



Plone: workflow e sicurezza

Release 0.1 pre-alpha

Luca Fabbri (keul)

September 27, 2016

1	Introduzione	1
1.1	Lo scopo del libro	1
1.2	Per chi è questo libro	1
1.3	Per chi <i>non</i> è questo libro	1
1.4	Che cosa verrà approfondito in questo libro	2
1.5	Cosa <i>non</i> verrà affrontato	2
1.6	Come è strutturato questo libro	2
1.7	Riferimenti alle versione utilizzata	2
2	Gli Utenti	3
2.1	Come si danno poteri agli utenti?	3
2.2	Che poteri predefiniti ha un utente?	3
2.3	L'utente anonimo	6
3	I Gruppi	7
3.1	I gruppi predefiniti di Plone	7
3.2	Quando creare un nuovo gruppo?	9
4	I Ruoli	11
4.1	I ruoli predefiniti	11
4.2	Come usare i ruoli	17
4.3	Creare nuovi ruoli: quando e perché	22
4.4	Come creare nuovi ruoli	25
4.5	Chi può assegnare i ruoli, e quali	28
4.6	Verificare i ruoli di un utente	29
4.7	Portare quanto fatto in un prodotto	31
5	I Permessi	33
5.1	La gestione dei permessi	33
5.2	Il funzionamento dei permessi nei contenuti	35
5.3	Analisi dei permessi esistenti	37
5.4	Dove i permessi incidono sull'interfaccia Plone	50
5.5	Creare nuovi permessi	60
6	Crediti	63
6.1	Autore	63
6.2	Segnalazioni e contatti	63
6.3	Plone e il logo Plone	63

7	Appendice A - tutti i permessi	65
7.1	ATContentTypes Topic: Add ... <i>Criterion</i>	65
7.2	Add portal member	65
7.3	Add portal topics	65
7.4	Allow sendto	66
7.5	Change portal topics	66
7.6	Content rules: Manage rules	66
7.7	Copy or Move	67
7.8	List portal members	67
7.9	List undoable changes	67
7.10	Mail forgotten password	67
7.11	Manage Groups	67
7.12	Manage users	68
7.13	Modify view template	68
7.14	Plone Site Setup: ...	68
7.15	Portlets: Manage own portlets	69
7.16	Portlets: View dashboard	69
7.17	Set own password	70
7.18	Set own properties	70
7.19	Use mailhost services	70
7.20	View Groups	70
7.21	View management screens	71
7.22	iterate : Check ...	71
7.23	Reply to item	71

Introduzione

1.1 Lo scopo del libro

Questo documento si prefigge lo scopo di fornire una guida completa per comprendere le funzionalità, la configurazione e la gestione dei **workflow** e della **sicurezza** nel CMS **Plone**¹.

Il sistema di workflow disponibile in Plone e tutto quello che concerne la sicurezza dei documenti nel CMS sono una tra le sue maggiori attrattive e una tra le caratteristiche più potenti, ma molto spesso incomprese.

Non sono un grande fan della documentazione, poiché questa tende a diventare obsoleta molto velocemente, ma questa parte della tecnologia Plone è la stessa da quando ho iniziato a lavorarvi (o poco è cambiato) e non credo ci saranno grosse novità nel prossimo futuro. Questo a mio parere giustifica lo sforzo.

1.2 Per chi è questo libro

Un programmatore Plone, ma molto spesso più semplicemente un amministratore del sito, *avrebbero* il potere di fare grandi cose e plasmare il funzionamento del sito al proprio volere, ma la documentazione è sparsa ed a volte assente.

Questo libro è pensato per entrambi questi tipi di utente:

- ad uno **sviluppatore Plone** verrà mostrato che cosa è possibile fare senza ricorrere allo sviluppo e i limiti oltre i quali lo sviluppo può intervenire (e in che modo).

Questo vuole evitare situazioni in cui lo sviluppatore non esperto di Plone *non sa* di una determinata funzionalità, e quindi la reinventa in modo non corretto.

- ad un **amministratore** verrà mostrato tutto quello che è possibile fare senza ricorrere allo sviluppo, semplicemente usando prodotti già presenti o aggiuntivi, o configurando a dovere il sistema.

Questo secondo scopo vuole far comprendere agli amministratori quante cose sia possibile fare senza bisogno di richiedere sviluppo dall'esterno.

1.3 Per chi *non* è questo libro

Sebbene io sia uno sviluppatore Plone, questo libro non insegnerà a sviluppare: verranno toccati i minimi argomenti necessari per arrivare allo scopo. Quando l'argomento analizzato inizierà ad addentrarsi nei meandri della programmazione verranno date le minime informazioni necessarie, fornendo poi riferimenti esterni, nel caso il lettore volesse approfondire oltre.

