

octonet - Segnalazione #192

Eccezione non gestita per home esistente

02/16/2017 10:19 AM - Christopher R. Gabriel

Status:	Chiuso	Start date:	02/16/2017
Priority:	Normale	Due date:	
Assignee:	Mark Caglienzi	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
Se provo a creare un utente indicando una directory home esistente per qualcun'altro, ottengo una eccezione non gestita (vedi sotto).			
Nel form di creazione utente, spostiamo il campo username come primo campo, e autogeneriamo il valore del campo Home directory a "/home/{{username}}", cosi' e' piu' veloce da inserire, e soprattutto non rischiamo che l'autocomplete dei browser metta un valore precedentemente inserito, che magari esiste gia'.			
in ogni caso, l'eccezione va gestita e mostrare come errore di validazione del form.			
L'eccezione:			
Environment:			
Request Method: POST			
Request URL: http://localhost:8000/users/user/create			
Django Version: 1.8.4			
Python Version: 3.4.2			
Installed Applications:			
['django.contrib.sessions',			
'django.contrib.messages',			
'whitenoise.runserver_nostatic',			
'django.contrib.staticfiles',			
'octonet',			
'hostqueue.apps.HostQueue',			
'firewall.apps.Firewall',			
'dhcp.apps.Dhcp',			
'dansguardian.apps.DansGuardian',			
'host.apps.Host',			
'polygen.apps.Polygen',			
'upgrade.apps.Upgrade',			
'asterisk.apps.Asterisk',			
'samba.apps.Samba',			
'users.apps.Users',			
'script.apps.Script',			
'quota.apps.Quota',			
'printers.apps.Printers',			
'auth.apps.Auth']			
Installed Middleware:			
['django.middleware.security.SecurityMiddleware',			
'whitenoise.middleware.WhiteNoiseMiddleware',			
'django.contrib.sessions.middleware.SessionMiddleware',			
'django.middleware.common.CommonMiddleware',			
'django.middleware.csrf.CsrfViewMiddleware',			
'django.middleware.locale.LocaleMiddleware',			
'octonet.middleware.AuthenticationMiddleware',			
'django.contrib.messages.middleware.MessageMiddleware',			
'django.middleware.clickjacking.XFrameOptionsMiddleware']			
Traceback:			
File "/usr/local/lib/python3.4/dist-packages/django/core/handlers/base.py" in get_response			
132. response = wrapped_callback(request, *callback_args, **callback_kwargs)			

```
File "/usr/local/lib/python3.4/dist-packages/django/views/generic/base.py" in view
71.         return self.dispatch(request, *args, **kwargs)
File "/home/cgabriel/git/fuss/octonet/octonet/mixins.py" in dispatch
105.         return super().dispatch(request, *args, **kw)
File "/usr/local/lib/python3.4/dist-packages/django/views/generic/base.py" in dispatch
89.         return handler(request, *args, **kwargs)
File "/usr/local/lib/python3.4/dist-packages/django/views/generic/edit.py" in post
215.         return self.form_valid(form)
File "/home/cgabriel/git/fuss/octonet/users/views.py" in form_valid
516.         self.user = self.root_tree.lcreate(["users", "users", name], user_data)
File "/home/cgabriel/git/fuss/pyoctofuss/octofuss/xmlrpc.py" in lcreate
84.         return self._wrap_call(self.server.create, "/" + path, dumps(value))
File "/home/cgabriel/git/fuss/pyoctofuss/octofuss/xmlrpc.py" in _wrap_call
42.         raise globals().get(name, UnknownException)(msg)

Exception Type: UnknownException at /users/user/create
Exception Value: The choosen home directory already exists
```

Associated revisions

Revision e31221dd - 04/20/2017 04:27 PM - Mark Caglienzi

Try to workaround the problem, in some (not perfectly neat) way... I don't know if this is enough. refs: #192

Revision 778a3857 - 04/20/2017 04:37 PM - Mark Caglienzi

Added comment and better message. refs: #192

Revision 774b1b8c - 04/21/2017 02:50 PM - Mark Caglienzi

Catch the two exceptions instead of all of them. refs: #192

Revision 54af64db - 04/21/2017 04:53 PM - Mark Caglienzi

Inject error in the home field or give standard message. refs: #192

Revision 103b82ee - 04/27/2017 12:18 PM - Mark Caglienzi

Merge branch 'master' into t192. refs: #192

Revision abfb24af - 04/27/2017 02:59 PM - Mark Caglienzi

Merge branch 'master' into t192. refs: #192

Revision 77f8c55e - 04/27/2017 04:04 PM - Mark Caglienzi

Manage custom exceptions for the homeDirectory field. refs: #192

Revision 57a5ddf3 - 04/27/2017 04:33 PM - Mark Caglienzi

Catch UserAlreadyExistsError exception. refs: #192

History

#1 - 04/18/2017 10:48 AM - Mark Caglienzi

- Status changed from Nuovo to In elaborazione

#2 - 04/18/2017 05:01 PM - Mark Caglienzi

- Assignee changed from Mark Caglienzi to Christopher R. Gabriel

Le richieste collaterali (spostare il campo username e autocompletare il campo home) sono state fatte in [#230](#) / [#225](#)

Il problema della home esistente non è semplicemente risolvibile, perché è octofussd a sapere se nel suo filesystem c'è una directory oppure no (e il controllo viene fatto all'atto della creazione dell'utente, quando il form è già validato), mentre octonet non può saperlo (perché octofussd può risiedere su una macchina remota rispetto a octonet).

