

## w.c.s. - Development #18406

### api: possibilité de rajouter des champs du formulaire dans l'export ics des demandes

01 septembre 2017 17:14 - Serghei Mihai

<b>Statut:</b>	Fermé	<b>Début:</b>	01 septembre 2017
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Serghei Mihai	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
En dehors du titre, la date, l'uuid et l'url backoffice des demandes il peut-être utile de spécifier les variables en plus. Par exemplan, prénom/nom du demandeur.			

#### Révisions associées

##### Révision c4d83209 - 31 octobre 2017 09:55 - Serghei Mihai

ics: add formdata description with backoffice url (#18406)

#### Historique

##### #1 - 01 septembre 2017 17:24 - Frédéric Péters

Ma proposition serait d'avoir le rendu de la description fait via `template_names = ['wcs/ics/description/%s.txt % formdef.url_name, 'wcs/ics/description.txt']`; ainsi on garde du temps pour réfléchir à un contenu par défaut et il reste possible d'adapter le format pour une commune où ça poserait problème.

(note à Thomas : me dire si je m'égare à profiter ainsi de la bascule à django).

##### #2 - 01 septembre 2017 17:38 - Thomas Noël

"il peut-être utile de spécifier" : je sais pas de quoi on parle, c'est juste une idée en l'air ou bien y'a un cas concret ?

##### #3 - 01 septembre 2017 17:43 - Serghei Mihai

Un cas concret c'est Villejuif qui aimerait avoir dans la description le nom du demandeur, l'url de traitement de la demande (le champ "url" actuel semble être ignoré par M\$ outlook).

##### #4 - 01 septembre 2017 17:44 - Frédéric Péters

M\$ outlook

Ah, l'adolescence...

##### #5 - 01 septembre 2017 17:45 - Frédéric Péters

Un cas concret c'est Villejuif qui aimerait avoir dans la description le nom du demandeur, l'url de traitement de la demande

Avoir un ticket côté Villejuif, le lier à celui-ci.

##### #7 - 01 septembre 2017 19:47 - Thomas Noël

J'ai du mal à suivre, un coup ça parle de description, l'autre coup de variables... bon comme je ne connais pas le format ICS, je vous laisser jouer ;)

##### #8 - 01 septembre 2017 20:43 - Frédéric Péters

J'ai du mal à suivre, un coup ça parle de description, l'autre coup de variables...

C'est normal vu la formulation dans les tickets... J'ai copié/collé le commentaire à l'origine du ticket dans #18408.

Cela étant posé, dans le format le champ "description" est du texte libre, qui va être repris comme étant le contenu de l'événement (le champ "description" des rendez-vous dans egroupware correspond). Pour le moment on n'inclut pas ce champ. Villejuif voudrait certaines informations dedans.

De là, on peut se poser la question "que mettre dedans ?" et une réponse pourrait être de reprendre l'ensemble des champs, un peu comme [form\_details] (pour lequel il faudrait une variante avec les données de traitement). Dans la pratique, je me dis que rapidement ça n'ira pas et c'est ça que j'anticipe en proposant de générer le contenu à partir d'un template django.

#### #9 - 01 septembre 2017 21:21 - Thomas Noël

Ok, pigé. Donc il s'agit d'avoir un `vevent.add('description').value = ...`

Ca rejoint une vieille discussion lors de l'implémentation du `display_id` : avoir un résumé de la demande. A noter qu'on avait eu un problème de perfs à ce moment : faire calculer toutes les variables sur un gros ensemble de formulaires, ça "tapait" trop (d'où le "minimal=True" dans `formdata::get_substitution_variables`). L'idée étant la même ici, j'ai peur qu'on ai le même soucis : il faudra vérifier "en vrai".

Et sinon je n'aime pas « `template_names = ['wcs/ics/description/%s.txt % formdef.url_name, 'wcs/ics/description.txt']` » parce que ça pose du paramétrage dans le thème... mais je n'ai pas d'autre proposition (et à Villejuif, client *on-premise*, ils sauront contrôler l'affaire, sans doute)

#### #10 - 03 octobre 2017 06:15 - Pierre Cros

Mail du 08/06 "Chrono à Villejuif" :

- Export ics : ça on l'avait promis et dedans il devrait y avoir : URL vers la demande en backoffice, type de rv, nom de la personne, créneau horaire

Par nom je voulais en réalité dire Nom et prénom évidemment.

Par créneau horaire comprends la date bien sûr.

#### #11 - 04 octobre 2017 15:00 - Serghei Mihai

- Fichier `0001-ics-allow-description-rendering-in-custom-templates-.patch` ajouté

- Statut changé de Nouveau à En cours

- Patch proposé changé de Non à Oui

Version avec un template de description vide par défaut.