¹ <http://plone.org/>

Qualche riga di codice potrebbe anche essere spiegata, ma sarà sempre ridotta all'osso e mai ben approfondita.

1.4 Che cosa verrà approfondito in questo libro

Verranno mostrare tutte le funzionalità riguardanti la sicurezza di Plone, partendo dalla **gestione degli utenti e dei gruppi**, per arrivare all'analisi dei **ruoli** che Plone fornisce di base e ad una spiegazione di come questi vanno intesi ed interpretati.

Verranno poi introdotti i **permessi** e il loro ruolo nella sicurezza, portando infine l'utente al cuore della sicurezza dei contenuti in Plone: il **workflow**. Verranno mostrati i workflow predefiniti del CMS, i loro limiti e difetti e il come superarli.

Infine verrà insegnato come creare nuovi workflow, come disegnarli e far fare loro cose non sempre semplici.

1.5 Cosa *non* verrà affrontato

Questo libro vuole essere *diretto*, ed eviterà tutti quegli argomenti che non ritengo importanti. Non verranno spiegati i “dogmi”, non verranno date inutili definizioni di qualcosa che (mi aspetto) sia già conosciuto dal lettore.

Il linguaggio sarà esso stesso *diretto*.

1.6 Come è strutturato questo libro

Il libro parte dalla “superficie”, mostrando Plone come si presenta una volta installato e spiegandone il funzionamento (e la *configurazione*) di base. Da qui si prenderà lo spunto per le prime riflessioni e domande su cosa sia possibile personalizzare, il che ci porterà a scavare sotto la superficie e sarà trampolino di lancio per personalizzare lo strumento e spingersi quindi verso i confini con la programmazione.

Sebbene il focus centrale degli argomenti sia il CMS Plone, gran parte degli argomenti mostrati sono applicabili alle tecnologie sottostanti (*Zope*², *CMF*³, ...), ma per semplicità di lettura queste differenze di tecnologia *non* saranno evidenziate nel libro.

1.7 Riferimenti alle versione utilizzata

Quanto mostrato è applicabile alla versione 4 di Plone (specificatamente: *Plone 4.2.x*⁴), ma sono quasi certo che se la vostra versione di Plone dovesse essere maggiore, una grossa percentuale di quanto qui riportato sarà comunque estremamente utile.

² <http://zope.org/>

³ <http://pypi.python.org/pypi/Products.CMFCore>

⁴ <http://plone.org/products/plone/releases/4.2>

Gli Utenti

Per quanto poco ci possa essere da dire sugli utenti, qualcosa vale la pena venga definito.

Un **utente** è un visitatore del sito che ha eseguito l'**autenticazione**, che possiede quindi un account per accedere al sito.

In alcuni siti, dove è abilitata l'autoregistrazione degli utenti, chiunque può diventare utente del sito.

Ogni utente del sito ha (di solito) il **ruolo Collaboratore (Member)**.

Ci possono essere eccezioni? Sì. Altre basi dati di utenti che non siano quella predefinita di Plone (queste fonti si ottengono tramite **plugin PAS**) forniscono di solito questo ruolo, ma nessuno vieta di configurare a dovere il plugin per fornire anche altri ruoli, o nessun ruolo.

2.1 Come si danno poteri agli utenti?

Gli utenti acquisiscono diversi poteri perché vengono loro assegnati **ruoli** oppure perché, facendo parte di **gruppi**, acquisiscono ruoli dati al gruppo.

Gli utenti non possono avere direttamente **permessi**.

2.2 Che poteri predefiniti ha un utente?

Ci si potrebbe aspettare che il possedere un account in un sito Plone fornisca qualche tipo di potere, ma questo *non è scontato*.

L'utente autenticato ha di certo (a meno di personalizzazioni poco comuni) una differenza nell'interfaccia grafica del sito: compare il suo nome e l'accesso alle proprie preferenze personali.

Per il resto (e nella versione attuale di Plone questo è vero di partenza) un utente del sito non ha di partenza nessun altro potere.

L'utente inizia ad acquisire poteri nel momento in cui gli vengono assegnati dei ruoli (direttamente, o tramite un gruppo).

2.2.1 Le cartelle personali

In alcuni siti l'amministratore potrebbe aver abilitato l'uso delle **cartelle personali** degli utenti, una speciale posizione del sito da considerarsi come la "casa" dell'utente (un po' come la directory "home" comune in tutti i sistemi operativi odierni).