Il flusso che ho decifrato è questo:

- octofusssd/plugins/usersplugin/common.py UserList.new_element() viene chiamato all'atto della lcreate(), e questo chiama UserDB.new_user()
- octofusssd/plugins/usersplugin/ldapusers.py UserDB.new_user() fa l'aggiunta a LDAP e poi invoca octofuss.hooks.invoke("new_user")

Il controllo dell'esistenza della directory viene fatto in @octofusssd/plugins/usersplugin/common.py UserDB._fillInNewUser(), in cui c'è il raise dell'eccezione se os.path.exists().

Le possibili soluzioni che mi vengono in mente sono:

- Aggiungere a octofusssd qualcosa di chiamabile dal clean_home() di UserForm/CreateUserForm (il fatto che sia richiamabile da qui e non ad esempio dalla view è un punto **fondamentale**) che wrappi soltanto la chiamata a os.path.exists e che dia True/False in modo da poter dire fin da subito che il campo home non è valido se la directory (sulla macchina ove risiede octofusssd) esiste già.
- Wrappare tutto il codice della view CreateUser a partire dalla lcreate in un try/except in modo che non faccia nulla se viene sollevata l'eccezione (e a questo punto sarebbe il caso di far fare a octofusssd il raise di una eccezione custom ben specifica, qualcosa tipo AlreadyExistingHomeDirectory, figlia della giusta sottoclasse di Exception), vedendo se poi può avere senso (e se funziona correttamente a livello di UX) fare return super().form_invalid(form) da dentro a form_valid() (e già l'idea mi piace pochissimo perché è sporca)
- Hai idee alternative/migliori?
- Se ti piace la mia prima proposta (delle due che mi son venute in mente è quella che preferisco, **se** è fattibile), ti vengono in mente consigli implementativi e possibili effetti collaterali (anche rispetto ad altre parti di codice) che a me possono sfuggire?

#3 - 04/19/2017 09:33 AM - Christopher R. Gabriel

- Assignee changed from Christopher R. Gabriel to Mark Caglienzi

Io sono per la seconda (quella dell'eccezione AlreadyExisting etc), perché la prima non saprei come inserirla nel protocollo rest-based di octofusssd.

#4 - 04/19/2017 05:10 PM - Mark Caglienzi

- Assignee changed from Mark Caglienzi to Christopher R. Gabriel

Ho provato a fare una eccezione diversa da ValueError (lavorando direttamente sulla VM dato che non ho altro modo per testare octofusssd in versione non mock). Python 3 ha FileExistsError (che in python 2 non esiste).

Ma anche così l'eccezione non son riuscito a catcharla nel form.

Ho provato (sempre sulla VM direttamente) a wrappare tutto il codice della form_valid dalla chiamata a octofusssd in poi dentro a un try come dicevo prima, ma all'except non ci arriva mai, perché la _wrap_call di xmlrpc.py fa raise di UnknownError anche se la exception è diversa (ValueError, presente in py2 e py3, FileExistsError, solo py3, o custom che sia).

#5 - 04/20/2017 09:27 AM - Mark Caglienzi

- Assignee changed from Christopher R. Gabriel to Mark Caglienzi

- Assignee changed from Mark Caglienzi to Christopher R. Gabriel

Oggi sono riuscito a "superare" uno dei problemi.

- Non ho toccato octofusd per ora
- Ho modificato la view perché prenda tutte le eccezioni (con `except Exception as e:`) dato che non sono riuscito in nessun modo a far fare a `xmlrpc` raise di qualcosa che non fosse `UnknownException` (ho fatto esperimenti con `globals()`, `__builtins__`, `__builtin__`, `builtins()`, in più e più modi, anche a cascata uno dopo l'altro, ma nulla da fare)

Lati positivi di questo approccio:

- L'eccezione viene presa dalla view, e mostrata nell'area messaggi (non è facile/possibile modificare gli errori del form da dentro la view, e non è possibile accedere a octofusd dalla classe del form)
- C'è il reload di **quasi** tutti i campi in caso di home già presente

Cose che ancora non funzionano:

- Dopo errore di home già esistente il contenuto di **quasi** tutti i campi viene rimesso automaticamente, TRANNE per i campi password (non ho capito il perché)
- Al POST con successo dopo un errore di home già esistente (quindi di fatto dopo aver corretto il campo home) viene creato l'utente ma la home è quella "standard" e non quella immessa dall'utente. mi spiego:
- Username: pippo, home automaticamente suggerita `/home/pippo` (IMPORTANTE: questa home **NON** esiste già), io la correggo in `/home/pluto` (e poniamo il caso che invece questa home esista già, per triggerare l'errore) e faccio POST
- Errore nell'area messages che dice che la home esiste già (comportamento corretto/atteso). Io allora metto ad esempio `/home/pippo2` (che non esiste, per non triggerare l'errore) e faccio POST
- L'utente viene creato, **ma** la home è `/home/pippo` (cioè quella automatica `/home/$USER` anziché quella POSTata da me).

