l'utilisateur et état du cycle de vie* pour plus d'explications et d'exemples sur la façon dont l'état du cycle de vie du projet et les niveaux de risque fonctionnent ensemble.

Projets multilingues et jeux de caractères étrangers

Il est possible de créer une version multilingue de votre application. Cela se fait en créant votre application dans la langue de base anglais (pas absolument nécessaire mais cela facilitera la mise à jour), puis en créant des traductions de tous les textes anglais dans d'autres langues en utilisant l'environnement de traduction WYSIWYG. Un utilisateur peut ensuite se voir attribuer une langue ou sélectionner la langue dans laquelle il préfère utiliser l'application. L'éditeur Epicraft lui-même n'est disponible qu'en anglais et en français. Consultez les fiches de référence Voozanoo 4 pour des instructions sur la façon de mettre en œuvre et d'utiliser Crowdin™.

Voozanoo utilise le système Unicode UTF-8 pour ses textes. Cela signifie que les jeux de caractères étrangers tels que chinois, russe, grec, hébreu, cyrillique et arabe peuvent être utilisés sur les pages.

Contrôle de l'intégrité technique du système

Nouveau dans la version 1.8 est un indicateur sur le côté droit de l'en-tête  (0) qui indique le nombre d'anomalies dans votre projet ont été détectées. Cliquez dessus pour voir une description de chaque anomalie trouvée. Par exemple, ce "Tech check" indiquera s'il y a des références à des tables ou à des variables qui ont été supprimées.

Rechercher

La nouveauté de la version 1.19 est la fonction recherche. Il permet de trouver tout texte, sensible à la casse ou non, dans les champs des pages, les dictionnaires et les variables archivées. Cliquez sur l'une des réponses pour être transporté directement dans le champ ou le dictionnaire. Le champ de recherche peut être ouvert avec le raccourci clavier alt-F. Les mots dans les composants bulk, les textes statiques et les sources de données ne sont pas recherchés.

The user interface

Le schéma ci-dessous illustre les principales fonctions de l'éditeur.



- | | | |
|--|---|---|
| <ul style="list-style-type: none"> 1. Aller à la liste des projets (Accueil) 2. Projet en cours > page en cours 3. Créer un nouveau formulaire, page d'accueil ou ressource 4. Recherche (alt-F) les composants de la page, dicos et archives 5. Utilitaire de débogage (pour les utilisateurs avancés) 6. Menu : Liste des projets (Accueil), import/export pages et dicos 7. Publier les modifications apportées au(x) environnement(s) 8. Première lettre de l'utilisateur connecté (vous) | <ul style="list-style-type: none"> 9. Paramètres : notes de version, langue de l'interface, nettoyage local de la base de données 10. Champs (types de variables) et éléments d'écran 11. Variables archivées pour cette page 12. Réglage de la largeur de la partie prévisualisation 13. Activer/désactiver l'icône de la corbeille pour chaque champ 14. Activer/désactiver les indicateurs de couleur des varsets 15. Propriétés de la page | <ul style="list-style-type: none"> 16. Requêtes de données et sources de données pour cette page 17. Nombre d'utilisateurs connectés actuellement sur ce projet 18. Utilisation de la mémoire disponible des variables de la page 19. Contrôle technique : le nombre d'incohérences du projet 20. Les propriétés du composant sélectionné 21. Discussion en chat pour ce composant 22. Les attributs de l'élément utilisés lors du déploiement |
|--|---|---|

Glossaire

Bulk : Élément à afficher sur une page qui permet un fonctionnement étendu. Ces fonctionnalités n'ont pas d'interface graphique simple dans Epicraft que l'utilisateur peut employer et donc sont décrites dans un format XML qui sera imputé dans le fichier ressource qui décrit une page donnée.

Champ : Permet à l'utilisateur de saisir des informations. Il existe cinq types de champs : **Texte :** permet à l'utilisateur de saisir du texte alphanumérique (a-z, 0-9). **Nombre :** permet de saisir uniquement des nombres. **Date/heure :** permet de sélectionner une date sur un calendrier et/ou l'heure. **Choix :** est un ensemble prédéterminé de réponses. **Booléen :** permet de sélectionner oui, non ou nul.

Projet Voozanoo : aussi connu sous le nom d'application, c'est l'ensemble des pages qui composent le système d'information que vous êtes en train de créer. Il peut être simple (un formulaire d'une page) ou devenir plus complexe (50 pages de formulaires et données).

Formulaire : Corresponds à ce que l'utilisateur voit et doit remplir. Un formulaire est composé de plusieurs éléments (champs) permettant à un utilisateur d'entrer des informations ainsi que des éléments organisationnels. Un nouveau formulaire définit un nouveau varset mais un même varset peut être utilisé par plusieurs pages.

Enregistrement : appelé aussi enregistrement de données. Il s'agit d'une ligne enregistrée dans une des tables de la base de données et qui regroupe les informations saisies sur une fiche.

Fiche : l'enregistrement d'un formulaire.

Page : on parle aussi du formulaire, c'est ce que l'utilisateur voit. Une page est composée de plusieurs éléments permettant à un utilisateur d'entrer des informations (champs de saisie, cases à cocher, boutons radio, menus déroulants et sélecteurs) ainsi que d'éléments organisationnels (onglets, groupes, matrices, colonnes, boutons, listings et infos-cartes). Une nouvelle

page définit un nouveau varset mais un même varset peut être utilisé par plusieurs pages.

Libellé : le texte que l'utilisateur voit à côté du champ de saisie (zone de texte, case à cocher, menu déroulant, sélecteur, etc.)

Parent - enfant : ces termes sont utilisés pour décrire la relation entre deux varsets. Un enregistrement créé dans un varset parent est lié à tous les enregistrements créés dans le varset enfant. Prenons un exemple typique : le varset *patient* contient plusieurs informations (nom, adresse, date de naissance, etc.) et constitue le varset parent ; il possède une relation avec un varset enfant appelé *visite* contenant d'autres données (poids, pression artérielle, etc.). Ainsi, un enregistrement *patient* sera lié à (ou possède) un ou plusieurs enregistrements *visite*, qui seront créés à chaque fois qu'une nouvelle fiche *visite* est saisie et sauvegardée.

Projet Epicraft : La représentation graphique et logique du système d'information qui peut être déployé sur une application Voozanoo. Il s'agit de l'ensemble des formulaires et des pages qui décrit le système d'information et la base de données.

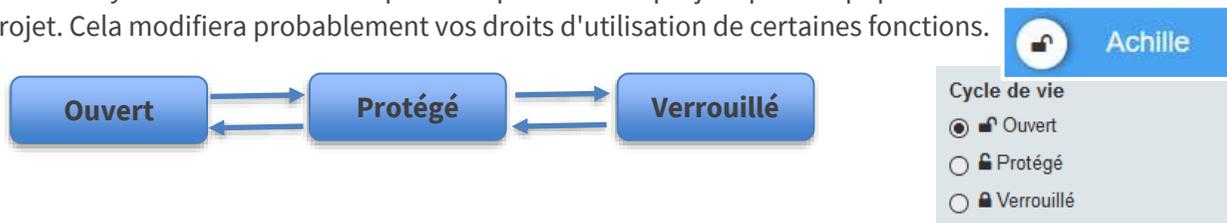
Environnement : Un projet Epicraft peut être configuré pour être déployé sur plusieurs serveurs qui peuvent se trouver dans des environnements physiques ou logiques différents. Un environnement est alors compris à partir de l'URL de l'application Voozanoo et peut être considéré comme un type tel que « staging », « pré-production » ou la « production ».

Varset : un regroupement logique de variables qui correspondent à une « table » de la base de données relationnelle sur laquelle l'application s'appuie. Un varset est souvent l'ensemble de variables d'une page mais peut être aussi pour plusieurs pages.

Voozanoo : Voozanoo est un écosystème d'applications web permettant de créer des systèmes d'information en ligne. Pour chaque projet dans une application Voozanoo, il y a plusieurs paramètres et la liste des utilisateurs et leurs droits dont un administrateur peut gérer.

En tant qu'utilisateur, votre capacité à créer, modifier ou supprimer des éléments ou à exécuter certaines fonctions sera limitée en fonction de votre rôle d'utilisateur et de l'état du cycle de vie du projet. Vous remarquerez que certaines options de menu de cette documentation manquent dans votre interface ou que des champs sont marqués d'un signe d'interdiction  rouge. Le chef de projet Epiconcept contrôle votre rôle et donc votre niveau d'accès. En fonction de votre rôle, vos limitations changeront très probablement en fonction de l'évolution de l'état du cycle de vie du projet.

L'état du cycle de vie est modifié par le responsable du projet Epiconcept pour refléter l'état réel du projet. Cela modifiera probablement vos droits d'utilisation de certaines fonctions.



Les rôles (niveaux 0 à 6) décrivent les risques autorisés dans chaque État du projet

Niveau 0 : permet uniquement de visualiser des éléments et valeurs, quel que soit l'état du projet. L'utilisateur ayant ce rôle ne peut pas modifier les champs ni exécuter des actions dans aucun des trois états du cycle de vie.

Niveau 1 : permet des actions dangereuses dans l'état Ouvert et la visualisation dans les autres états. L'utilisateur ayant ce rôle peut ajouter, modifier et supprimer des pages, des ressources, des blocs, des variables et des dictionnaires et publier pendant que le projet est Ouvert. Cependant, une fois que le projet est réglé sur Protégé ou Verrouillé, cet utilisateur ne peut que visualiser les pages, les ressources, les paramètres et les valeurs.

Niveau 2 : permet des actions dangereuses lorsque le projet est ouvert, mais seulement des actions bénignes lorsqu'il est protégé.

Le même que le niveau 1 lorsque le projet est à l'état Ouvert, sauf que cet utilisateur peut également effectuer des actions bénignes à l'état protégé,

Niveau 3 : permet des actions dangereuses lorsque le projet est dans l'état Ouvert ou Protégé, mais en affichage uniquement lorsqu'il est verrouillé.

L'utilisateur ayant ce rôle a plus de contrôle sur le projet. Ce n'est que lorsque le projet est verrouillé qu'il ne peut exécuter des actions dangereuses.

Niveaux 4 & 5 : Ces rôles sont réservés aux chefs de projet Epiconcept car ils permettent toutes les actions critiques.

Le responsable ayant ce rôle peut effectuer toutes les actions, y compris les actions critiques, sur le projet, quel que soit l'état du cycle de vie dans lequel il se trouve. Ces utilisateurs peuvent également créer des nouveaux projets.

Niveau 6 : Réservé à l'équipe qui gère le serveur Epicraft.

Explication

Niveaux de risque pour les champs et les actions

Vous remarquerez que vous serez parfois empêché de faire certaines actions. Chacune des plus de 300 actions et champs d'Epicraft ont été codés en fonction du risque qu'ils présentent à l'intégrité de la base de données du système. Par exemple, changer l'étiquette d'une variable est considéré comme bénigne. Changer l'étiquette d'un champ de **Genre** à **Sexe** ne modifie pas la façon dont les données sont ou seront stockées dans la base de données. En revanche, la possibilité de supprimer l'un des environnements est considérée comme critique et un tel droit est rarement accordé à un utilisateur. Les niveaux de risque (cumulatifs) sont indiqués ci-dessous.

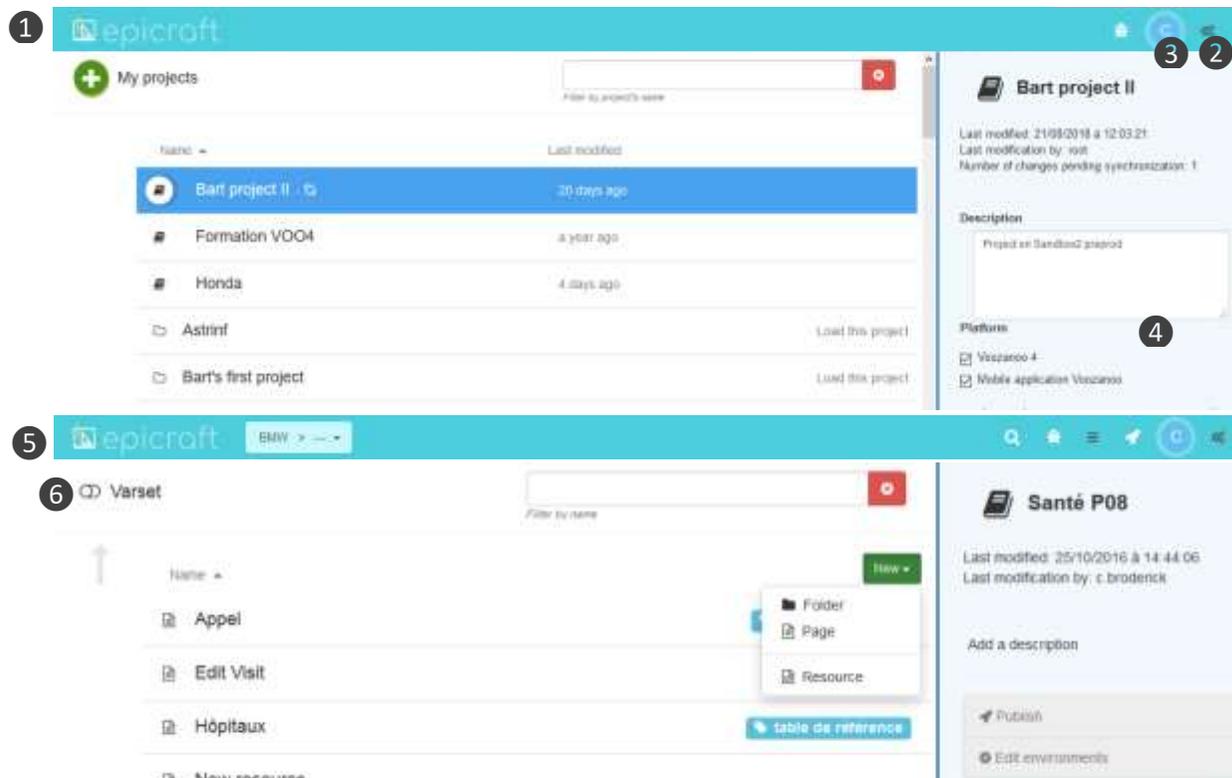
Affichage uniquement : permet de voir les valeurs et les options, mais pas de les modifier. Il n'y a aucun risque qu'un utilisateur puisse nuire au projet.

Bénigne : la modification de ces champs ne fait que changer les informations sur l'écran de l'utilisateur ou la façon dont elles sont présentées. Elle ne nuit en aucune façon à l'intégrité des données stockées ou à la structure de la base de données.