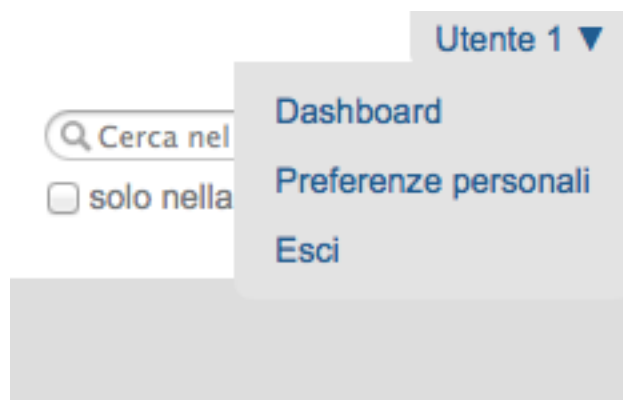


Fig. 2.1: Il menù degli strumenti personali, espanso

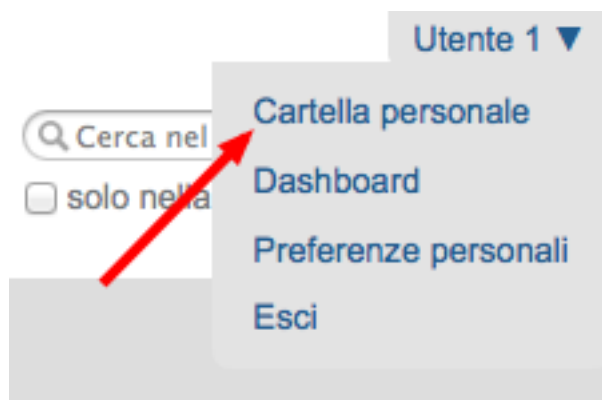


Fig. 2.2: Il link per raggiungere la propria cartella personale

Questa impostazione ad oggi è disattivata nelle impostazioni predefinite ma può ancora essere attivata dalle *impostazioni di sicurezza* del sito.

[Configurazione del sito >](#)

Impostazioni sicurezza

Impostazioni sulla sicurezza in questo sito.

— Impostazioni sicurezza —

☐ **Consenti l'auto-registrazione**

Consenti agli utenti di iscriversi al portale. Se non viene selezionato, solo i manager potranno aggiungere nuovi utenti.

☒ **Permetti agli utenti di scegliere la propria password**

Se non selezionato verrà generato un URL che sarà inviato per email. Gli utenti verranno avvisati di cliccare sul link presente nella mail per completare la registrazione al sito e cambiare la propria password; questa procedura verifica inoltre che sia stato inserito un indirizzo email valido.

☐ **Abilita le cartelle utenti**

Se selezionato, le cartelle personali dove gli utenti possono inserire contenuti verranno create al loro primo accesso.

☐ **Consenti a chiunque di vedere le informazioni personali**

Se non selezionato solo gli utenti riconosciuti potranno consultare le informazioni relative a chi ha creato un certo contenuto e quando è stato modificato.

Fig. 2.3: Come abilitate le cartelle personali degli utenti

Che cosa può fare un utente comune nella propria cartella personale? Come ripeterò molto spesso nel resto del libro: dipende. L'unica cosa che succede sempre è che l'utente ha il ruolo di **Possessore (Owner)** della cartella e quindi acquisisce tutti i poteri del ruolo quando lavora in quella sezione.

In una installazione base questo significa che l'utente è in grado di creare qualunque contenuto egli voglia all'interno della propria cartella e, con i workflow che oggi sono predefiniti in Plone, pubblicare i contenuti.

C'è sono vari motivi per cui questa sezione è ora disabilitata di base:

- si evita l'inutile creazione di cartelle (nei siti dove queste non sono in realtà mai usate)
- si evita che l'utente “salti” il workflow del sito e il sistema di revisione.

Il secondo punto è il più importante. Nella mia esperienza (soprattutto con le vecchie versioni di Plone, precedenti alla 3, dove questa impostazione era di base abilitata e ci si dimenticava di disattivarla) questa impostazione provocava vari problemi.

Quando si disegna un sito con un workflow per ospitare una redazione molto complessa, si passa diverso tempo ad ideare una struttura del sito più o meno complessa, dove utenti e gruppi abbiano compiti ben definiti. Ad esempio: l'ipotetico “Ufficio 5” può avere una sua cartella e si vuole che gli utenti scrivano lì dentro i contenuti relativi all'ufficio.