Le formulaire est exposé sous forme de dictionnaire ce que veut dire que l'accès à ses variables se fera de la manière `form.var_prenom`, `form.var_nom`, etc.

Le test est à améliorer pour prendre en compte un `description.txt` par formulaire. Je suis en train de le faire.

#### #12 - 04 octobre 2017 15:02 - Serghei Mihai

- Fichier `0001-ics-allow-description-rendering-in-custom-templates-.patch` supprimé

#### #13 - 04 octobre 2017 15:02 - Serghei Mihai

- Fichier `0001-ics-allow-description-rendering-in-custom-templates-.patch` ajouté

#### #14 - 04 octobre 2017 16:42 - Serghei Mihai

- Fichier `0001-ics-allow-description-rendering-in-custom-templates-.patch` ajouté

Sinon ça me paraît plus simple et clair d'utiliser `'wcs/ics/description-%s.txt % formdef.url_name` pour la description par formulaire, sans créer un répertoire dédié.

Test à jour pour créer un fichier de description temporaire par formulaire.

#### #15 - 04 octobre 2017 17:21 - Frédéric Péters

Le formulaire est exposé sous forme de dictionnaire ce que veut dire que l'accès à ses variables se fera de la manière `form.var_prenom`, `form.var_nom`, etc.

Plutôt hostile à l'idée de venir avec une nouvelle façon d'accéder aux données. Que le template reçoive `get_substitution_variables`, comme tout le monde.

#### #16 - 04 octobre 2017 19:29 - Thomas Noël

Je rappelle de mon côté « à noter qu'on avait eu un problème de perfs à ce moment ».

Je pense qu'on pourrait répondre à ce qui est demandé avec un `formdata.display_label`, sur le modèle du `display_id`, qui serait utilisé à la place du

nom de la demande (form\_name + id) quand il existe. Et une action de workflow pour le modifier, et voilà. (vieille idée, en fait, pour avoir des demandes nommées "Inscription de Truc Machin" au lieu de "Inscription de votre enfant - 45")

#### #17 - 04 octobre 2017 19:42 - Frédéric Péters

Je rappelle de mon côté « à noter qu'on avait eu un problème de perms à ce moment ».

De manière générale, exécuter du code sur l'ensemble des formdata, c'est coûteux. (dans le code de w.c.s. même mais aussi au-delà, genre la recherche des templates django réalisée à chaque itération). (→ mesurer à quel point c'est coûteux, avoir une idée du nombre d'itérations à gérer correctement, etc.)

Je pense qu'on pourrait répondre à ce qui est demandé avec un "formdata.display\_label", sur le modèle du "display\_id", qui serait utilisé à la place du nom de la demande (form\_name + id) quand il existe. Et une action de workflow pour le modifier, et voilà. (vieille idée, en fait, pour avoir des demandes nommées "Inscription de Truc Machin" au lieu de "Inscription de votre enfant - 45")

Ce qui est demandé doit être mal décrit; il y a besoin là que la de description reprenne l'URL vers la demande, par exemple.

#### #18 - 09 octobre 2017 17:35 - Serghei Mihai

- Fichier 0001-ics-add-formdata-description-with-backoffice-url-184.patch ajouté

Ok. Patch minimal qui rajoute le l'url en backoffice dans le champ description.

Pour le rajout d'autres informations, l'idée d'avoir un champ du formulaire display\_label mis à jour à chaque store du formdata me semble bien. Mais c'est un sujet plus complexe qui devrait être traité séparément, à mon avis.

#### #19 - 09 octobre 2017 17:39 - Frédéric Péters

Ok. Patch minimal qui rajoute le l'url en backoffice dans le champ description.

Mouais.

Mais quand même retirer la mécanique de template.

#### #20 - 09 octobre 2017 18:05 - Serghei Mihai

- Fichier 0001-ics-add-formdata-description-with-backoffice-url-184.patch ajouté

Je ne suis pas convaincu non plus, mais ça répond en partie à la demande.  
Patch, épuré, à jour.

#### #21 - 09 octobre 2017 18:09 - Frédéric Péters

Je ne suis pas convaincu non plus, mais ça répond en partie à la demande.

Mais comme "en partie" ("aimerait avoir dans la description le nom du demandeur, l'url de traitement de la demande"); on veut vraiment ça, qui n'est ni bien ni adéquat ?

#### #22 - 13 octobre 2017 12:02 - Benjamin Dauvergne

- Assigné à mis à Serghei Mihai

#### #23 - 13 octobre 2017 23:18 - Thomas Noël

Discuté au bureau : en fait le bogue est dans Outlook ou Exchange, en tout cas dans leur non prise en compte du champ "url" envoyé dans le vevent par `vevent.add('url').value = formdata.get_url(backoffice=True)`. Je serais d'avis d'assigner le bogue à Microsoft sur leur redmine...