Dangereux : la modification de ces champs présente le risque de modifier la structure des données de votre application et/ou de permettre la perte de données dans la base de données.

Critique : l'exécution de ces commandes peut avoir des effets négatifs irrémédiables très graves si elles ne sont pas utilisées correctement.

Lorsque vous ouvrez Epicraft, vous entrez au niveau projet où tous les projets de votre groupe sont listés. Cliquez sur le mot Epicraft à tout moment pour revenir à ce niveau de vision générale. Double-cliquez sur un projet pour l'ouvrir et voir les pages qu'il contient.



- * Cocher la case de la plate-forme Voozanoo 4 permet de
- activer la jauge de stockage du varset (voir page 13.5) pour un projet Voozanoo 4.
 - activer des modèles Voozanoo 4 spécifiques pour créer des ressources, propriétés personnalisées et des bulks.
 - limiter la longueur des propriétés des variables afin de respecter les exigences de Voozanoo 4.

Explication

1. Affichage des projets

Double-cliquez sur un dossier de projet pour le charger dans votre navigateur, puis double-cliquez sur le projet pour l'ouvrir. Utilisez les paramètres de l'éditeur ② pour choisir la langue de l'interface.

3. Etat de la connexion

Non connecté au serveur Epicraft (toutes vos modifications sont stockées dans le stockage local). Cliquez pour vous connecter à votre compte.	Identifié et connecté au serveur Epicraft (travail en ligne). La lettre représente votre première initiale.	Le serveur traite actuellement les informations, veuillez patienter.	Connecté au serveur de projets mais la connexion avec le serveur NodeJS est perdue.

4. Plateforme et paramètres avancés

Si le projet sera publié dans un environnement* Voozanoo 4 et / ou utilisé sur un appareil mobile, cochez ces options. Voir le site web de Voozanoo pour plus d'informations sur l'application mobile et un lien vers une vidéo en ligne décrivant comment elle fonctionne avec Epicraft.

5. Affichage des pages

Un projet ouvert montre ses pages. Les dossiers n'ont aucun impact sur la structure logique ou les relations entre les varsets. En cliquant sur le bouton varset ⑥, seuls les varsets et ressources (pas les dossiers d'organisation) seront affichés. Cliquez une fois sur une page pour voir ses propriétés. Double-cliquez dessus pour la modifier.

Si vous travaillez sur de grands projets, l'organisation thématique de vos pages peut aider l'équipe à les retrouver plus aisément. Le système de dossiers ne modifie en rien la relation entre les pages ni le type de liaison qui les unie ; que ce soit au niveau de la relation Parent-enfant ou de la navigation.

Les pages et les dossiers seront toujours affichés dans l'ordre alphabétique

1. Nouveau Dossier

Crée un nouveau dossier à la racine de votre projet.

2. Homepage

Créer une page d'accueil. Référez-vous à *Pages d'accueil* pour plus de détails.

3. Form

Créer un nouveau formulaire (page) et son varset.

4. Ressource

Permet les développeurs de créer des nouveaux ressources XML (CSS, requêtes, etc.) pour l'application.

5. Déplacer dans

Déplacez une page ou un dossier dans un autre dossier en faisant glisser l'élément sur l'icône du dossier.

Dans cet exemple, l'utilisateur a fait glisser la page « Laboratoire » sur l'icône du dossier « non-utilisé ».

6. Déplacer dans un nouveau dossier

Pour déplacer deux pages dans un nouveau dossier, faites glisser une page sur l'icône d'une autre et les deux iront dans un nouveau dossier.

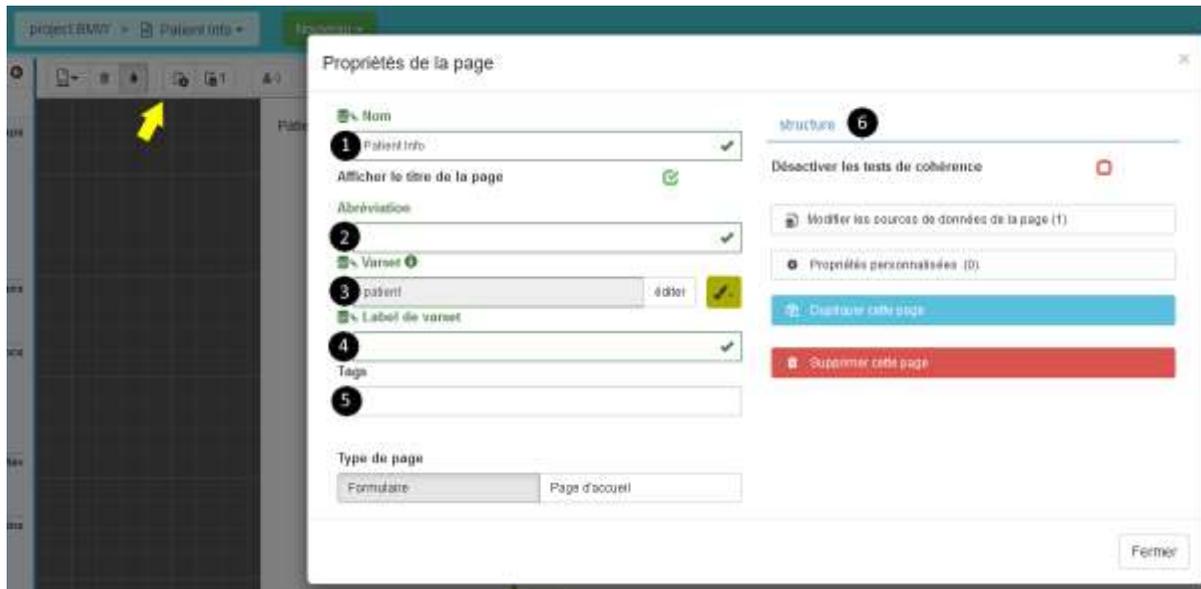
7. Filtrer

Pour retrouver une page, tapez quelques lettres pour afficher uniquement les pages dont le nom contient ces lettres. Ouvrez les dossiers pour en filtrer également le contenu. Rechercher les tags en utilisant le préfixe #.

8. Ouvrir

Pour ouvrir ou fermer un dossier, vous pouvez cliquer sur l'icône + ou - ou double-cliquer sur la ligne.

Chaque page a ses propres propriétés qui peuvent être modifiées. Une page est basée forcément sur un varset. Chaque fois on créer une nouvelle page, un nouveau varset est créé.

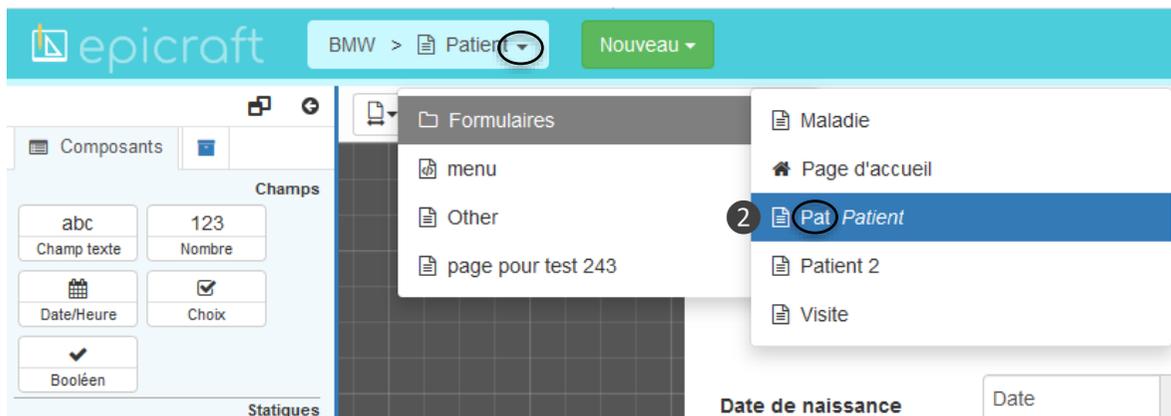


Propriétés d'une page

- 1. Nom :** Les noms des pages sont énumérés à gauche. Ce sont les noms descriptifs sans aucune restriction.
- 2. Abréviation :** Lorsqu'on recherche une page dans la liste de toutes les pages (dont les noms peuvent être longs), si l'abréviation est saisie, elle est utilisée pour aider à identifier la page. Sinon, le nom de la page est affiché dans le menu déroulant.
- 3. Varset* :** Le varset est le regroupement logique des variables qui correspondent à une « table » de la base de données relationnelle sur laquelle l'application s'appuie. Il convient de donner un nom qui vous rappelle ce regroupement de variables. Le nom ne doit pas comporter des espaces ni des caractères accentués.
- 4. Varset Label :** C'est ce qui est affiché en entête dans l'application mobile à l'écran dans le récapitulatif des données à envoyer.
- 5. Tags :** Ce sont des balises qu'on peut ajouter à une page pour aider à comprendre quel type de page il s'agit.
- 6. Structure :** Vous pouvez dupliquer, supprimer la page ici ou désactiver les contrôles de cohérence du formulaire. Pour le profil Avancé uniquement, les propriétés personnalisées permettent des ajouts à la page via XML.

Fusionner les varsets de deux pages

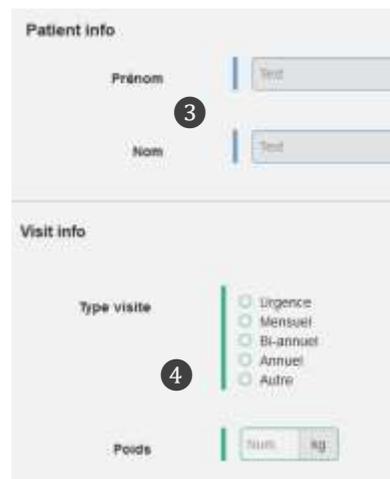
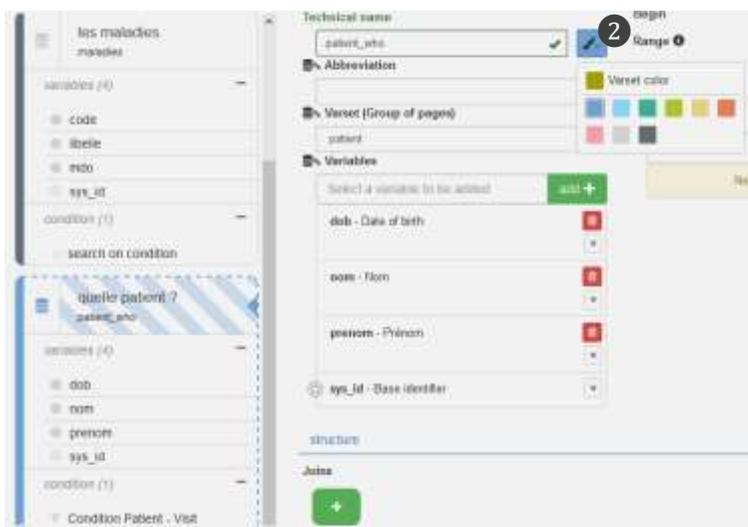
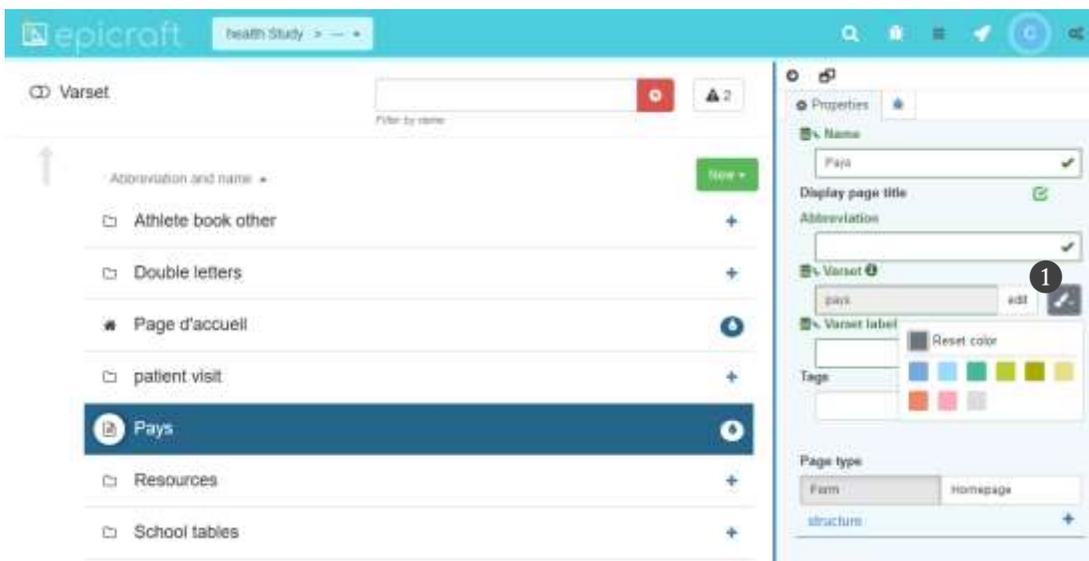
Si vous avez deux pages, chacune avec ses variables de son propre varset, on peut fusionner les deux varsets afin d'avoir un seul varset qui contient les variables des deux différentes pages. Il suffit simplement de renommer la page que vous voulez fusionner avec une autre, au nom de la dernière. Un message vous avertira du fusionnement. Cette opération fait perdre toutes les données du varset qui est renommé et donc est utilisée seulement à la construction initiale du projet.



* les noms de varset suivants sont réservés et ne peuvent pas être utilisés.

"user", "dico", "evtlog", "monitoring", "query", "queryvar", "resource", "querycache", "project", "userproject", "oauthnonces", "oauthtokens"

Certaines pages peuvent contenir des variables de différents varsets en utilisant l'interface des sources de données (voir la section *Avancé : Requêtes de données*) pour une page. Pour aider le concepteur du projet à distinguer les variables d'un varset à l'autre, Epicraft vous permet de colorer un varset ou une requête de données pour identifier visuellement à quel varset appartient une variable donnée.



Explication

Show/hide data sources colours

L'icône "cacher/afficher" les couleurs est située en haut à gauche de l'écran de la page.



1. Varsets

Dans la liste des pages, cliquez sur une page et sélectionnez une couleur dans la palette. Toutes les varsets ont la même couleur par défaut. Si le varset de la page est utilisé pour d'autres pages, vous remarquerez que ces pages changeront également à cette même couleur.