Molto spesso gli utenti trovano la strada più semplice: si accorgono di avere una cartella personale dove possono scrivere i propri documenti... e lì scrivono. Al termine del lavoro di solito viene chiesto agli amministratori di spostare i contenuti altrove. In questo modo non c'è bisogno di imparare come funziona il workflow del sito o dove si trova la propria area di lavoro.

Quanto usare le cartelle personali

La cartella personale è una bella funzionalità che va valutata e che può tornare utile, ma di solito va presa in considerazione assieme ad una modifica dei workflow e della sicurezza del sito.

Se il vostro sito vuole creare una community dove però ogni utente deve avere il proprio spazio, la cartella personale può essere una strada molto comoda.

2.3 L'utente anonimo

Anche *il visitatore anonimo è un utente* e possiede un ruolo speciale: **Anonimo (Anonymous)**. L'unica differenza con un utente autenticato è che non ha link agli strumenti personali o un'area dove salvare le proprie preferenze, e che non identifica un singolo visitatore ma una intera schiera di visitatori.

Sia che stiate realizzando una sito pubblico o una intranet, *questo utente va previsto*.

In un **sito pubblico** l'utente anonimo è quello che genera la maggior parte del traffico (i **bot e crawler** dei motori di ricerca sono utenti anonimi). Va capito cosa e cosa non possono vedere, e la sicurezza del sito deve essere calibrata su questa idea.

Per **una intranet** le cose possono sembrare più semplici, e in certi contesti lo sono, ma non ignorate il problema. Ho visto casi dove il fatto che una intranet non sia accessibile al pubblico lasciava intendere di essere al sicuro, per poi accorgersi che una qualunque persona in grado di collegare il proprio computer alla rete dell'azienda o dell'ente potesse poi accedere a documenti importanti.

In altri casi ho visto come l'obbligo di autenticazione di una intranet venisse ottenuto limitandosi a rendere privata la home page del sito. Ci si accorgeva poi col tempo che un utente fosse in grado di saltare la home page e accedere direttamente ad altre pagine (ad esempio: la pagina di ricerca di Plone) e fosse quindi in grado di vedere documenti che si credevano segreti.

Warning: Il “ <i>workflow intranet</i> ” fornito da Plone può non essere ottimale per i vostri scopi!

In questi casi la ricerca di Plone può darvi da subito un'idea della situazione ed aiuta molto a trovare potenziali problemi ma può non bastare poiché esistono tipi di contenuto che sono configurati per non essere “ricercabili”. La loro visibilità va comunque verificata

I Gruppi

I gruppi sono definibili semplicemente come un **accorpamento di utenti**. A basso livello, nei meandri del codice Plone, molto spesso gruppi e utenti sono visti allo stesso modo.

I gruppi sono estremamente utili nel momento in cui è necessario assegnare *ruoli locali* qua e là per il sito.

Solo gli utenti reali possono far parte dei gruppi, l'utente anonimo non è identificabile in nessun gruppo.

3.1 I gruppi predefiniti di Plone

Alla creazione di un sito Plone troverete già **quattro gruppi predefiniti**, tre di questi hanno assegnati dei **ruoli globali**, che non ha di solito senso modificare, ma può aver senso eliminare il gruppo o evitare di usarlo (nessuno vi obbliga). Se uno o più di questi gruppi non serve: cancellatelo! Non ci sono problemi.

Elenco generale dei gruppi

[Torna alle impostazioni di Plone](#)

I gruppi sono insiemi logici di utenti che condividono una certa caratteristica, come un dipartimento, o un ufficio. Non sono direttamente legati ai permessi, in generale: normalmente per quello scopo si usano i ruoli, assegnandoli poi ai vari gruppi. Il simbolo  indica un ruolo che eredita i permessi da un gruppo

Aggiungi un nuovo gruppo

Nome del gruppo	Ruoli							Rimuovi gruppo
	Contributore	Editor	Collaboratore	Lettore	Revisore	Amministratore del sito	Manager	
 Administrators (Administrators)	☐	☐	☐	☐	☐	☐	☒	☐
 Authenticated Users (Virtual Group) (AuthenticatedUsers)	☐	☐	☐	☐	☐	☐	☐	☐
 Reviewers (Reviewers)	☐	☐	☐	☐	☒	☐	☐	☐
 Site Administrators (Site Administrators)	☐	☐	☐	☐	☐	☒	☐	☐

Fig. 3.1: Come si presenta la gestione dei gruppi

3.1.1 Il gruppo “Administrators”

Il gruppo degli amministratori è estremamente utile, anche se con la versione 4.1 di Plone e l'introduzione del nuovo ruolo **Amministratore del sito (Site Administrator)** ha forse perso parte della sua importanza.