#### #24 - 14 octobre 2017 00:30 - Frédéric Péters

On est d'accord pour dire que malheureusement cette suggestion ne peut pas être suivie ?

#### #25 - 14 octobre 2017 10:04 - Benjamin Dauvergne

Thomas Noël a écrit :

Ok, pigé. Donc il s'agit d'avoir un `vevent.add('description').value = ...`

Ca rejoint une vieille discussion lors de l'implémentation du `display_id` : avoir un résumé de la demande. A noter qu'on avait eu un problème de perfs à ce moment : faire calculer toutes les variables sur un gros ensemble de formulaires, ça "tapait" trop (d'où le `minimal=True` dans `formdata:get_substitution_variables`). L'idée étant la même ici, j'ai peur qu'on ai le même soucis : il faudra vérifier "en vrai".

J'ai l'impression qu'on a la réponse ici, il existe ce code pour avoir un résumé/intitulé de la demande ou pas ? On y ajoute l'URL du formulaire et basta, [stacoverflow](#) indique que toute URL mise dans la description d'un VEVENT devient cliquable sous Outlook et dans l'agenda iOS (et Google gère l'attribut "url" et les autres s'ils sont pas trop cons ils font l'un ou l'autre),

#### #26 - 14 octobre 2017 10:14 - Benjamin Dauvergne

Je m'auto-répond, donc on a rien de ce genre on a juste `get_display_id()` qui renvoie une chaîne de formatage en dur dans la classe `FormDef`. Est-ce qu'on a vraiment besoin que ce soit configurable ici ? On peut pas décider d'un template en dur suffisant, "[nom du formulaire] — [display\_id] - [status\_label]\n[form\_user\_display\_name] — [form\_user\_email]\n[backoffice\_url]" ?

C'est pas non plus une fonctionnalité critique le support .ics via un template, si ?

De toute façon c'est soit ça soit on va directement à du `get_substitution_variable(minimal=True)` et un template EZT définissable au niveau du `FormDef` (et donc faut mesurer les perfs de ça, avec 50 000 formulaires genre). À noter que les performances ça s'améliore aussi en faisant du cache et en mettant à jour de manière asynchrone (via une tâche cron par exemple).

#### #27 - 14 octobre 2017 11:20 - Frédéric Péters

On y ajoute l'URL du formulaire et basta

C'est l'option prise dans le dernier patch de Serghei et je demandais alors si on se satisfaisait de ce patch qui ne remplissait qu'en partie la demande.

#### #28 - 14 octobre 2017 17:47 - Thomas Noël

Frédéric Péters a écrit :

On y ajoute l'URL du formulaire et basta

C'est l'option prise dans le dernier patch de Serghei et je demandais alors si on se satisfaisait de ce patch qui ne remplissait qu'en partie la demande.

Pour rappel, la demande c'est « on ajoute **par exemple** nom du demandeur, l'url de traitement de la demande. » (#18408). Le nom du demandeur étant un champ de la demande (formulaires anonyme), on est coincé, on peut effectivement pas répondre pour l'instant, question de perf. Peut-être un sujet à étudier, avoir un truc moins couteux que toutes les variables de contexte, par exemple juste les colonnes du SQL "telles quelles" et balancer dans un template Django, comparer ce que ça donne sur 50000 demandes.

#### #29 - 14 octobre 2017 22:09 - Benjamin Dauvergne

Sans passer par `get_substi..()` ce n'est pas plus coûteux de sortir l'URL que le nom du demandeur ou je rate un truc ? Entre `formdata.id` ou `formdata.data['f1']` je ne vois pas trop la différence, on a quand même fait un `SELECT * FROM formdata_...` pour y arriver. Le seul truc que je comprends ici c'est que le nom du demandeur sort de champs du formulaire (de traitement ou pas) et que donc ça impose un template spécifique et utiliser `get_substitution_variables()`.

Si `get_substitution_variables()` est lent on pourrait voir d'où ça vient et penser à une implémentation différente, par exemple plutôt que renvoyer un simple dico se baser sur le fonctionnement des `Context()` Django qui permettent d'avoir une pile de `Context` et donc de réutiliser par exemple les variables de substitution issues du `formdef` pour tous les formulaires issues du même `FormDef` durant une même requête (je ne dis pas que c'est la solution mais c'est une possibilité).

Je me dis aussi qu'on pourrait avoir un `FormDataContext()` qui soit purement virtuel et ne résoudre les références qu'à la demande, i.e. pas de construction par avance d'un dico.