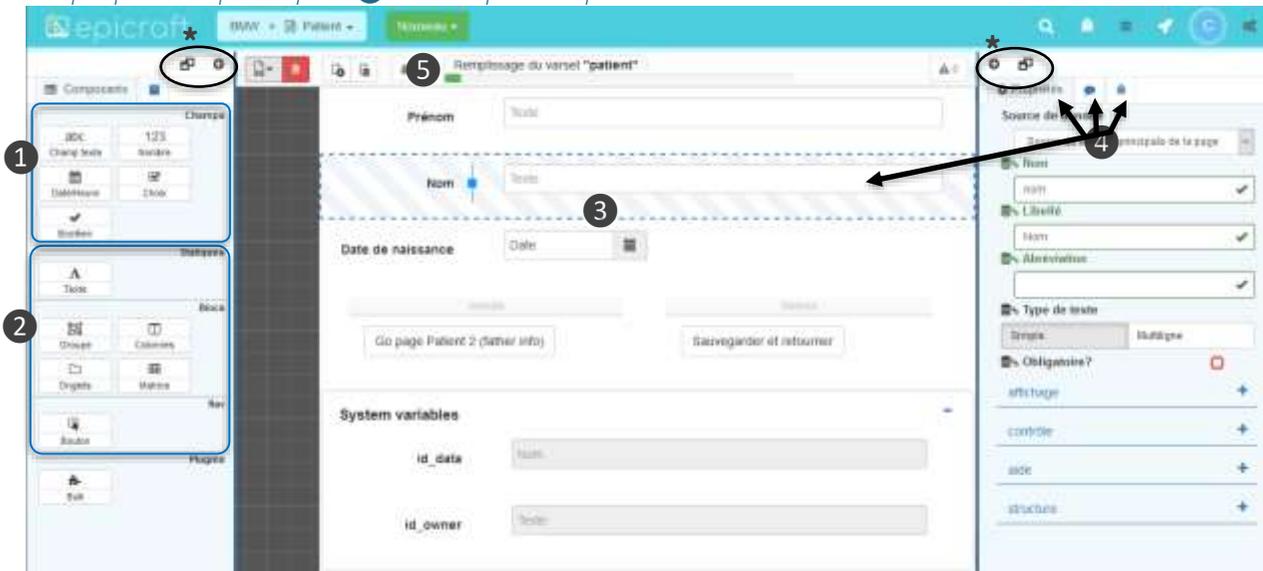
2. Requêtes de données

Si vous créez une requête de données dans les sources de données d'une page, elle prendra la même couleur que le varset qu'elle interroge. Cependant, vous pouvez passer outre cette couleur et en choisir une nouvelle. Si vous le souhaitez, vous pouvez toujours revenir à la couleur d'origine de la requête.

Apparence de la page

Lorsque vous consultez vos pages dans l'éditeur, vous verrez une barre verticale colorée devant la variable indiquant la couleur de la requête principale. Dans cet exemple de la page Visite, il y a un en-tête avec le prénom et le nom du patient qui proviennent naturellement d'une requête de données au varset Patient et ainsi on peut rapidement distinguer quelles variables de la page proviennent de la requête principale ④ ou d'une autre requête ③.

Une application est organisée par pages. Lorsque vous travaillez sur une page, vous allez ajouter des éléments pour saisir des données ① et d'autres pour en améliorer l'organisation ②. Vous verrez une prévisualisation ③ de la page (ce que l'utilisateur verra) et aurez accès aux propriétés spécifiques ④ de chaque composant.



1. Composants de saisie

Ce sont les types de variables pouvant être placés à l'écran. Si vous cliquez sur l'un d'eux, il sera inséré à la suite de l'élément en surbrillance à l'écran. Vous pouvez également les faire glisser directement vers la zone souhaitée.

2. Composants organisationnels

Ces éléments ne sont pas destinés à la saisie des données mais plutôt à l'organisation de la page.

3. Page (prévisualisation)

À part quelques exceptions mineures, vous voyez la page telle que l'utilisateur la verra. Lorsque vous cliquez sur un élément, le panneau sur la droite affiche ses propriétés. Vous pouvez faire glisser et déplacer un élément sur l'écran à votre guise. Vous pouvez sélectionner plusieurs éléments en maintenant la touche **Ctrl** du clavier enfoncée. Vous pouvez cliquer droit sur un élément pour des actions spécifiques.

4. Propriétés, commentaires et débogage

Vous pouvez ici définir en détail, pour le composant sélectionné (barres zèbres), le mode de fonctionnement de chaque élément. Du fait que plusieurs développeurs peuvent collaborer à la création d'une application, chaque élément peut porter son propre fil de conversation. L'onglet debug donne les actions techniques qui seront envoyées au projet Voozanoo pour ce composant lorsque l'utilisateur publiera ses modifications.

* Ces panneaux peuvent être fermés ou déplacés si désiré.

5. Remplissage du varset

Ceci est un indicateur visuel de la quantité d'espace de stockage utilisée pour le varset sur lequel la page est basée. Il y a une limite au nombre de variables que l'on peut créer dans un varset (il est assez grand). Cliquez sur la barre pour voir une ventilation des types de variables prenant quelle quantité d'espace de stockage.

Le champ de texte est typique de la saisie de données, il nous donne un bon aperçu des propriétés que peuvent avoir les autres éléments. D'autres éléments et champs de saisie possèdent des propriétés particulières qui seront abordées en détail dans la section qui leur est consacrée. Penchons-nous d'abord sur les propriétés communes.

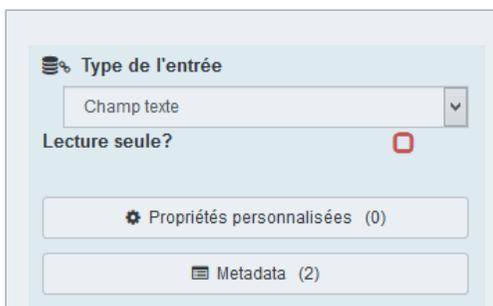
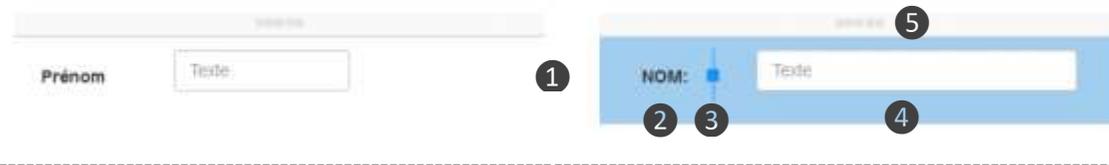
Prévisualisation

Quand un élément est ajouté (soit en cliquant dessus soit en le faisant glisser sur la page), il est mis en évidence en bleu et ses propriétés sont affichées.

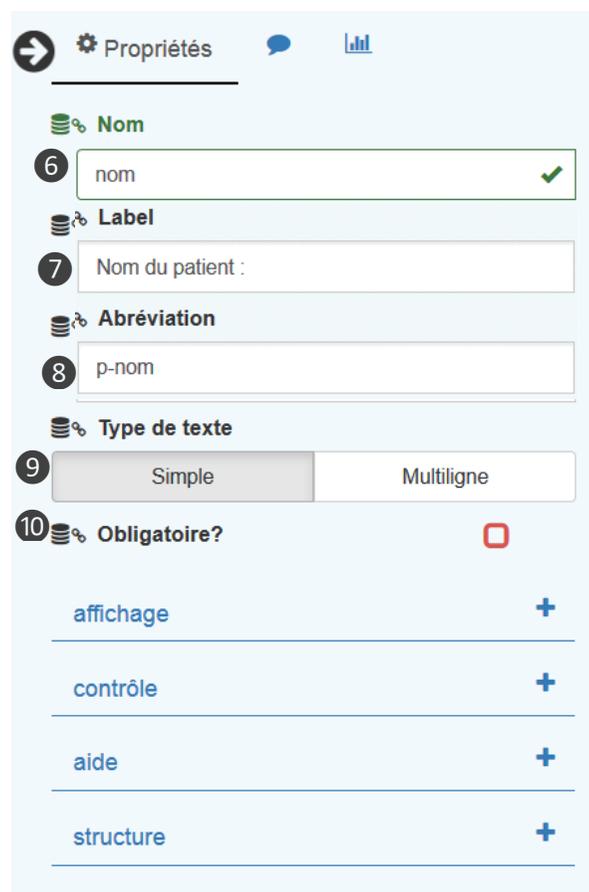
1. L'élément est sélectionné et ses propriétés (cf. ci-dessous) sont affichées.
2. Label (appelé également « question »).
3. Poignée de largeur. Faites glisser celle-ci pour ajuster la largeur du label.
4. Champ de texte. Zone de saisie des réponses.
5. Les points indiquent que le champ s'insère dans une colonne (un élément organisationnel).

Propriétés de base

6. **Nom** : Le nom qui sera utilisé dans la base de données. Il ne peut contenir que des caractères alphanumériques (a-z, 0-9) et des tirets bas _, non des caractères accentués.
7. **Label** : Le texte que l'utilisateur voit devant le champ de saisie. C'est aussi appelé « la question ».
8. **Abréviation** : Un autre nom (généralement plus court) affiché comme entêtes des listings et exports.
9. **Type de texte** : Spécifique pour le champ de texte, l'utilisateur verra soit une ligne unique d'une capacité max. de 255 caractères ou une zone de texte pouvant en accepter jusqu'à 65536.
10. **Obligatoire** : Si cette option est cochée, l'utilisateur doit rentrer une réponse (saisie de données, cocher une case, sélectionner une date, etc.). Sinon, la fiche ne peut pas être sauvegardée.



Metadata : pour chaque variable, des métadonnées peuvent être ajoutés. Ces informations (clé : valeur) sont comme les propriétés en plus des propriétés standards. Elles peuvent être exploitées par des fonctions ou actions spécifiquement programmé par un développeur.



Remarque : Les icônes de la base de données avec une chaîne indique qu'ils sont les noms des propriétés dans la base de données MvSOI de l'application.

Toujours avec le même champ texte, voyons à présent comment les propriétés d'affichage permettent de définir le mode de visualisation à l'écran.

Ressource CSS

```

1 .blue-text{
2   color: #33F;
3 }
4
5 #attention1 {
6   font-size: 150%;
7   background: yellow;
8 }
9

```

Résultat HTML

```

<div id="blue_group" class="col-xs-12">
  <div id="pname" class="form-group blue-text">
    ::before
    <label id="vzn5b97845ae031f9_56671834" class="text-right style="">
      <span data-field="nom">Nom :</span>
    </label>
    <div class="field col-md-9 text-left">
      <input id="vzn5b97845ae03404_07178345" class="name="vzn5b97845ae03404_07178345" value="" minlength="3" type="text">
    </div>
    ::after
  </div>
</div>

```

Résultat visuel

Propriétés Commentaires (0)

- Nom**: nom ✓
- Label**: NOM:
- Type de texte**: Inline Multiline
- Obligatoire?**
- display**: —
- Specific Label** ⓘ
- 1 Patient name
- 2 Largeur du label: 3
- 3 Cacher le label?
- 4 Position du label: Au dessus En ligne
- Classes CSS**
- 5 blue-text
- Id spécifique** ⓘ
- 6 pname
- 7 Traduction des valeurs saisies?

Propriétés d'affichage

1. Label spécifique : Dans les rares cas où vous auriez placé une variable à deux endroits différents de l'écran ou dans différents onglets ou pages, vous pouvez attribuer un label spécifique qui remplacera le label original.

2. Largeur du label : Largeur du label en douzièmes de largeur d'écran. La largeur de l'écran est divisée en 12 colonnes transparentes. Afin de permettre un alignement aisé des questions et de leurs champs de saisie, on détermine le nombre de largeurs de colonne que le label doit occuper. Si le label est trop long pour l'espace donné, il créera automatiquement une deuxième ou troisième ligne. La valeur par défaut de 3 signifie que l'espace pour le label est limité à 25% de l'écran, marge à gauche.

3. Cacher le label ? : Si les circonstances l'imposent, vous pouvez masquer le label d'un élément ou d'une variable.

4. Position du label : Par défaut, le label précède le champ ou les options sur la même ligne. Sélectionnez la position *Au dessus* pour placer le label au-dessus du champ. Ce dernier est recommandé si le formulaire est saisi dans l'application Voozanoo mobile.

5. Classes CSS : Ajoutez les noms des classes CSS que vous souhaitez spécifiquement appliquer à ce contrôle.

6. ID spécifique : attribue à l'élément un attribut HTML id spécifique.

7. Traduction des valeurs saisies : en cochant cette case, les valeurs saisies dans le formulaire peuvent être traduites. Ceci est généralement pour une table de référence dont les données saisies doivent être traduites.

Toujours avec le même champ texte, voyons comment les propriétés de contrôle vous permettent de limiter l'affichage d'un composant ainsi que les possibilités de saisie.

Nom ✓

Label ✓

Abréviation ✓

Type de texte
 Simple Multiligne

Obligatoire?

affichage

contrôle

1 **Longueur**

2 **Afficher cet élément quand...** ✓

3 **Garder les données si le champ est caché?**

4 **Valeur par défaut**

aide

Propriétés de contrôle

1. Longueur : Pour les champs texte et nombre, cette valeur définit le nombre maximum de caractères autorisés pour la saisie.

2. Afficher cet élément quand... : Il s'agit de la condition à remplir (écrite en JavaScript) pour afficher cet élément à l'écran. Cette option est utile pour afficher, sous certaines conditions, une variable ou un groupe d'éléments contenant plusieurs variables. Si par exemple trois questions concernent seulement les femmes (et si le sexe du répondant a déjà été renseigné), vous pourriez créer un groupe pour y placer ces trois questions et conditionner son affichage à la contrainte suivante : `{sex}==0` (basé sur les valeurs du code de dictionnaire 0 pour les femmes et 1 pour les hommes). Pour en savoir plus sur les conditions JavaScript, consultez la section *Affichage conditionnel : Instructions JavaScript*.

3. Garder les données masquées : Lorsqu'une valeur est saisie dans un champ qui a été révélé, mais qui a ensuite été cachée à cause d'une action de l'utilisateur, souhaitez-vous conserver les données précédentes ou les effacer ? Cochez cette case pour conserver les données dans un groupe caché.

4. Valeur par défaut : Une propriété à utiliser avec parcimonie et qui consiste à prédéfinir la valeur d'une variable. Elle peut bien entendu être modifiée par l'utilisateur. Logiquement, cette option peut être risquée car il est impossible de savoir à la lecture des données si ce point a été négligé par l'utilisateur ou s'il voulait effectivement répondre de la sorte.

Les propriétés Aide et Structure guident l'utilisateur vers le type de réponse souhaité et permettent de définir le type de variable le plus approprié.

aide -

Préfixe

1

Suffixe

mm

Placeholder

2 ex: DuPont

Aide à la saisie

3 Saisissez seulement le nom de famille

structure -

4 Type de l'entrée

5 Champ texte

Lecture seule?