Nella configurazione base (e non ho mai trovato motivi per cambiare questo comportamento) il gruppo possiede nativamente il ruolo di **Manager**, il ruolo più potente di Zope.

Il ruolo di *Manager* non è un ruolo pensato per essere dato localmente a sezioni del sito, quindi la presenza di questo gruppo e il fatto che il ruolo ad esso assegnato sia un ruolo globale è una scelta corretta.

All'interno del gruppo vanno inseriti tutti gli utenti (pochi... o nessuno) che devono avere questo potere.

3.1.2 Il gruppo “Site Administrators”

Il gruppo degli amministratori del sito è stato introdotto di recente, con l'introduzione del nuovo ruolo **Amministratore del sito (Site Administrator)**.

Le osservazioni fatte per il gruppo *Administrators* si ripetono qui: è un gruppo estremamente utile e vi vanno inseriti tutti gli utenti del sito che dovranno gestire il sito in futuro.

3.1.3 Il gruppo “Reviewers”

Il gruppo dei revisori dei contenuti, che di solito hanno poteri importanti nei meccanismi di pubblicazione. Tutti i componenti di questo gruppo hanno il potere di **Revisore (Reviewer)** nel sito.

A differenza degli altri due gruppi descritti precedentemente, l'utilità di questo gruppo non è assicurata.

Lo consiglio in presenza di piccoli siti, dove la redazione è estremamente limitata ed in effetti chi revisiona i contenuti lo fa in tutto il sito. In tal caso questo gruppo può servire.

Va però tenuto presente che il ruolo di *Revisore* è assegnato in modo globale, quindi bisogna anche avere la certezza che non ci siano eccezioni di sorta (nessuna cartella “speciale” dove il gruppo *non* deve avere questo potere).

In molti altri casi, dove ci sono gruppi separati e redazioni separate, questo gruppo finisce sempre per rimanere vuoto. A questo punto consiglio di eliminarlo: non ci sono problemi nel farlo e non otterrete altro che evitare problemi o possibili errori.

3.1.4 Il gruppo virtuale “Authenticated Users”

Questo gruppo è un gruppo *virtuale*, non un vero gruppo e non può essere eliminato. Rappresenta tutti gli utenti autenticati nel sito: non appena un utente esegue il login, diventa parte del gruppo.

Warning: Questa funzionalità è **pericolosa** e va capita fino in fondo

Dare un qualunque potere a questo gruppo (soprattutto nel pannello di controllo dei gruppi, quindi come *ruolo globale*) significa dare il ruolo a *chiunque* ha un account nel sito.

Il suo uso può invalidare altre configurazioni fatte per altri gruppi o altri utenti.

Un esempio: se assegnassimo un qualunque ruolo al gruppo “*Authenticated Users*”, per esempio **Lettore**, questo renderebbe inutile l'aver assegnato (o assegnare in futuro) quello stesso ruolo a veri utenti e gruppi (globalmente o localmente: non importa). È inutile dire “il gruppo dell'Ufficio 5 è *Lettore* sulla cartella /uffici/ufficio-5/documenti-privati se *tutti* gli utenti del sito hanno quel potere e possono quindi leggere da quella cartella.

Il rischio è trovarsi in una situazione in cui non si capisce come mai anche i membri del gruppo “Ufficio 3” possono leggere i documenti riservati in quella sezione.

Perché non lo consiglio?

Questo gruppo è usato molto spesso a sproposito perché *sembra* facile da capire e usare, più semplice che comprendere fino in fondo come configurare per bene i workflow di Plone e verificato lo stato di revisione dei contenuti del sito.

Ne potete probabilmente fare a meno.

3.2 Quando creare un nuovo gruppo?

La risposta breve: **sempre!**

I gruppi sono una bella funzionalità e limitano i problemi della gestione di utenti per strutture complesse.

Tornando all'esempio del nostro “Ufficio 5”, se ammettessimo che questo ufficio è composto da 20 persone e non avessimo i gruppi, saremmo spesso costretti a gestire tutte insieme questo insieme di utenti. Non appena un utente lascia o entra nell’“Ufficio 5”, dovremmo ricordarci dove questo aveva poteri per modificarli oppure trovare dove i suoi colleghi ne hanno per fornirgli gli stessi poteri.

Poter invece gestire tutti questi utenti, che nella vita reale sono già un gruppo, come un'unica entità, ha una comodità senza lati negativi.