La cosa è visibile nella VM di test, lanciando la webapp da dentro `/var/www/octonet`, che è impostata sul branch `t192` anziché `master`, perché gli esperimenti non sono ancora stati mergiati ovviamente, visto il comportamento non totalmente corretto.

Mi piacerebbe almeno:

- Far fare a octofusd/xmlrpc il raise di un'eccezione specifica, in un modo che django riesca a catchare e capire (per evitare il brutto `except Exception as e:`)
- Avere un workflow corretto in toto (form ricompilato completamente e soprattutto utente creato come da campi POSTati) anche dopo un errore di home già esistente

#7 - 04/21/2017 05:04 PM - Mark Caglienzi

- Status changed from *In elaborazione* to *Commenti*

[54af64db](#):

- Eccezione `FileExistsError` quando la home esiste già
- Se l'errore è questo, l'errore viene injectato nel campo `homeDirectory`, diversamente viene mostrato il classico `messages.error`
- Purtroppo non è semplice fare eccezioni custom perché non c'è un vero punto comune in cui definirle (o va ragionato meglio, tipo in `pyoctofuss` probabilmente, se è una libreria importata sia da `octonet` che da `octofussd`)
- La modalità di decisione su dove far vedere l'errore non è solidissimo (di fatto guarda che sia una `FileExistsError` che contenga 'home' nel messaggio di errore)
- Con il pull di `octofussd` master anziché da pacchetto `debian` sembra risolto anche il problema della home creata in modo errato (come spiegato nel messaggio precedente).
- Resta invece il problema della non ricompilazione dei campi password in caso di home già esistente (ma probabilmente è una funzionalità del browser anziché un bug di `octonet`)

NON mergiato in master ma lasciato in t192

#8 - 04/26/2017 04:00 PM - Enrico Zini

Ho pushato un branch "exceptions" per `pyoctofuss` e uno per `octofussd`.

Ora possiamo mettere le eccezioni che vogliamo in `pyoctofuss` in `octofuss/exceptions.py` e passano per `xmlrpc.py`

Inoltre, in `xmlrpc.py` si possono mettere le eccezioni che vogliamo nella whitelist delle eccezioni serializzabili.

Ho aggiunto un minimo di test a `pyoctofuss` e `octofussd`, che passano.

L'unica cosa che non ho fatto è stato aggiungere il throw di `HomeDirectoryExistsError` nel plugin `users` non mock, perché quello non lo posso testare.

#9 - 04/27/2017 12:14 PM - Mark Caglienzi

- Status changed from *Commenti* to *In elaborazione*

- Assignee changed from *Christopher R. Gabriel* to *Mark Caglienzi*

#10 - 04/27/2017 04:45 PM - Mark Caglienzi

- Status changed from *In elaborazione* to *Commenti*

- Assignee changed from *Mark Caglienzi* to *Elena Grandi*

Grazie a Enrico per il refactoring e la pulizia in `octofussd` e `pyoctofuss`, che agevola le customizzazioni che spiego sotto.

[1ca1ed89](#):

- `pyoctofuss/octofuss/exceptions.py` contiene le eccezioni custom, che vanno aggiunte a `__all__` perché vengano automaticamente whitelizzate (anche questo automatismo è molto utile)

- Questa libreria è importata sia da octofusssd che da octonet
- Nel plugin users di octofusssd (nel file common.py) dove si fa `_fillInNewUser()` ho fatto il raise delle eccezioni custom singole (`HomeDirectoryExistsError`, `HomeDirectoryOutsideHomeError`, e `UserAlreadyExistsError`, definite dentro a `pyoctofuss`) dove necessario, al posto della generica `ValueError`, che rendeva complicato e sporco il capire dove mostrare gli errori lato django
- Nella view di creazione utente (in octonet) ho fatto i catch delle tre eccezioni perché mostrino il messaggio di errore direttamente nel campo relativo ('homeDirectory' o 'user', secondo i casi).

Già che c'ero ho pensato di proseguire con la customizzazione delle eccezioni, visto che adesso è semplice far parlare il backend col frontend in modo che sia chiaro quale campo del form ha causato l'errore stesso, e ho notato problemi, che elencherò in un ticket a parte in modo da poterne parlare.

Il tutto è stato mergiato in master (sia qui in octonet, che in pyoctofuss e octofusssd)

Pacchetti da ribuildare per questo ticket:

- octonet
- pyoctofuss
- octofusssd

#11 - 04/28/2017 11:11 AM - Elena Grandi

- Assignee changed from Elena Grandi to Mark Caglienzi

Buildati e uploadati pacchetti nuovi per octonet, pyoctofuss e octofusssd

#12 - 05/11/2017 10:58 AM - Christopher R. Gabriel

- Status changed from Commenti to Chiuso

Chiudo, per me ok.