6 Propriétés personnalisées (0)

7 Metadata (0)

Propriétés Aide

1. **Préfixe / suffixe** : Typiquement pour les entrées numériques, ces champs permettent d'afficher un préfixe ou un suffixe dans le champ (voir l'exemple à gauche).
2. **Placeholder** : Pour les champs texte et nombre, ce que vous placez ici sera affiché en gris clair dans la zone de saisie. C'est un moyen de leur rappeler le type de réponse que vous attendez.
3. **Aide à la saisie** : Ce texte apparaîtra au-dessus du libellé quand la souris de l'utilisateur passera dessus. Utilisez cette option pour donner plus de détails ou des instructions à propos de la question.

Propriétés Structure

4. **Type de l'entrée** : Vous pouvez modifier le type de la variable mais alors, la plupart des propriétés vont changer pour s'adapter à l'autre type de variable.
5. **Lecture seule ?** : Si elle est cochée, la valeur de la variable reste lisible mais elle est protégée en écriture. Une propriété utile si vous voulez, par exemple, rappeler à l'utilisateur la valeur d'une variable déjà saisie dans le questionnaire. Vous ajoutez votre variable comme d'habitude et, si celle-ci doit réapparaître, vous la rajoutez mais cette fois en cochant *Lecture seule*.
6. **Propriétés personnalisées** : disponible uniquement pour le profil « Avancé », il est destiné aux développeurs, pour insérer des lignes de codes supplémentaires (XML, JavaScript ou Markdown).
7. **Métadonnées** : permet d'ajouter n'importe quel nombre de paires **nom : valeur** en tant que métadonnées à la ressource varset.

123
Nombre

Les variables numériques possèdent des propriétés spécifiques. Pour toutes les autres propriétés, voir les chapitres Général, Affichage, Contrôle, Aide et Structure ci-dessus.

Pourcentage de temps en dehors du pays principal :

NB : Pour les variables de type entier, il est possible de présenter l'utilisateur avec un « curseur à glisser » au lieu d'un champ de saisie. Voir le document en ligne **Astuces Epicraft** via bit.ly/voodocs.

Exemples d'utilisation des icônes Font Awesome et Glyphicon. À gauche, la valeur saisie dans les propriétés "Aide" du champ affiché dans Epicraft. À droite, leur apparition sur la page Voozanoo.

Propriétés spécifiques des champs numériques

1. Sous-type : Sélectionnez le type de valeur souhaitée, *Entier* ou *Décimal*.

Attention : Les valeurs entières sont limitées à environ +/- 2 milliards. En outre, le type entier ne doit pas être utilisé pour les codes postaux, puisque la valeur 01610 sera sauvegardée en 1610. Utilisez une variable texte dans ce cas.

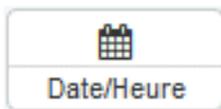
2. Précision : Si le sous-type est défini comme *Décimal*, vous pouvez préciser le nombre de décimales après la virgule que l'utilisateur peut saisir.

3. Minimum : Vous pouvez définir la valeur minimale acceptable pour la valeur saisie. Si l'utilisateur tente d'entrer une valeur inférieure à ce nombre, un message d'erreur est affiché et l'enregistrement devient impossible.

4. Maximum : Vous pouvez définir la valeur maximale acceptable pour la valeur saisie. Si l'utilisateur tente d'entrer une valeur supérieure à ce nombre, un message d'erreur est affiché et l'enregistrement devient impossible.

Remarque : Si vous avez besoin de créer des contraintes plus spécifiques pour les valeurs admissibles, par exemple des valeurs dépendant d'autres variables, vous pouvez utiliser les fonctions MySQL décrit dans la section de *Contrôle de cohérence* de la documentation *Fiches de référence Voozanoo 4*.

5. Préfixe et suffixe : Le préfixe et le suffixe peuvent contenir du texte, une icône ou des informations mixtes pour faciliter la compréhension du champ de saisie. Vous pouvez utiliser les icônes gratuites des sites web Font Awesome et Glyphicon.



Les variables date possèdent aussi quelques propriétés spécifiques. Pour toutes les autres propriétés, voir les chapitres Général, Affichage, Contrôle, Aide et Structure ci-dessus.

1 **Sous-type**

Date

Date

Date et heure

Heure

control

2 **Min**

Maintenant 01/07/2016

3 **Max**

Maintenant 31/12/2016

4 **Valeur par défaut**

Maintenant

Afficher cet élément quand... ⓘ

Variables de système

Il est possible d'afficher les variables du système sur une page si désiré. Les « sys vars » sont écrites par Voozanoo pour garder des informations sur une fiche. Pour afficher un de ces informations, par exemple la date de création de la fiche, il suffit d'ajouter une variable sur la page et le nommer sys_creation_date. EpiCraft va forcer cette variable à « lecture seule » et elle apparaîtra sur la page.

Les sys vars sont : id_data (la valeur ordinale et identité unique d'une fiche), sys_creation_date (date et heure à laquelle fiche a été créée), sys_last_mod_date (la date et heure à laquelle la fiche a été modifiée), id_owner (l'identifiant numérique de l'utilisateur qui a créé la fiche), sys_last_mod_user (l'identifiant numérique du dernier utilisateur qui a modifié la fiche). Attention : l'ajout de la variable sys_group (le(s) groupe(s) auquel appartient cet enregistrement) à une page a la particularité de demander à l'utilisateur de sélectionner le groupe souhaité dans lequel l'enregistrement sera stocké et de ne pas simplement afficher le nom du groupe.

Propriétés spécifiques des champs date

1. Sous-type : Détermine si la donnée saisie doit être une date, une date-heure (**Fonction beta**: disponibilité limitée), ou une heure seulement.

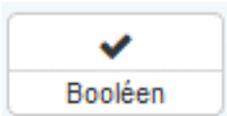
2. Minimum : Vous pouvez définir la valeur la plus ancienne acceptable pour la valeur saisie. Si l'utilisateur tente d'entrer une valeur antérieure à cette date, date-heure ou heure, un message d'erreur est affiché et l'enregistrement devient impossible.

3. Maximum : Vous pouvez définir la valeur la plus récente acceptable pour la valeur saisie. Si l'utilisateur tente d'entrer une valeur postérieure à cette date, date-heure ou heure, un message d'erreur est affiché et l'enregistrement devient impossible.

4. Valeur par défaut : Cette valeur sera placée dans le champ et acceptée comme réponse de l'utilisateur s'il ne la modifie pas.

Maintenant : En cochant *Maintenant* pour les valeurs minimum, maximum ou par défaut, l'enregistrement retiendra la date, la date-heure ou l'heure à laquelle l'utilisateur a ouvert la page.

Remarque : si vous avez besoin de créer des contraintes plus spécifiques pour les valeurs admissibles, par exemple des valeurs dépendant d'autres variables ou une date « aujourd'hui plus 30 jours » ou afficher des messages d'erreur spécifiques, vous pouvez utiliser les fonctions MySQL qui sont expliquées dans la section *Contrôle de cohérence* de la documentation *Fiches de référence Voozanoo 4*.



Le type booléen ne possède pas de propriétés spécifiques au-delà de celles de base. Il est cependant important de savoir comment il fonctionne. Pour toutes les autres propriétés, voir les chapitres Général, Affichage, Contrôle, Helper et Structure ci-dessus.



Ce que l'utilisateur voit

Bien qu'une valeur booléenne puisse être soit vraie soit fausse, l'utilisateur peut en fait opter pour trois états. Par défaut, la valeur d'une variable booléenne est null. Elle reste ainsi dans la base de données si elle n'est pas modifiée. Les propriétés permettent de définir la valeur par défaut sur vrai ou faux en fonction des besoins du développeur.

L'utilisateur peut répondre à la question par



ou pour vrai (true)



ou pour faux (false)

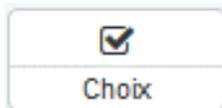


ou pour null

En fonction des directives mises en place dans l'environnement de publication du projet, les valeurs peuvent être affichées sous forme d'icônes dans la colonne de gauche ci-dessus. Pour afficher les valeurs dans le nouveau style vert / rouge, procédez comme suit.

1. Ajoutez une propriété personnalisée à la variable booléenne.
2. Nommez-le **form.widget.value.option**
3. Entrez la déclaration XML suivante:

```
<option output = "html" option_name = "widget" value = "WidgetBoolean" />
```



L'ensemble prédéterminé de réponses (communément appelé choix multiple) repose sur une liste de paires réponse/code appelée dictionnaire ou, de façon plus informelle, "dico". Parce que les dictionnaires font partie intégrante des systèmes d'information et de leur gestion, une toute nouvelle interface de dictionnaire a été ajoutée à Epicraft en version 1.18.

Type de choix

1 Simple Multiple

Dictionnaire

2 genre_film 3 Edition

Réponses actives

DRA	Drame
COM	Comédie
DOC	Documentaire
HIS	Historique

7

Dictionary Description 6 Dictionary Usage 3

Name of this dico

genre_film ✓

Active Responses (4)

DRA	Drame
COM	Comédie
DOC	Documentaire
HIS	Historique

4

Archived Responses (2)

ACT	Action	🗑️
POL	Policier	🗑️

6

Add one or more responses

5 [code] Label +

Sort responses

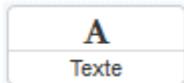
By code By label

Be careful: The dictionary is used 3 times. Any modification will impact all those fields and variables.

Explication

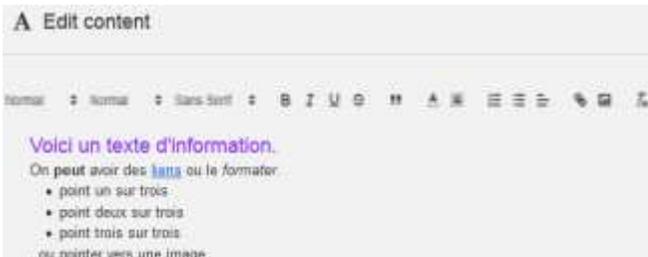
- Type de choix** : « Simple » pour une réponse unique par bouton radio ou via le menu déroulant. « Multiple » pour afficher des cases à cocher ; l'utilisateur coche autant de réponses qu'il veut.
- Dictionnaire partagé** : Vous pouvez sélectionner un dico déjà existant pour votre variable. Un des dictionnaires souvent réutilisés est celui pour les réponses Oui/Non/NSP.
- Edition** : Si vous n'utilisez pas un dictionnaire partagé, vous devez éditer un nouveau dictionnaire. Vous pouvez également importer des dictionnaires à partir d'un tableur. Voir la section *Importer des dictionnaires*.
- Éléments de dictionnaire** : Éléments disponibles pour l'utilisateur final. Le nombre 6 sur l'onglet indique le nombre d'éléments du dictionnaire. L'ordre affiché est l'ordre dans lequel ils seront affichés. Vous pouvez glisser-déposer des éléments pour modifier leur position.
- Ajouter des réponses** : Taper ou coller des éléments du dictionnaire (Epicraft assignera des codes si vous n'en spécifiez pas).
- Réponses archivées** : Réponses qui ont pu être utilisées à un moment donné, mais qui ne sont plus nécessaires. Faites glisser un dico de la colonne de gauche vers cette colonne pour l'archiver. La conservation des dicos archivés permet aux anciennes fiches de conserver leurs valeurs. Pour des raisons de sécurité, il faut archiver un dico avant de pouvoir le supprimer définitivement et ainsi faire d'anciens enregistrements erronés.
- Utilisation du dictionnaire** : Cliquer sur cet onglet pour voir tous les endroits où ce dico est utilisé. Vous pouvez cliquer sur le nom de la variable et être transporté jusqu'à la page où se trouve le dictionnaire. Le nombre 3 sur l'onglet indique le nombre de fois que ce dictionnaire est utilisé dans votre projet.
- Obtenir des dictionnaires** : Cliquer sur cette icône pour placer les éléments du dictionnaire avec leurs codes dans le presse-papiers du système.

NB : Il est possible d'afficher les réponses dans un ordre aléatoire chaque fois que la question est présentée. Voir le document en ligne **Astuces Epicraft** à www2.voozanoo.net.



On peut placer des textes, des liens, des images, variables ou fonctions des variables dans la page. Ce n'est pas un champ de saisie de données mais plutôt une fonction pour l'affichage.

1



2



2



1. L'éditeur de texte riche

Ce composant est un éditeur de texte riche qui vous permet de disposer sur la page des textes, des liens, des listes et des images à votre gré.

2. Utilisation avancée

Réservé aux utilisateurs avancés, il existe une fonction intégrée à l'éditeur qui permet d'accéder à toute donnée provenant d'une requête de données dans la page via son nom `{nom_de_la_requete.varname}`. On peut afficher ces données sur d'autres parties de l'écran ou afficher les résultats d'une opération JavaScript sur la (les) variable(s). En utilisant le tilde ~ comme le préfixe et le suffixe de la variable ou l'instruction JavaScript comme indiqué, au lieu de l'afficher comme texte, Voozanoo l'évaluera.

NB : Si l'on met en forme le texte de la variable (gras, italique, etc.), veillez à inclure les tildes dans la mise en forme.



Groupe

Les groupes sont utilisés pour organiser de façon claire un groupe de champs de saisie. Plus important encore, ils permettent de conditionner l'affichage des questions qu'ils contiennent en fonction de certaines conditions.

Vu par l'utilisateur avec un titre et sans transparence.

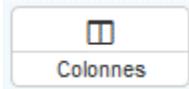
Vu par l'utilisateur sans titre et avec la transparence activée.