Gli utenti possono far parte di più gruppi senza alcun limite. Gli utenti del nostro “Ufficio 5” possono essere spezzati in vari sotto-gruppi (“Ufficio 5: capi ufficio”, ...) eppure la nostra organizzazione può prevedere anche gruppi trasversali all'idea di ufficio (“Gruppo Calchetto”, a cui partecipano vari utenti di più uffici).

Non disdegnate nemmeno l'idea di creare gruppi atti a contenere un solo utente (“Custode”) perché molto spesso l'appartenenza al gruppo definisce una certa mansione dell'utente. Se l'utente cambia il suo profilo all'interno del sito è molto più semplice cambiare la persona del gruppo che modificare impostazioni qua e là nel sito stesso.

I Ruoli

La miglior definizione di “ruolo Plone” che posso trovare è questa:

I ruoli sono un accorpamento di **permessi** e quindi rappresentano un insieme di **poteri** di un utente nel sito.

Quando assegnate un ruolo ad un utente, state fornendo in realtà una serie di permessi (il vero motore della sicurezza di Zope). Come già detto in precedenza, **non potete in nessun modo assegnare permessi agli utenti o ai gruppi, ma potete solo assegnare ruoli.**

La funzionalità dei ruoli è una delle prime caratteristiche che fanno assaporare la potenza del sistema sottostante. Prendiamo ad esempio in ruolo del **Lettore** (che verrà introdotto meglio in seguito): cosa significhi essere un “lettore” non è qualcosa scritto nella roccia; siti diversi (e senza dover per forza sviluppare del codice) possono differire nella definizione aumentando o limitando i poteri (permessi) del ruolo.

Va da sé che il nome di un ruolo deve poter ricordare a chi lo andrà ad utilizzare il suo scopo.

4.1 I ruoli predefiniti

Plone (e il framework Zope sottostante) forniscono una serie di ruoli predefiniti il cui funzionamento è bene approfondire e comprendere. Sebbene sia possibile creare nuovi ruoli l’eventualità di doverlo fare è meno frequente di quanto si pensi (anche se non da escludere a priori).

Warning: Uno errore molto comune nelle applicazioni Plone mal scritte è definire una grande serie di nuovi ruoli. **Più ruoli aggiungete più complesso diventa gestire la sicurezza del sito.**

4.1.1 I ruoli forniti di Zope

I ruoli forniti da Zope sono quattro e non sono modificabili o eliminabili. Sulla loro presenza si basa il funzionamento della sicurezza dell’application server, e quindi di Plone.

Anonimo (Anonymous)

Anonimo (Anonymous) è il ruolo assegnato agli utenti anonimi. È un ruolo speciale ed ha un comportamento diverso da tutti gli altri ruoli. Non è infatti possibile definire un permesso accessibile dal ruolo Anonymous ma non da altri ruoli: un potere dato ad un anonimo è di certo fornito anche a *qualunque* utente del sito.

Roles			
Anonymous	Authenticated	Manager	Owner
☐	☐	☒	☐
☐	☐	☒	☐
☐	☐	☒	☐
☐	☐	☒	☐
☐	☐	☒	☐
☐	☐	☒	☐
☐	☐	☒	☐
☐	☐	☒	☐
☐	☐	☒	☐
☐	☐	☒	☐

Fig. 4.1: I quattro ruoli Zope, visti da ZMI

Il ruolo Anonimo è per sua natura associato con il tipo di utente anonimo. Nel disegnare workflow per i vostri contenuti questo ruolo assume parecchia importanza: un errore nell'assegnare un permesso di troppo a questo utente e la sicurezza del vostro sito potrebbe essere compromessa.

Autenticato (Authenticated)

Autenticato (Authenticated) è il ruolo automaticamente assegnato ai visitatori del sito che hanno eseguito un qualunque tipo di autenticazione, anche tramite [autenticazione basic](http://en.wikipedia.org/wiki/Basic_access_authentication)⁵ via ZMI.

Un utente con ruolo *Autenticato* è quindi un utente a livello Zope.

Questo ruolo non è solitamente molto utilizzato nei workflow di Plone preferendovi, per correttezza, il ruolo di *Collaboratore*.

Manager

L'Alpha e l'Omega dei ruoli. Chi ha ruolo **Manager** ha solitamente potere assoluto nel sito Plone ed entra senza restrizioni in ZMI.

È un ruolo ovviamente pericoloso e non va mai assegnato a sproposito. Come regola generale, se il vostro utente *non ha* bisogno di accedere alla ZMI, *Manager* non è il ruolo migliore da assegnargli ma va preferito il ruolo di *Amministratore del Sito*.