#### #30 - 16 octobre 2017 09:44 - Frédéric Péters

Ça part peut-être trop long trop compliqué. Serghei ? Peut-être alors avec comme justification les performances, en attendant de les étudier et de mettre en place des choses (autres tickets), partir sur un modèle fixe contenant uniquement des champs "faciles". (et si c'est uniquement l'url, so be it).

#### #31 - 16 octobre 2017 09:54 - Serghei Mihai

Dans l'immédiat je pense qu'on devrait partir sur des choses simples, comme le suggère Benj dans le commentaire 26.

Au lieu de taper dans les attributs du formulaire pour retrouver le demandeur, je serais pour récupérer le nom de l'utilisateur attaché à la demande, s'il existe.

Je fais un patch pour ça.

### #32 - 16 octobre 2017 10:35 - Benjamin Dauvergne

Serghei Mihai a écrit :

Dans l'immédiat je pense qu'on devrait partir sur des choses simples, comme le suggère Benj dans le commentaire 26. Au lieu de taper dans les attributs du formulaire pour retrouver le demandeur, je serais pour récupérer le nom de l'usager attaché à la demande, s'il existe.

Je fais un patch pour ça.

Euh mais là on est sur un formulaire majoritairement demandé par des personnes non connectées non ? Ou bien j'ai mal compris une remarque précédente (faut garder en vu le cas d'usage présent tout de même).

Moi ça m'irait que pour un premier patch qu'on ait un truc battard qui cherche un champ varname="prenom" et un champ varname="nom" si aucun utilisateur n'est associé à la demande, c'est en dur c'est moche, mais ça fait le job pour une v0, si un jour migre vers un template explicite on corrigera ça.

### #33 - 16 octobre 2017 18:23 - Serghei Mihai

- Fichier 0001-ics-add-formdata-description-with-backoffice-url-184.patch ajouté

Je pense qu'on aura les 2 cas: usager connecté et non.

Je préfère partir sur l'exemple de l'usager connecté, dans quel cas l'ics contiendra son fullname. Sinon juste le titre, id, statut, lien de la demande.

J'ai un peu galéré avec vobject qui ne propose rien pour gérer un champ description multiligne. Donc pour respecter la norme et avoir des lignes séparées j'ai rajouté des espaces jusqu'à 74..

### #34 - 16 octobre 2017 18:27 - Frédéric Péters

J'ai un peu galéré avec vobject qui ne propose rien pour gérer un champ description multiligne. Donc pour respecter la norme et avoir des lignes séparées j'ai rajouté des espaces jusqu'à 74..

Uh?

Example: A multiple line value of:

```
Project XYZ Final Review
Conference Room - 3B
Come Prepared.
```

would be represented as:

```
Project XYZ Final Review\nConference Room - 3B\nCome Prepared.
```

### #35 - 16 octobre 2017 18:35 - Serghei Mihai

RFC2445, 4.1:

Lines of text SHOULD NOT be longer than 75 octets, excluding the line break. Long content lines SHOULD be split into a multiple line representations using a line "folding" technique. That is, a long line can be split between any two characters by inserting a CRLF immediately followed by a single linear white space character (i.e., SPACE, US-ASCII decimal 32 or HTAB, US-ASCII decimal 9).

### #36 - 16 octobre 2017 19:20 - Frédéric Péters

Ton passage dit que les lignes dans le fichier ne doivent pas dépasser 75 caractères.

Mon passage explique comment représenter des retours à la ligne.

Et au-delà des RFC, vobject fait le taf correctement : il ne produit pas de longue ligne et échappe les retours à la ligne :

```
>>> import vobject
>>> cal = vobject.iCalendar()
>>> vevent = vobject.newFromBehavior('vevent')
>>> vevent.add('description').value = 'Hello%s\nWorld' % ('X'*100)
>>> cal.add(vevent)
```



0001-ics-allow-description-rendering-in-custom-templates-.patch	4,5 ko	04 octobre 2017	Serghei Mihai
0001-ics-add-formdata-description-with-backoffice-url-184.patch	4,18 ko	09 octobre 2017	Serghei Mihai
0001-ics-add-formdata-description-with-backoffice-url-184.patch	1,89 ko	09 octobre 2017	Serghei Mihai
0001-ics-add-formdata-description-with-backoffice-url-184.patch	3,05 ko	16 octobre 2017	Serghei Mihai
0001-ics-add-formdata-description-with-backoffice-url-184.patch	2,78 ko	27 octobre 2017	Serghei Mihai
0001-ics-add-formdata-description-with-backoffice-url-184.patch	3,13 ko	30 octobre 2017	Serghei Mihai