Titre et Transparence

1. Titre : Affiche le titre au-dessus du groupe.

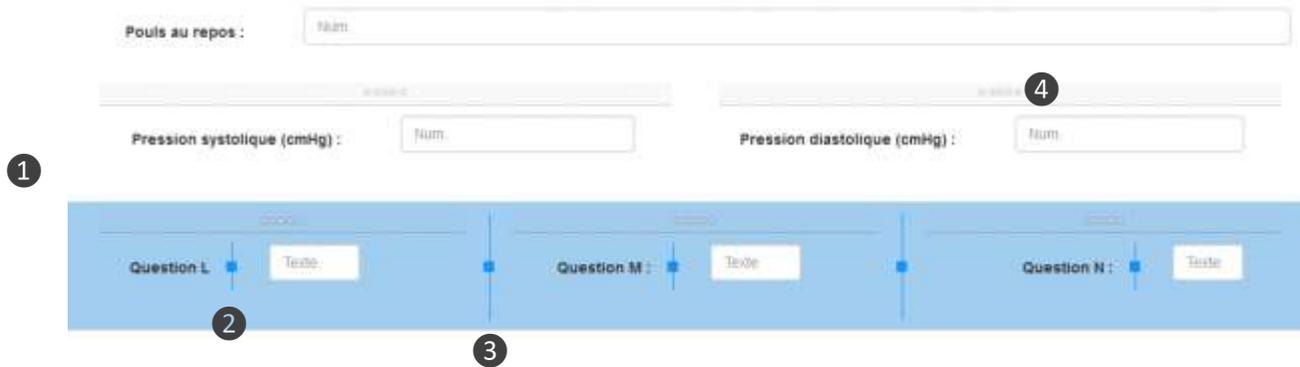
2. Description : Vous permet de donner une description du groupe qui ne sera visible que dans Epicraft et non sur la page elle-même. Ceci est utile pour les groupes qui ne sont pas étiquetés avec un titre. Il permet au groupe d'être identifié dans Epicraft pour les auteurs.

3. Est transparent ? : Si cette option est cochée, aucune ligne de contour ne vient délimiter le groupe qui devient dès lors moins évident. Voir la section *Affichage conditionnel : Instructions JavaScript* pour des exemples de conditions qui affichent le groupe lorsque la condition est vraie.



Cet élément organisationnel vous permet de créer des colonnes. Une fois mises en place, vous pouvez faire glisser les questions vers n'importe quelle colonne.

Vu dans l'éditeur

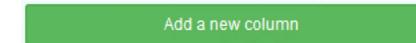


Vu par l'utilisateur

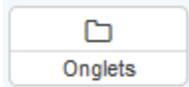


Colonnes

Une fois le conteneur mis en place, vous pouvez ajouter d'autres colonnes avec ce bouton situé dans les propriétés.



- 1. Exemple :** Vue d'une, puis de deux et trois colonnes (en surbrillance) dans l'éditeur.
- 2. Poignée du label :** La poignée de la colonne détermine le nombre de douzièmes de colonne que le label utilisera en largeur. Elle peut être déplacée à la souris ; vous pouvez également entrer une valeur comprise entre 1 et 11 dans la propriété *Largeur du label*. Ainsi, si elle est paramétrée sur 6 unités, elle occupera la moitié de cette colonne.
- 3. Poignée de colonne :** Ces poignées peuvent être déplacées, au besoin, pour définir la largeur des colonnes.
- 4. Indicateur de colonne :** Les cinq points indiquent la colonne.



Les onglets sont des conteneurs destinés à l'organisation ; ils permettent à l'utilisateur d'accéder à différentes parties thématiques ou logiques d'une page.

Vu dans l'éditeur

Vu par l'utilisateur

Onglets

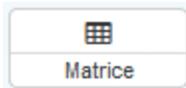
Un onglet peut intégrer toute une page (si vous y placez toutes vos variables) ou n'en représenter qu'une petite partie (en affichant une sélection des variables de la page). Chaque onglet est fixe et son affichage ne peut être soumis à condition. Chaque onglet est une sorte de conteneur dans lequel les autres éléments peuvent être glissés et déposés. Cliquez sur le bouton + pour ajouter un autre onglet.

Pour supprimer un onglet, sélectionnez-le puis appuyez sur la touche Suppr du clavier (**Attention** : toutes les variables à l'intérieur d'un onglet seront également supprimées).

Note sur l'utilisation des colonnes dans l'app mobile

Si votre page est utilisée à partir de l'application mobile Voozanoo, les onglets s'affichent correctement s'ils contiennent toutes les variables et les boutons de la page. S'ils ne le font pas, ils seront affichés comme des groupes.

Exemple de deux onglets sur une page de l'application mobile



Les matrices permettent de mieux présenter les variables d'un dictionnaire. Au lieu d'avoir, pour chaque réponse, une question et son label, la matrice vous permet d'avoir l'ensemble des en-têtes et plusieurs rangées de questions.

2 Matrix 1 questions

	Monday	Tuesday	Wednesday	Thursday	Friday
Go to a movie	<input type="radio"/>				
1 Go to a restaurant	<input type="radio"/>				
Exercise	<input type="radio"/>				
Go to work	<input type="radio"/>				

Text 1 Text 2*

abc Text field	123 Number
 Date/Time	 Choice
 Boolean	 Searchfield
 Text	
 Group	 Columns
 Tabs	 Matrix 1

display

Description

2 Matrix 1 questions

CSS class

display

Specific Label ⓘ

3 Label width 3

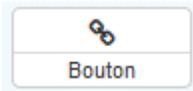
Add a columns offset 2

Les matrices

La matrice 1 est un conteneur* qui contient des variables type **Choix** d'un dictionnaire unique. Le dictionnaire de la matrice est défini comme la sélection du dictionnaire de la première variable **Choix** qui est placée à l'intérieur de la matrice. Après cette définition, seules les variables avec le même dictionnaire peuvent être ajoutées à la matrice. Tout nouveau composant **Choix** qui est ajouté à cette matrice prendra le dictionnaire déjà défini pour la matrice. Vous pouvez changer le dictionnaire d'une matrice mais en conséquence cela changera toutes les variables de cet ensemble des choix. Par exemple, à gauche, il peut s'avérer nécessaire de remplacer le dictionnaire de cette matrice par un qui inclut les jours du week-end.

La description de la matrice 2 n'apparaît que dans Epicraft (pour vous aider à identifier le groupe) et non sur l'écran de l'utilisateur.

Pour un effet visuel plus agréable, vous pouvez décaler 3 (ajouter l'espacement à droite) une variable donnée dans la matrice (qui sera attribuée à toutes les variables). La largeur du label fonctionne normalement.



Les boutons sont utilisés pour sauvegarder les données saisies, accéder à une autre page, revenir à une page précédente ou revenir à la page d'accueil.

Label

Sauvegarder et aller à la page B

1 Action: enregistrer les données?

2 **Redirection**

Aller à la page...

Retourner à la page précédente

Aller à la page d'accueil

3 Aucune

4 **Page**

Three page B

Garder le contexte? (garder la clé primaire)

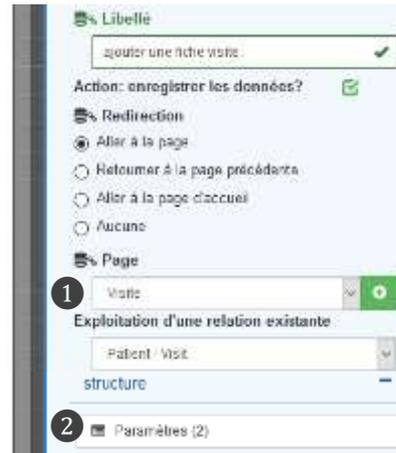
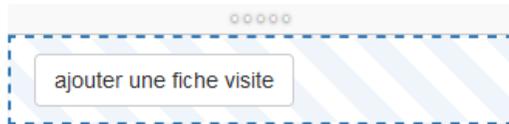
Vu par l'utilisateur

Boutons

1. **Enregistrer les données** : Sauvegarder les données saisies dans la base de données puis se rediriger vers une autre page.
2. **Redirection** :
 - **Aller à la page...** : Aller à la page indiquée dans 3.
 - **Retourner à la page précédente** : Aller à la page précédente.
 - **Aller à la page d'accueil** : Aller à la page d'accueil pour le rôle de l'utilisateur.
 - **Aucune** : n'aller à aucune page.
3. **Page** : Choisissez la page de destination si "Aller à la page..." a été sélectionné.
4. **Gardez le contexte ?** Lorsque vous avez plusieurs pages basées sur le même varset, le choix de passer à une autre page avec ce même varset affichera cet élément. Cela devra être coché dans la plupart des cas. Ne le décochez que pour des cas particuliers où vous ne voulez pas que chaque page ait la même clef primaire (pare exemple quand on crée un bouton « Enregistrer et remplir une nouvelle fiche » qui va à la même page).



Lorsqu'un bouton envoie l'utilisateur à une page qui a une relation 1-N (voir la section *Avancé : Création de liens entre les tables*) avec le varset principal de la page en cours (la page cible étant le N), un autre champ sera affiché vous permettant d'envoyer des valeurs qui prérempliront les champs du varset principal de la page cible. Ceci est généralement utilisé pour envoyer la valeur de la clé étrangère à un varset enfant pour les boutons du type « ajouter un nouvel événement lié à cet enregistrement », par exemple, « Ajouter une fiche visite [pour ce patient] ».



1. Une page liée

Lorsque le bouton pointe sur une page qui a une relation avec ce premier, le champ "Exploiter la relation" apparaît. Si l'on choisit de l'utiliser (comme c'est le cas dans l'exemple ici), l'*id_data* de l'enregistrement en cours sera envoyé à la page cible et préremplira le champ "Clé étrangère". Ainsi, ce nouvel enregistrement aura un lien avec son enregistrement parent.

2. Les paramètres

Par défaut, l'*id_data* de la page actuelle est envoyée à la page cible mais on peut envoyer n'importe quelle valeur à la requête principale de la page cible. L'exemple d'interface de passage de valeur présenté ici (cas d'utilisation typique), montre le passage de la valeur initiale (créée automatiquement) ainsi qu'une valeur constante qui a été ajoutée manuellement.

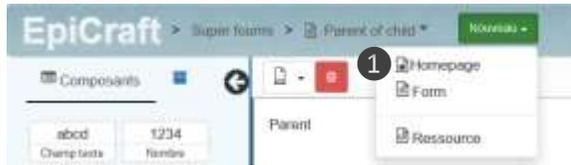
Glissez et déposez les points finaux pour relier les valeurs de la page source à la page cible. Pour créer un nouveau paramètre à transmettre, il suffit de cliquer sur le nom de la variable sur le côté gauche. Une nouvelle variable apparaîtra. Vous pouvez également modifier le nom de la variable utilisée via l'icône du crayon.

3. Réception de la valeur

Vous pouvez voir sur la requête principale de la page cible qu'elle préremplit certains champs. Par exemple, la requête de données principales de la page Visite dans notre exemple ressemble à ceci.



Une page d'accueil est la première page à laquelle l'utilisateur est envoyé lorsqu'il se connecte. La page d'accueil à laquelle l'utilisateur est envoyé est déterminée par son rôle.



Explication

1. Créer une page d'accueil

Commencez par le bouton **Nouveau > Page d'accueil**.

2. Propriétés

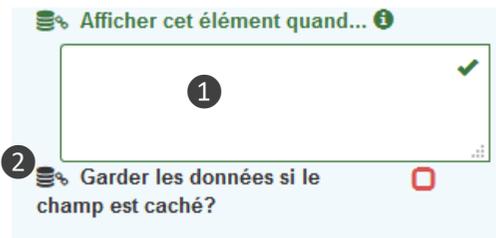
- **Nom, Abréviation, Varsset** : Donnez à la page d'accueil un nom (et une abréviation si vous le souhaitez). Modifiez le nom du varset par quelque chose de compréhensible.
- **Varsset Label** : Donnez un libellé pour le varset (si vous allez utiliser cette page sur l'application mobile).
- **Tags** : Entrez des balises si vous le souhaitez.
- **Rôles** : Entrez-le ou les rôles d'utilisateur définis dans l'application Voozanoo qui doivent être dirigés vers cette page d'accueil après la connexion. Dans l'exemple à gauche, tout utilisateur ayant le rôle de **medecins** ou **infirmieres** sera dirigé vers cette page d'accueil après s'être authentifié.
- **Destination** : Vérifiez la (les) destination(s) vers laquelle vous souhaitez diriger cette page d'accueil.

NB : Etant donné que la page d'accueil est liée à un varset, s'il n'y a que des boutons sur la page, assurez-vous de décocher "Enregistrer les données" pour chacun des boutons. Sinon, vous ne pourrez pas aller dans d'autres pages.

Si un utilisateur a plus d'un rôle, lors de l'ouverture de session, il sera envoyé à la première page d'accueil qui lui a assigné un de ses rôles.

Si une page d'accueil n'a aucun rôle qui lui est attribué, ce sera celle des personnes dont le rôle n'en a pas.

Lorsque l'instruction JavaScript évaluée est vraie, l'élément est affiché à l'utilisateur. Dans le cas contraire, il ne l'est pas.



<code>{sexe} == 1</code>	; si le code de l'élément sélectionné dans le dictionnaire est 1.
<code>{sexe} == "F"</code>	; si le code de l'élément sélectionné dans le dictionnaire est F.
<code>{couleur}.indexOf('4') != -1</code>	; si l'élément d'un dictionnaire avec le code 4 a été coché.
<code>{variable_name}.indexOf('value') != -1</code>	; la formulation générale pour case à cocher
<code>{ddn} > "1999-12-31"</code>	; si la date est postérieure au 31-12-1999.
<code>{nom_variable} comparaison "aaaa-mm-jj"</code>	; comparaison générique pour les dates
<code>{ddn} > "1999-12-31" && {sexe} == 0</code>	; si les conditions sont toutes deux vraies (ET logique)
<code>{surg_hr} > "17:59:59"</code>	; si la variable heure <i>surg_hr</i> est postérieure à 17:59:59
<code>{accord} == 1 {accord} == 2</code>	; si au moins une des deux conditions est vraie (OU logique)
<code>!{temp}</code>	; si la variable <i>temp</i> est vide (est null)
<code>!!{temp}</code>	; si la variable <i>temp</i> possède une valeur (n'est pas null)
<code>{bmi} >= 25</code>	; si la variable décimale <i>bmi</i> est égal à ou supérieure à 25
<code>{name}.length > 5</code>	; si la variable texte <i>name</i> comporte plus de cinq caractères
<code>{fumeur} == true</code>	; si la variable booléenne <i>fumeur</i> est vrai
<code>{user->roles}.indexOf('medecin') != -1</code>	; si le rôle de l'utilisateur est <i>medecin</i> .
<code>{user->groups}.indexOf('Ile de France') != -1</code>	; si le groupe actuel de l'utilisateur est <i>Ile de France</i> .

Affichages conditionnels

1. La plupart des éléments possèdent une propriété d'affichage conditionnel. La condition est écrite en JavaScript. A gauche, vous verrez quelques exemples de déclarations qui affichent l'élément à l'écran lorsque l'évaluation de la condition résulte vraie.

Pour plus d'informations sur la syntaxe JavaScript, consultez *JavaScript Comparisons* à l'adresse : http://www.w3schools.com/js/js_comparisons.asp.