È lecito avere installazioni di Plone dove esiste un solo utente con questo ruolo: **admin**, l'utente predefinito a livello Zope che è di solito il creatore dei siti Plone.

Note: A differenza del ruolo *Anonimo* (il ruolo con meno poteri), la sua natura di essere il ruolo “con i super poteri” non è predeterminata.

⁵ http://en.wikipedia.org/wiki/Basic_access_authentication

Un esempio: sarebbe possibile dare il permesso di vedere un contenuto ad un *Collaboratore* del sito ma non al *Manager*. Questo è possibile, anche se totalmente illogico e sconsigliato. Il *Manager*, avendo poi accesso alla ZMI e quindi al sistema di associazione di ruoli e permessi, potrebbe poi ri-assegnarsi il permesso mancante.

Possessore (Owner)

Il concetto di **Possessore (Owner)**, per quanto orribilmente tradotto in italiano, nasce a livello Zope. Un primo esempio: l'utente *admin* ha di solito il ruolo di *Owner* sull'“oggetto sito Plone” poiché solitamente è questo utente che crea i nuovi siti all'interno del database di Zope.

È un ruolo che va ben compreso:

- di solito deve essere assegnato ad un solo utente
- è possibile fornirlo a più utenti (ciò oggi è fortunatamente più difficile da farsi da interfaccia Plone, mentre in versioni precedenti del CMS era purtroppo un modo di operare molto comune) ma bisogna essere certi di che cosa si sta facendo
- è possibile avere a che fare con workflow dove il *Possessore* non ha importanza (o sarebbe meglio non l'avesse)

In una configurazione base, un *Possessore* mantiene un certo livello di potere sui propri contenuti. Detto in poche parole: può modificarli e poi sottoporli a revisione (ma questo dipende molto dal workflow).

Possessore e Creatore

Nella maggior parte dei casi è un ruolo che è direttamente associato con il creatore del contenuto. Se “Utente 1” crea una pagina, Plone lo rende anche *Possessore* della pagina stessa.

Questo si può vedere anche dal campo “*Creatori*” comune a tutti i contenuti Plone, ma non bisogna farsi trarre in inganno: il valore di questo campo è solo un'informazione testuale che può essere facilmente modificata.

Modifica Pagina

Default ■ Categorizzazione Date **Proprietario** Impostazioni

Creatori
 Persone responsabili della creazione del contenuto di questo elemento. Inserisci un elenco di nomi, uno per riga. L'autore principale dovrebbe essere messo al primo posto.

utente1

Fig. 4.2: La vista del campo “*Creatori*”, nelle informazioni di “*Possessore*”

Cambiando il valore di “*Creatori*” con un altro utente del sito non assegna il ruolo di *Possessore* al nuovo utente specificato. Il fatto che tale campo sia nell'insieme dei campi raggruppati sotto la sezione “*Proprietario*” non fa altro che aumentare la confusione.

Cambiare Possessore di un contenuto è possibile

Le recenti versioni di Plone hanno reso più difficile assegnare questo ruolo a sproposito a più utenti ma rimane possibile (e lecito) cambiare proprietario di un contenuto.

Esiste una vista speciale, raggiungibile solo conoscendone l'URL (una particolarità introdotta, a mio parere per errore, in Plone 3): `ownership_form`. Questa vista va lanciata sul contesto del documento al quale si vuole cambiare proprietario e permette di modificare l'utente che ha ruolo di *Possessore* sul contenuto.

Cambia proprietario

Questo modulo consente di trasferire il possesso di un elemento a qualcun altro.

L'attuale proprietario è **utente1**.

— Cambia il proprietario —

Nuovo proprietario ■

Modifica il proprietario dell'oggetto corrente.

Nuovo proprietario

Fig. 4.3: La vista “change_ownership” lanciata su un contesto

Esiste un comodissimo prodotto che permette di manipolare in blocco il ruolo di *Possessore* e volendo anche il campo “Cratori” per più contenuti del sito: plone.app.changeownership⁶.

Quanto è importante il ruolo Possessore?

Dipende.

Nel momento della creazione di un contenuto questo ruolo ha di certo importanza, poiché ovviamente l'utente che sta salvando per la prima volta il documento deve avere i poteri di modifica. Nel seguito invece la sua importanza dipende dalla natura del vostro sito.

Se state realizzando la sicurezza di un tipo di contenuto dove, per sua natura, il creatore è importante (ad esempio: il contenuto rappresenta la prenotazione di un'auto aziendale) allora il creatore continua ad avere una grande importanza per tutto il ciclo di vita del contenuto.

