

w.c.s. - Bug #46475

Bloc de champs et option de workflow : quelques soucis...

08 septembre 2020 17:02 - Benjamin Dauvergne

Statut:	Rejeté	Début:	08 septembre 2020
Priorité:	Bas	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Non		

Description

Ça

```
[2020-09-08 16:41:03] exception caught
Exception:
  type = '<class 'AttributeError'>', value = '<NoneType' object has no attribute 'data''

Stack trace (most recent call first):
  File "/home/bdauvergne/wd/eo/wcs/wcs/variables.py", line 857, in inspect_keys
    855     def inspect_keys(self):
    856         if self._field.max_items > 1:
>  857             data = self._formdata.data.get(self._field.id) ['data']
    858             return [str(x) for x in range(len(data))]
    859         else:

locals:
  self = <wcs.variables.LazyFieldVarBlock object at 0x7f8159118400>
```

et ça :

```
Stack trace (most recent call first):
  File "/home/bdauvergne/wd/eo/wcs/wcs/qommon/substitution.py", line 214, in __getitem__
    212             # TypeError will happen if indexing is used on a string
    213             if i == 1:
>  214                 raise KeyError(key)
    215             else:
    216                 parts = parts[i:]

locals:
  __class__ = <class 'wcs.qommon.substitution.CompatibilityNamesDict'>
  current_dict = <wcs.variables.LazyFieldVarBlock object at 0x7f0183ffc48e0>
  i = 1
  key = 'form_option_signatures_0'
  part = '0'
  parts = ['0']
  self = {'form': <wcs.variables.LazyFormData object at 0x7f0183ffc730>, 'attachments': <wcs.workflowflows.AttachmentsSubstitutionProxy object at 0x7f0183ffc7c0>, 'category_name': 'Cat1', 'category_id': 'cat1', 'category_slug': 'cat1', 'category_description': None}
```

Ça semble réglé avec le patch suivant :

```
diff --git a/wcs/variables.py b/wcs/variables.py
index abc43a21..ed9d606e 100644
--- a/wcs/variables.py
+++ b/wcs/variables.py
@@ -854,7 +854,7 @@ class LazyBlockDataVar(LazyFormDataVar):
  class LazyFieldVarBlock(LazyFieldVar):
      def inspect_keys(self):
          if self._field.max_items > 1:
-             data = self._formdata.data.get(self._field.id) ['data']
+             data = self._data.get(self._field.id) ['data']
+             return [str(x) for x in range(len(data))]
```

```

        else:
            return ['var']
@@ -868,7 +868,7 @@ class LazyFieldVarBlock(LazyFieldVar):
            int(key)
        except ValueError:
            return super().__getitem__(key)
-         data = self._formdata.data.get(self._field.id)['data'][int(key)]
+         data = self._data.get(self._field.id)['data'][int(key)]
        return LazyBlockDataVar(self._field.block.fields,
                                data,
                                formdata=self._formdata,
@@ -877,7 +877,7 @@ class LazyFieldVarBlock(LazyFieldVar):
    )

    def __len__(self):
-         data = self._formdata.data.get(self._field.id)['data']
+         data = self._data.get(self._field.id)['data']
        return len(data)

    @property
@@ -886,7 +886,7 @@ class LazyFieldVarBlock(LazyFieldVar):
        return self[0]

    def __iter__(self):
-         data = self._formdata.data.get(self._field.id)['data']
+         data = self._data.get(self._field.id)['data']
        for i in range(len(data)):
            yield self[i]

@@ -897,7 +897,7 @@ class LazyFieldVarBlock(LazyFieldVar):
        break
    else:
        raise AttributeError('No such attribute %r' % key)
-         return [data.get(field.id) for data in self._formdata.data.get(self._field.id)['data']]
+         return [data.get(field.id) for data in self._data.get(self._field.id)['data']]

class LazyUser(object):

```

Historique

#1 - 08 septembre 2020 17:11 - Frédéric Péters

- Statut changé de Nouveau à Rejeté

Un ticket par soucis, stp. Et si tu veux faire un patch, tu fais, mais pas poser du diff là comme ça sans rien exposer d'une situation et imaginer que quelqu'un prenne derrière le boulot d'intégrer ça correctement.