2. Normalement, si la condition est fausse, vous voudriez que la variable cachée prenne une valeur nulle. Si dans le cas cependant vous souhaitez conserver une valeur entrée dans un champ alors qu'elle était visible au départ et est maintenant cachée, cochez cette case. Cela conserve la valeur de la variable même si un événement la cache. Généralement, cette option n'est pas cochée afin de réinitialiser « à nulle » la variable lorsqu'elle est masquée.

Lorsque vous avez beaucoup de variables, il est parfois plus facile de créer vos pages dans un tableur. Vous pouvez voir trente ou quarante variables à la fois et vous pouvez ajuster les propriétés très rapidement, puis les téléverser dans Epicraft. La page suivante montre comment votre fichier CSV doit être structuré et les termes clés à utiliser pour créer le formulaire ci-dessous.

Mon appartement

Formulaire d'appréciation du logement.

Description de mon appartement * 1
Deux chambres, immeuble des années 50, 2ème étage, centre ville...

Date d'entrée dans l'appartement * 2
01/01/2020

Mètres carrés * 3
100

Mon attitude le jour n° 1 4
 Euphorique !
 Joyeux(se)
 Content(e)
 Neutre
 Pas très satisfait(e)
 Cet appartement est un succès !

Action à entreprendre 5
 Je veux déménager

Préférez-vous contacter 6
 SOS Bad Apartments
 SOS Vacation time
 SOS Zen now!

Les composants (et le sous-type le cas échéant)

1 Champ texte

2 Date/heure

3 Nombre

4 Check

5 Boolean

6 Check

Certaines des propriétés des différentes variables

1,2,3 Obligatoire?

1 Placeholder
Deux chambres, immeuble des années 50, 2ème

1 Aide à la saisie
Donnez une description qui comprend le type

4 Nombre de colonnes : 3

2 Valeur par défaut
Maintenant : 01/01/2020

6 Valeur par défaut
SOS Bad Apartments

Explication

La variable et ses propriétés

Sur la page suivante, les colonnes A à F de la feuille de calcul indiquent la page et les variables, et les colonnes G à S ses propriétés.

Colonne A : type : Voici les cinq types d'éléments

Page : établit une nouvelle page avec un nouveau varset

Variable : la plupart de votre formulaire sera des variables

Texte : vous pouvez aussi inclure du texte statique.

Bloc : démarrer un groupe

FB : fin de bloc (groupes imbriqués sont acceptables)

Colonne B : varset : quand vous créez une nouvelle Page, donnez ici le nom du varset.

Colonne C : label : Le texte que l'utilisateur voit sur l'écran.

Colonne D : field_type : De quel type de variable s'agit-il ?

1. **Text :** Un champ qui accepte tout texte que l'on saisit.
- Text_multiline :** Un champ qui accepte beaucoup de texte
2. **date :** pour saisir la date par clavier ou via un calendrier
3. **integer :** Ce champ n'accepte que les nombres entiers.
decimal : Ce champ accepte les nombres décimaux.
4. **radio/list :** Sélectionnez une option parmi plusieurs.
5. **bool :** un bouton binaire oui/non (vrai/faux)
6. **checkbox :** cocher plusieurs réponses du dictionnaire

Colonne E : field_name : C'est le nom de la variable.

Colonne F : dico : Si votre variable est un radio, list ou checkbox, vous devez donner le nom d'un dictionnaire. Le dictionnaire doit déjà exister ou être téléversé à Epicraft en même temps dans un autre fichier CSV.

Ci-dessous vous voyez le fichier d'importation CSV utilisé afin de créer le formulaire sur la page précédente. Notez qu'on ne peut pas importer le placement des variables dans les colonnes. Des colonnes ont été ajoutées au formulaire après l'importation pour ajuster la présentation. Pour commencer, choisissez l'option **Importer** dans le menu.

* Obligatoire

** Nécessaire si le type est **V** (variable)

*** Nécessaire si **Field_type** est **radio**, **checkbox** ou **list**.

**** Lorsque le type **P** est donné, il crée une nouvelle page et un varset pour toutes les variables qui le suivent.

Utilisez la colonne varset si vous voulez donner un nom spécifique au varset de la page. L'utilisation d'un nom de varset existant créera une nouvelle page de variables.

Noms d'entêtes valables: type, varset, label, field_type, field_name, dico, tooltip, show_on, placeholder, len, label_position, mandatory, read_only, min, max, default_value, precision, column_count, css_class

Exemple à télécharger : www2.voozanoo.net/sites/default/files/documents/exemple_page_upload_point-vergule.csv

Créez un fichier CSV avec un tableur. Quand vous sauvegardez le fichier, choisissez obligatoirement le jeu de caractères UTF-8 Unicode (sinon, les caractères accentués peuvent ne pas s'afficher correctement). Un guide pour transformer un fichier au format UTF-8 est disponible dans la documentation du site web www.voozanoo.net.

Règles pour le fichier Varset au format CSV

- ⚠** Ne jamais utiliser le caractère # dans ce fichier !
- Les noms d'en-tête CSV doivent être exacts (type, label, etc.).
- Les colonnes peuvent être dans n'importe quel ordre.
- Tous les en-têtes ne sont pas obligatoires, voyez-les * à gauche.
- Ne pas inclure la ligne 2, ni les astérisques de cet exemple.
- Les variables dicos sans dictionnaires correspondants produiront une erreur.
- Vous pouvez téléverser les fichiers CSV du dictionnaire en même temps que les pages des variables.

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	type *	varset	label *	field_type **	field_name **	dico ***	tooltip	show_on	placeholder	len	label_position	mandatory	read_only	min	max	default_value	precision	column_count	css_class
2	P (page) T (text) V (variable) B (block start) FB (finish block)	Only used when the type column is a page "P". If left blank a random varset name will be created.	The text the user will see.	Valid options: text text_multiline integer decimal date time radio checkbox list bool	string /^[a-z][a-z_0-9]*	The name of the dictionary to use.	Help text the user will see if he/she places the mouse over the icon of the question.	The javascript expression when evaluated as true will display the question.	Text which will be shown in grey inside the text field to give an example of the type of input that is expected.	The maximum number of characters allowed to be typed into a text field.	Where the label of the variable is with respect to the entry field. inline (default) or above	If = 1, then this is a required field. If = 0 or is left blank, not a required field.	If set to 1, the user cannot modify the value of this variable. The default value is 0 (writable).	Examples Integer: 12 decimal: 39.5 date: 2015-12-25 datetime: 1999-31-12 23:59 time: 5:30	same format as min	caution. text: a string text_multiline: a string integer: an integer decimal: a decimal number Date: 2015-7-14 datetime: 2015-7-14 10:00 dictionary: code of label bool: 1 or 0	For decimal variables, the number of digits after the decimal point.	For dictionary variable with radio buttons or checkbox response, indicate the number of columns the responses should have.	The name of the CSS class the label should use.
3	P	appartement	Mon appartement																
4	T		Formulaire d'appréciation du logement.																
5	V		Description de mon appartement	text	appart_desc		Donnez une description q		Deux chambres	100	above		1						
6	V		Date d'entrée dans l'appartement	date	ent_appart_date						above		1			2000-01-01			
7	V		Mètres carrés	integer	surface						above		1		200		1		
8	V		Mon attitude le jour n° 1	radio	attitude	feeling					above		0						3
9	B		Action à entamer																
10	V		Je veux déménager	bool	demenager						inline								
11	V		Préférez-vous contacter	checkbox	contact			(feeling)==1			above								show-red
12	FB																		
13	P	Maison	Formulaire d'appréciation de votre maison.																
14	T		Formulaire d'appréciation du logement.																

Il est parfois plus facile d'importer plusieurs dictionnaires (dicos) en même temps au lieu de les saisir à la main dans l'éditeur. Dans le menu, choisissez l'option *Importer depuis un fichier*.

1

	A	B	C
1	dico_name	label	code
2	yorn	yes	1
3	yorn	no	0
4	yorn	Unknown	99
5	yorn	Refused to respond	88
6	langs	English	en
7	langs	Français	fr
8	langs	Deutsch	de
9	langs	Español	es
10	langs	Italiano	it
11	langs	Other	ot
12	gender	male	1
13	gender	female	0
14	gender	unknown	9



2

Dictionary Description 4 Dictionary Usage 1

Name of this dico

yorn ✓

Active Responses (3)

1	yes
0	no
99	Unknown

Archived Responses (1)

88	Refused to respond
----	--------------------

3

	A	B	C
1	dico_name	label	code
2	yorn	Yes	1
3	yorn	no	0
4	yorn	Refused	88
5	yorn	Non binary	3
6			

Prévisualiser

Regarder la liste des actions

4

avant modifications		après modifications		Ligne CSV
1	yes	1	Yes	2
0	no	0	no	3
88	Refused to respond	88	Refused	4
99	Unknown	3	Non binary	5
		99	Unknown	

Je comprends et accepte les changements dans ce dictionnaire.

Explication

Importation de nouveaux dictionnaires

Pour les nouveaux dictionnaires, créez un fichier CSV 1 avec vos dicos comme indiqué, puis téléchargez-le via l'option de menu **Importer depuis un fichier**. Assurez-vous que le fichier CSV est au format Unicode UTF-8. S'il ne l'est pas, les caractères accentués et non-latins ne seront pas transférés correctement.

Importation de mises à jour de dictionnaires

La version 1.19 a apporté des changements importants à la façon dont les dictionnaires sont mis à jour par le téléchargement d'un fichier CSV. Il est fortement recommandé de télécharger d'abord le ou les dictionnaires que vous souhaitez modifier, puis de modifier ce fichier CSV pour téléverser les modifications. Sinon, vous risquez d'archiver involontairement des éléments en raison des différences entre ce que vous pensiez que le dico contenait et ce qu'il contient réellement.

Le code d'un élément ne peut pas être modifié par des téléchargements. Il s'agit de l'identifiant unique. Si vous essayez de faire cela, vous allez simplement créer un nouvel élément avec le même label et archiver l'élément original.

Pour illustrer les différents changements qui peuvent avoir lieu lors du téléchargement d'un dictionnaire qui existe déjà dans votre projet, nous allons modifier le dico **yorn** existant 2 avec le fichier CSV 3. Comme pour l'importation de nouveaux dicos, vous pouvez modifier plusieurs dicos à la fois si vous le souhaitez. Notez bien la colonne de droite, **après modification**.

- Le jaune indique que le libellé a été modifié.
- Le vert indique qu'un nouvel élément a été créé.
- Le gris indique qu'un élément dico a été archivé. En excluant un élément dans le CSV, Epicraft l'archivera.
- Le blanc indique qu'il n'y a pas eu de modification.

*À un certain moment après l'importation du dictionnaire, quelqu'un a déplacé l'élément avec le code 88 dans la colonne " Archivé ".

Pour créer un nouveau projet identique ou basé sur un autre projet, il faut exporter l'ensemble ou une partie des pages et des dictionnaires. Vous pouvez ensuite importer ces fichiers CSV dans un nouveau projet.

The screenshot shows the export process in Voozano. At the top, a menu is open with 'Exporter les pages du projet' highlighted (1). Below, the 'Export pages' dialog is shown. It has a 'Séparateur' dropdown set to ':' (2). Under 'Pages', there are checkboxes for 'Tout sélectionner' (5) and a 'Filtrer par nom...' dropdown (4). A list of pages is shown, with 'Other', 'Person', and 'Visit' checked (3). At the bottom, a green button says 'Exporter 3 pages' (6). Under 'Dictionnaires', there is also a 'Filtrer par nom...' dropdown (4) and a list of dictionaries. 'langs (EMMYKd, EMMYKd, EMMYKd...)' and 'prof_type (pharmacist, oncologist, general practitioner...)' are checked (7). A green button at the bottom says 'Exporter 2 dictionnaires'.

Export

Vous pouvez exporter votre choix de pages ou de dictionnaires. Commencez par sélectionner ① Exporter dans le menu principal.

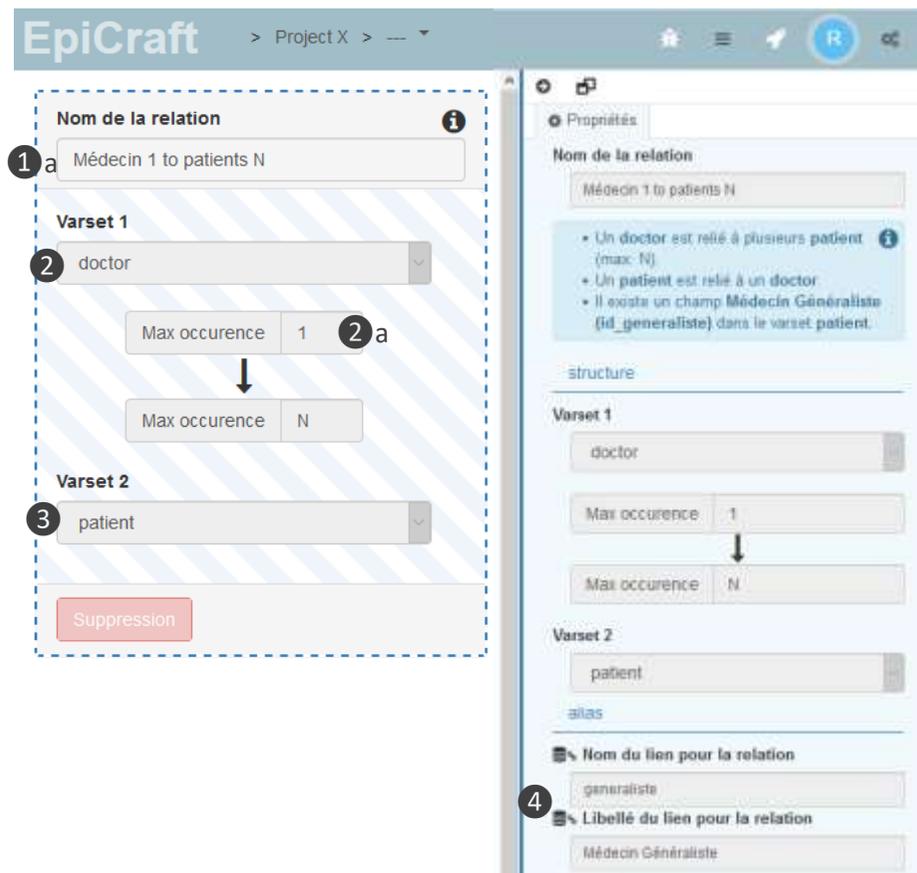
Boîte de dialogue de l'export

- ② Choisissez le séparateur pour les champs CSV (généralement une virgule pour les États-Unis et le Royaume-Uni, et un point-virgule pour le reste de l'Europe). Les autres pays varient.
- ③ Les éléments bleus sont les pages qui seront exportées. Cliquez sur une page pour la sélectionner / désélectionner.
- ④ Utilisez le « filtrer par nom... » pour trouver des pages particulières à inclure ou exclure. Ou cliquez sur ⑤ pour sélectionner ou désélectionner toutes les pages.
- ⑥ Exportez les pages sélectionnées dans un fichier CSV au format de caractères international UTF-8. Si vous ouvrez le fichier avec Microsoft Excel®, il peut interpréter incorrectement les caractères accentués et étrangers. Il est recommandé d'utiliser LibreOffice™ si vous voulez voir le contenu du fichier exporté. Les dictionnaires ⑦ sont exportés séparément et de la même manière.

Ces fichiers CSV peuvent maintenant être importés dans un autre projet. **Attention:** Seules les variables sont exportées, pas les éléments tels que les boutons, les bulks, les conteneurs matrices, les onglets et les colonnes. Les attributs des variables non exportés incluent : **largeur du label, cacher le label, label spécifique, propriétés personnalisées, métadonnées, affichage conditionnel, garde de données sur champ caché, nombre de colonnes et préfix/suffixe.**

NB : les variables archivées et les dictionnaires archivés ne seront pas inclus dans l'exportation.

L'écran ci-dessous vous permet de décrire une relation entre deux tables et la limite (le cas échéant) du nombre d'enregistrements pouvant être liés à un autre enregistrement. Notez que cette fonction ne fait qu'établir la relation. Il appartient à l'utilisateur d'utiliser la fonction **Sources de données** de chaque page pour accéder aux données des tables associées. Ici, les termes "parent" et "enfant" font référence à une relation et non à une description du contenu de la table.



1. Créer une nouvelle relation

Cliquer sur le nom du projet (haut à gauche) pour obtenir des informations sur le projet. Ouvrir le menu **avancé** et sélectionner **Relation entre varsets**, puis cliquer **+** pour créer une nouvelle relation. Lui donner un nom/titre **1 a** descriptif.

2. Sélectionner le varset parent

Sélectionner le varset qui sera le varset "parent". C'est-à-dire qu'un enregistrement de ce varset peut avoir plusieurs enregistrements "enfants" qui lui "appartiennent". Le chiffre 1 **2 a** ne signifie pas qu'un seul dossier parent (médecin) peut être créé. La relation peut être lue comme suit : "Pour chaque occurrence d'un dossier parent (médecin), nous pouvons avoir n'importe quel nombre (N) de dossiers enfants (patients)".

Note : Si vous entrez un nombre supérieur à un pour le varset (table) parent, vous allez créer une relation bidirectionnelle et une table de jonction spéciale sera créée par Epicraft. Un exemple de relation N à N pourrait être celui d'un hôpital qui peut avoir plusieurs études de recherche et une étude de recherche peut avoir plusieurs hôpitaux.

3. Sélectionner le varset enfant

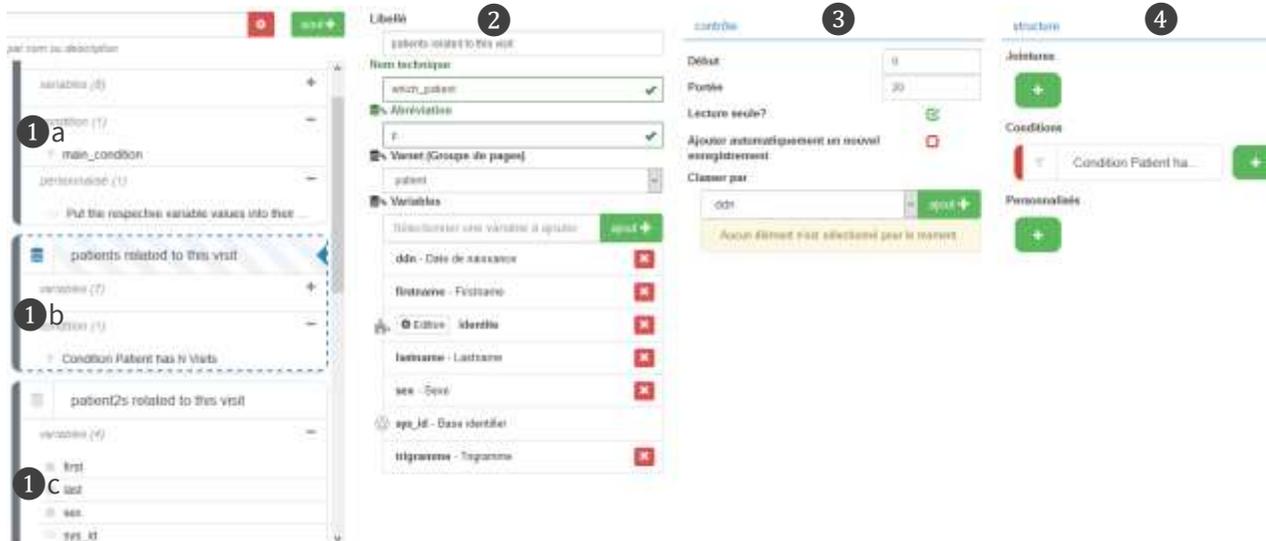
Saisir N, car nous ne savons pas combien de ces enregistrements seront créés. On peut définir une limite stricte au nombre d'enregistrements enfants appartenant à un enregistrement parent si on le souhaite.

4. Renommer le lien et l'étiquette

Si on le souhaite, on peut renommer le lien et libellé à autre chose que les noms proposés par Epicraft.

Explication

Lorsqu'une page est appelée, les variables d'un enregistrement donné doivent être affichées à l'écran ou un nouveau formulaire vide doit être affiché. Cela se fait par une requête de données lorsqu'une `id_data` spécifique (l'identifiant unique d'un enregistrement de données) est envoyée à la page lors de son ouverture. Depuis la version 1.18 d'Epicraft, sans écrire une seule ligne de XML, vous pouvez ajouter d'autres requêtes de données sur n'importe quelle table de la base de données quand on ouvre une page donnée. Cliquer sur l'icône à gauche de l'en-tête d'une page pour accéder à l'écran de Sources de données.



1. La liste des requêtes de données

Cette colonne affiche toutes les requêtes de données qui seront faites à la base de données lorsque la page sera ouverte. La requête **Source de données principale de la page** (1a) est générée automatiquement et existe pour obtenir les données de cette page. Dans cet exemple, deux autres requêtes (1b, 1c) ont été ajoutées par l'utilisateur. Noter que la requête **patients related to this visit** est actuellement sélectionnée (trait bleu pointillé). Les colonnes 2, 3 et 4 sont des paramètres spécifiques pour chacune des requêtes de données.

2. La table source et les variables

Cette colonne est l'endroit où l'on nomme la requête et sélectionne la table et les variables qui doivent être requêtés. On peut aussi ajouter des colonnes personnalisées, comme une nouvelle colonne calculée à partir d'autres variables de la page.

Cette section vous permet de déterminer comment l'ensemble de données (dataset) sera retourné : groupes de x enregistrements ; par ordre alphabétique et par champ ; le mode lecture/écriture ou lecture seule.

4. Structure

Cette section permet d'ajouter et d'afficher les jointures (joindre deux ensembles de données par une colonne commune), les conditions (à quelle condition la ligne de données sera incluse dans l'ensemble de données retourné) et les options personnalisées pour une requête donnée.

Une requête de données peut être de l'un des trois types suivants : les données de n'importe quelle table de la base de données, les données d'une table liée à la table de la page en cours, ou les requêtes sous forme libre utilisant exclusivement XML. Cette dernière n'est nécessaire que dans des cas très particuliers. Pour un exemple étape par étape de l'utilisation des requêtes de données dans les varsets liés, voir l'article d'EpiDocs "Comment exploiter une relation 1-N dans des formulaires avec le système de Sources de données ?" sur

https://epiconcept-paris.github.io/epidocs/exploit_relationship_datasources_ui.html

The screenshot shows the Epicraft interface for configuring data requests. On the left, a list of requests is shown, with the first one highlighted as the 'main source of the page'. The configuration panel on the right allows setting a label, technical name, abbreviation, varset, and variables. A legend at the bottom explains the three types of data sources: Varset (based on a varset without context reference), Relation (based on a relation), and Libre (based on handwritten XML).

Source de données principale

La requête principale ① est toujours présente. Par défaut, elle récupère un ensemble de données d'une ligne (à cause de la condition ②) de toutes les variables de cette page basée sur le paramètre **sys_id** lorsque la page est appelée. Vous pouvez ajouter des variables clés étrangères en cliquant dans le champ devant **+ Add**. Vous pouvez également ajouter des éléments (**jointures, conditions, personnalisés**) à cette requête par défaut ③ mais vous ne pouvez pas supprimer la requête.

D'autres requêtes ④

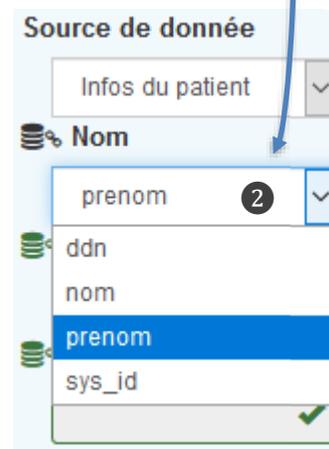
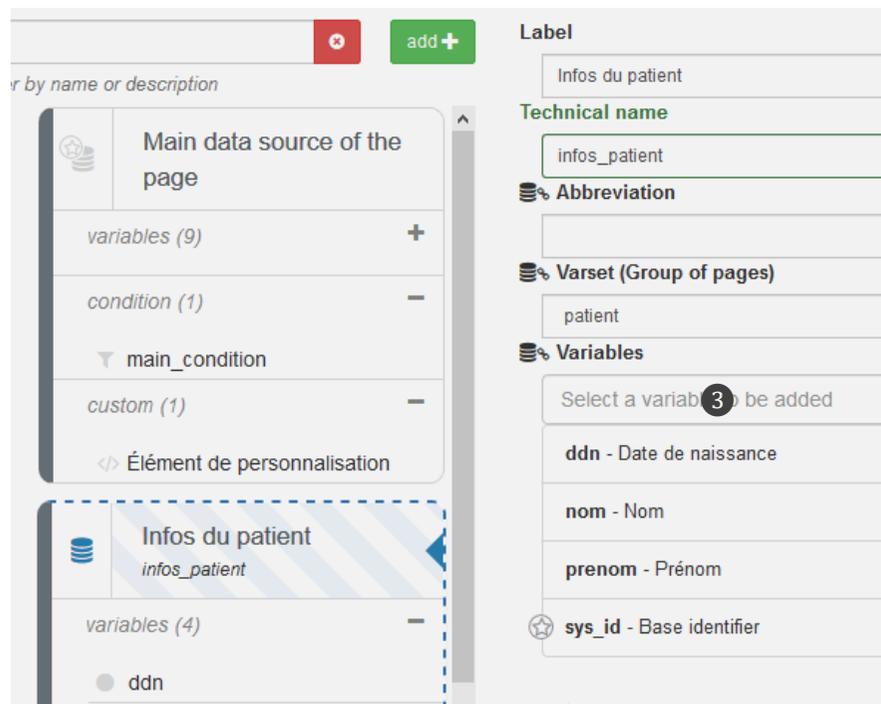
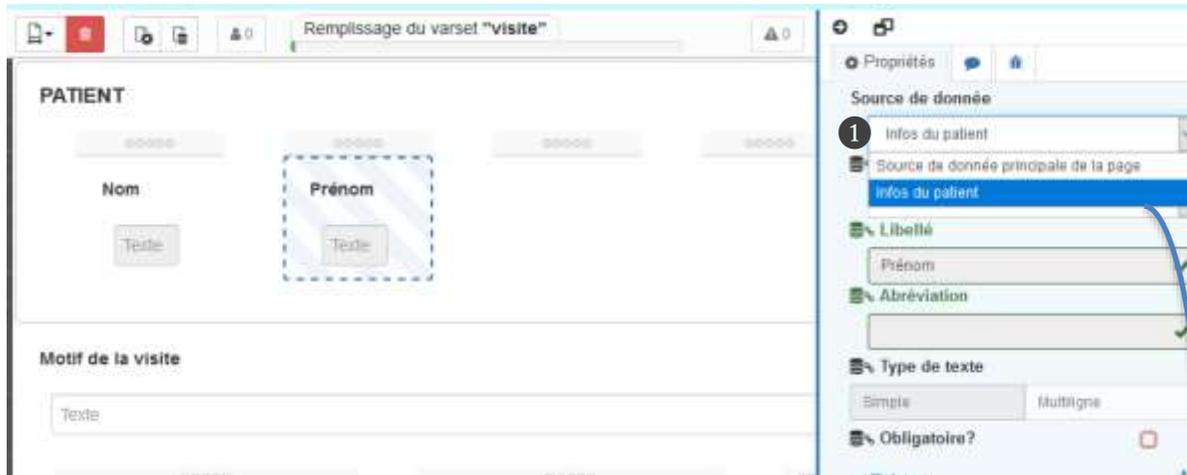
Vous pouvez ajouter autant de requêtes ⑤ à une page que nécessaire.

Varset : Une requête **Varset** retournera simplement un ensemble de données (dataset) des 20 premières lignes (ou la valeur que vous avez définie dans la portée de la page) de la table sélectionnée avec les variables sélectionnées ⑥.

Relation : Une requête à une table liée nécessite que, lors de la création de cette requête, vous spécifiez la table liée (voir la section ci-dessus **Création de liens entre les tables**). Dans les relations 1-N, Epicraft crée automatiquement une condition pour récupérer uniquement l'enregistrement dans le varset correspondant auquel l'enregistrement courant est lié (via la clé étrangère). Dans les relations N-N, Epicraft crée automatiquement une condition et une jointure à une table de jointure pour récupérer uniquement les enregistrements auxquels l'enregistrement courant est lié.

Personnalisé : Des modèles XML sont disponibles pour les utilisateurs avancés afin d'ajouter divers nœuds XML au nœud **data_structure** d'une page.

Si une requête de données est ajoutée à une page, qu'elle soit de type Varset, Relation ou Libre, cela signifie qu'un ensemble de données différent de la requête de données principale de la page sera renvoyé. Vous remarquerez que lorsqu'une ou plusieurs requêtes de données sont ajoutées à une page, les variables de l'une de ces requêtes peuvent être choisies.



Explication

Dans cet exemple, nous voulons afficher le nom et le prénom du patient lorsque nous remplissons un formulaire de visite. Pour afficher la valeur d'une variable d'une autre requête de données, commencez par placer une nouvelle variable de texte sur la page. Bien que nous soyons sur la page de visite, nous sélectionnerons la requête de données **Infos du patient** qui a été ajoutée manuellement.

Nouveau champ

Chaque fois qu'il y a une requête de données autre que la requête principale, on verra apparaître le nouveau champ Source de données ① avec un menu déroulant pour choisir la requête que l'on souhaite utiliser pour obtenir une variable spécifique. Les sources de données "Libre" ne sont pas sélectionnables.

Choix de la variable

Une fois la source de données choisie, vous avez le choix de la variable ② à afficher (parmi celles qui ont été ajoutées ③ lors de la création de la requête). Cette variable sera en lecture seule.

Dans cet exemple, la requête **infos_patient** était une requête Relation, ce qui signifie qu'il y avait une condition qui limitait le résultat de la requête à une seule ligne de la base de données. Comme il est logique que lorsque nous disposons d'un formulaire de visite avec une relation Patient à Visite de 1-N, chaque visite sera liée à un seul patient spécifique.

Si la requête n'était pas relationnelle, elle renverrait la liste complète des patients et n'afficherait que le premier résultat de la requête qui pourrait être long de plusieurs centaines de lignes. Ainsi, l'affichage des données d'une autre requête sur un formulaire n'a de sens que si la requête est de la type Relation ou Libre (avec les conditions adéquates) qui renvoie une seule et unique valeur.

Une variable calculée est celle qui n'obtient pas sa valeur à partir de la saisie de l'utilisateur, mais à partir d'un calcul d'autres variables (dans le même varset ou dans un varset relié). La variable calculée est une variable en lecture seule qui a un type (entier, décimal, texte, date, etc.) et est calculée avec le langage MySQL.



À la date de ces consignes, l'éditeur Epicraft n'a pas d'éditeur simple de "variable calculée" comme le fait l'application Voozanoo 4. Ainsi, une variable calculée doit être implémentée via une ressource et des éléments personnalisés dans Epicraft.

Les Étapes pour créer une variable calculée

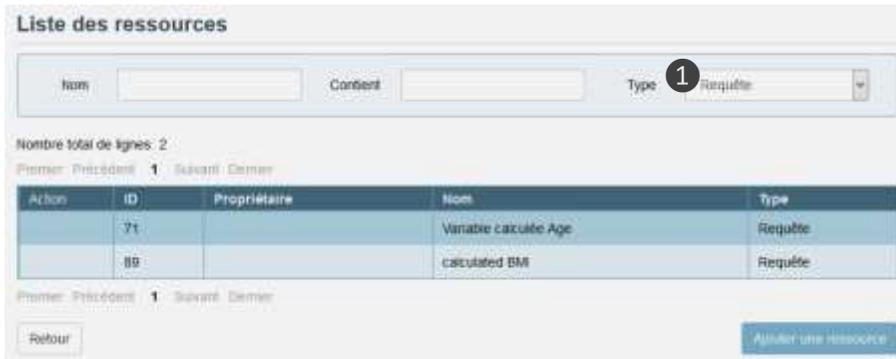
- 1) Ecrire l'expression en SQL
- 2) Ecrire la ressource XML de requête de ce dernier
- 3) Ajouter la nouvelle variable à la page

Using Voozanoo 4 to create your XML

Une variable calculée est en fait une requête à la base de données avec l'expression de calcul. La manière la plus simple de créer la requête est d'utiliser la page de variable calculée 1 dans l'application Voozanoo. La liste des variables calculées 2 et un bouton 3 pour créer une nouvelle variable sont présentés. En créant une variable temporaire 4, vous pouvez utiliser la ressource de la requête XML que Voozanoo crée comme base pour votre fichier de ressource dans Epicraft.

Notez 5 que lorsque vous créez une variable calculée, vous ne choisissez pas le varset auquel elle appartiendra. Voozanoo le fera lui-même parce qu'il aura besoin de placer la variable calculée soit dans le varset où les variables de la formule sont trouvées, soit dans le varset de niveau enfant le plus bas si les variables proviennent de varsets liés 1-N. Le message **varset inconnu** s'affiche si vous avez un nom de varset ou nom de variable inconnu dans le SQL. Veillez à utiliser uniquement les variables des varsets reliés.

Afin d'afficher la variable calculée sur une page, vous allez créer une ressource XML (la variable calculée) dans Epicraft.



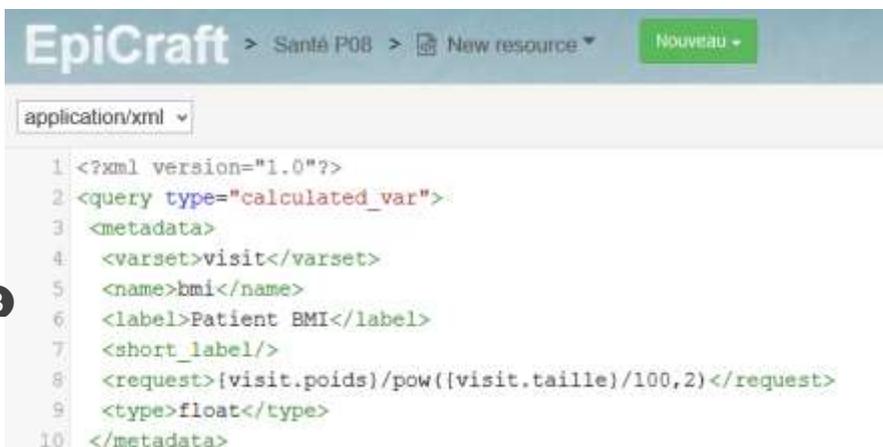
À ce stade, si vous n'avez pas besoin d'afficher la variable calculée sur une page, le travail est terminé. Votre variable calculée peut être utilisée dans les Listings, les Filtres et les Exports. Toutefois, si vous devez l'afficher sur une page ou déployer le projet sur plusieurs applications, vous devrez créer une nouvelle ressource dans Epicraft ainsi que certains éléments personnalisés.

Copier le XML de la requête

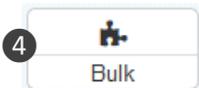
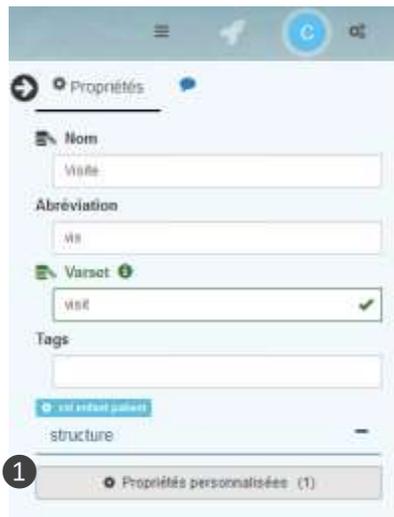
Allez à l'éditeur de ressources et **1** retrouvez la requête représentant votre variable calculée. Ouvrez-la et copiez l'XML. Retournez à la liste des requêtes et supprimer cette variable calculée.

Créer une nouvelle ressource dans Epicraft

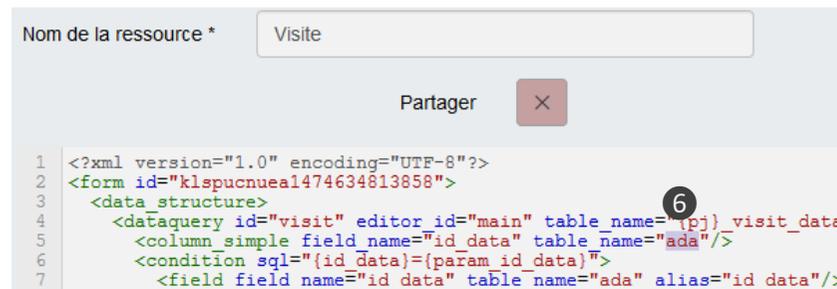
Dans Epicraft, créez une nouvelle ressource **2** pour votre application et donnez-lui un nom. Ensuite, collez le XML **3** dans cette nouvelle ressource. L'éditeur Epicraft va créer une nouvelle requête (variable calculée pour remplacer celle que vous avez supprimé) quand vous déployez ces modifications. Enregistrez votre nouveau fichier ressource.



Finalement, il faut ajouter la variable à la page, puis déployer les modifications de l'éditeur.



Vous trouverez le trigramme de **table_name** dans le formulaire de la page dans l'application Voozanoo sur laquelle vous allez placer la variable. Dans cet exemple



Ajouter une colonne à la requête du formulaire

Localisez la page sur laquelle vous allez afficher la nouvelle variable calculée et **1** créez une nouvelle propriété personnalisée. Si le calcul est basé sur des variables d'un seul varset, vous devez placer la variable calculée sur une page qui utilise ce varset. Si la variable est calculée à partir de deux ou plusieurs varsets, vous verrez la balise `<varset>` (4ème ligne du fichier ressource vue sur la page précédente) et vous devrez placer la variable sur une page utilisant ce varset. Une fois les propriétés personnalisées ouvertes, **2** ajoutez la propriété

`form.dataquery.main.column_simple`.

Dans celui-ci, ajoutez une simple ligne de XML **3** :

```
<column_simple field_name="nom_var" table_name="main" />
```

NB: Les projets créés en version inférieure à 2.26 doivent utiliser le trigramme du dataquery **6** au lieu de `main`.

Ajouter un « Bulk » pour la mise en page

Ajoutez l'élément **Bulk** **4** où vous le souhaitez à la page, puis ouvrez ses propriétés personnalisées et ajoutez la propriété `form.widget`. À l'intérieur, ajoutez **5** le XML suivant.

```
<form_row>
```

```
  <label dataset="nom_varset" field="nom_var" class="col-xs-3 text-right"/>
```

```
  <value dataset=" nom_varset " field="nom_var" class="col-xs-9"/>
```

```
</form_row>
```

Déployez vos modifications dans Epicraft. Le composant Bulk affichera la variable calculée pour l'utilisateur.

Parfois, il ne suffit pas de pouvoir placer les champs de saisie sur vos formulaires. Vous voudrez afficher une liste des informations provenant de la base de données sur une page. Comme il n'y a pas de composant "liste de données" dans Epicraft, il faut ajouter un **Bulk type form.widget** et utiliser la balise **Placeholder**. Dans l'exemple ci-dessous, on voit la partie supérieure du formulaire qui affiche un listing des données de la base de données.



2



3



4



Explication

1. Créer un Listing

Allez dans votre projet (application) et créer un nouveau listing. Pour notre exemple, nous l'appellerons « Adhérents ».

2. Ajouter un Bulk au formulaire

Au retour à Epicraft, ajouter le composant Bulk où vous voulez sur votre formulaire. Nous le faisons en haut.

3. Ajouter une propriété personnalisée

Ajouter une propriété personnalisée **form.widget** à votre composant Bulk.

4. Ajouter la balise <placeholder>

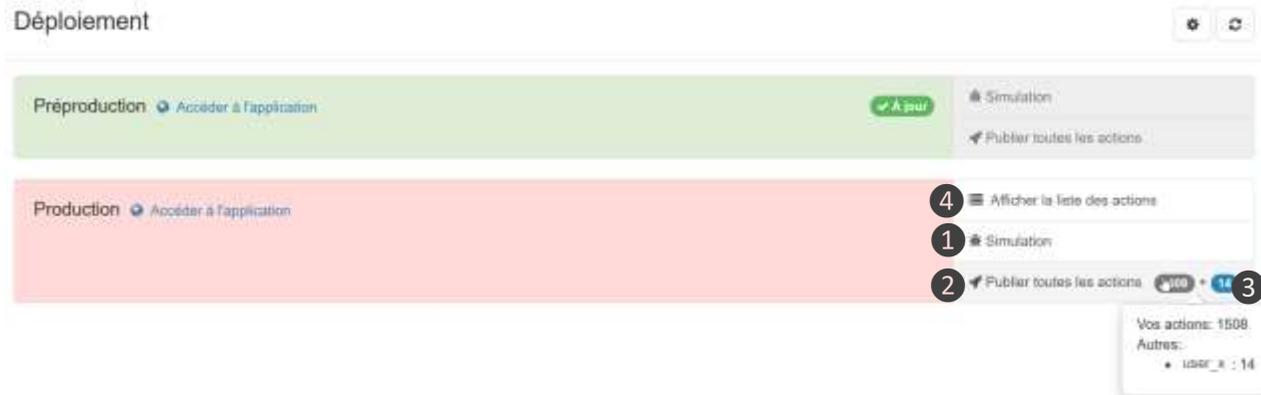
Entrez ou collez le XML suivant.

```

<placeholder type="listing">
  <option output="html" option_name="sid" value="Adhérents" />
</placeholder>
    
```

Votre page affichera maintenant la liste « Adhérents » où le Bulk a été placé. Si vous ou quelqu'un d'autre modifie le listing dans votre projet, ces modifications seront reflétées lorsqu'un utilisateur affiche cette page.

Après avoir ajouté et modifié vos pages, vous voulez généralement voir qu'elles fonctionnent comme prévu avant de les publier sur votre système de travail. La publication (également appelée déploiement) de votre projet envoie toutes les actions que vous avez effectuées depuis votre dernier déploiement à votre application Voozanoo où cette dernière lit ensuite les actions et crée et modifie les fichiers de ressources de l'application en conséquence.



- 1. Simulation** : Epicraft traitera toutes les actions pour vérifier qu'il n'y a pas d'incompatibilité et rapportera les résultats. Il n'appliquera pas réellement les actions à votre application.
- 2. Publier** : Envoyer les dernières actions qui seront lues et appliquées à l'application.
- 3. Actions** : Le nombre d'actions que vous ou d'autres personnes ayant modifié le projet, ont été effectuées depuis son dernier déploiement. Informe l'utilisateur si d'autres actions que les siennes vont être publiées.
- 4. Liste des actions** (seulement pour le profil Avancé) : Cliquez dessus pour voir les détails de chaque action.
- 5. Environnements** : Modifiez ou ajoutez des environnements. Cette icône est visible seulement aux personnes ayant la responsabilité de mettre en place et de connecter ce projet.
- 6. Actualisez** cet écran.
- 7. Afficher** le contenu de l'action.



- 8. Publier** jusqu'à cette action.
- 9. Supprimer** l'action. Une fois la suppression confirmée, une simulation sera effectuée pour indiquer quelles d'autres actions liées seront également supprimées. Décochez la case Simulation et cliquez à nouveau sur Supprimer pour réellement supprimer les actions. Un indicateur peut être présent pour prévenir l'utilisateur s'il ne peut pas supprimer une action.
- 10. Retrouver** une action spécifique par son nombre.
- 11. Sauter** directement au groupe de cinq actions.
- 12. Télécharger** un fichier texte de la liste complète des actions.