Se i poteri che un utente deve avere su un contenuto dipendono dal suo stato o dalla sua appartenenza ad un gruppo allora il *dato* relativo al creatore può avere la sua importanza, ma la persona che ha creato il contenuto no.

Un esempio: l'Utente 1 ha scritto un documento mentre lavorava per l'Ufficio 5. Poco importa chi abbia creato il documento, ma dopo la sua creazione l'utente non deve avere permessi particolari sul contenuto, o di certo non deve continuare a mantenerli se in futuro lascerà l'Ufficio 5.

Warning: Anche in questo caso i workflow base di Plone non sono ottimali per tutte le situazioni.

Se volete maggiori dettagli su questo argomento, l'ho affrontato lungamente nel mio articolo [Plone, security and workflows: when rely on Owner role is bad](#)⁷ (in lingua inglese).

4.1.2 I ruoli definiti da Plone

Plone è un'applicazione costruita sull'application server Zope. Per raggiungere i suoi scopi esso definisce di partenza alcuni ruoli aggiuntivi.

⁶ <http://plone.org/products/plone.app.changeownership>

⁷ <http://blog.keul.it/2011/09/plone-security-and-workflows-when-rely.html>

La differenza principale con i ruoli di Zope visti alla sezione precedente è che questi ruoli *non* sono necessari per il funzionamento di Zope (e in realtà nemmeno di Plone).

Plone dà alcuni “suggerimenti” su una configurazione ottimale, non troppo semplice né troppo complessa. I ruoli forniti di Plone sono ottimi per la maggior parte delle configurazioni e permettono di avere un minimo meccanismo di revisione e una buona suddivisione delle competenze.

Contributore (Contributor)

Il **Contributore** (un altro ruolo la cui traduzione ufficiale italiana lascia a desiderare) è la persona che “porta contributi al sito”. Una traduzione migliore sarebbe probabilmente quella di *Autore* o *Redattore*.

Il *Contributore* è una persona che può inserire nuovi contenuti nel sito. Nella configurazione predefinita di Plone, questo include i permessi per inserire *tutti* i contenuti (ad esclusione delle **Collezioni**).

Il Contributore può modificare i propri contenuti?

Nella configurazione iniziale di Plone, la risposta è sì.

Questo potere però non dipende dal ruolo di *Contributore* stesso e dai suoi poteri ma dal fatto che un utente che crea un contenuto ne diventa *Possessore*.

Questo concetto è molto importante.

Editor

L'**Editor** è un utente che ha poteri di *modifica* sui contenuti. E qui ci si ferma!

Un *Editor* può modificare quindi *tutti* i contenuti su cui ha potere, ma non è nella sua natura creare nuovi contenuti.

Nella mia idea di sito un editor deve poter aggiungere e modificare

Non siete gli unici. Questo in Plone può essere fatto in due modi.

Il primo sarebbe quello di modificare i poteri del ruolo *Editor* per fornirgli anche i poteri del *Contributore*. Il modo che però consiglio è quello di **assegnare al vostro utente due ruoli**: il ruolo di *Editor* e **Contributore**.

Editor e le Collezioni

Un editor può modificare anche le collezioni (che un *Contributore* non potrebbe normalmente creare. Questa particolarità non è ben giustificabile e credo crei un po' di confusione (ad ogni modo: è solo una configurazione di base, che può essere facilmente modificata).

Per di più: prima di Plone 4.2 (con le vecchie collezioni) la modifica si limitava ai soli campi del “contenuto collezione” ma non ai criteri, che comparivano in un’altro tab; nelle nuove collezioni chi può modificare una collezione ha potere anche sui criteri.

Il motivo sarà discusso in seguito nel *capitolo dei permessi*.



Fig. 4.4: Come si presenta il tab dei “Criteri” nei cercatori vecchio stile

Collaboratore (Member)

Il **Collaboratore** è l’utente autenticato nella concezione di Plone (che si distingue dal ruolo di *Autenticato* definito da Zope, visto in precedenza).

La presenza di questo doppio ruolo crea qualche confusione. Di base questo ruolo viene fornito automaticamente a tutti gli utenti del sito, come *ruolo globale*.

Nota che i ruoli vengono applicati direttamente all’utente. Il simbolo  indica un ruolo che eredita i permessi da un gruppo.

Aggiungi un nuovo utente

Ricerca collaboratore: Cerca Mostrali tutti

Nome utente	Ruoli							Azzera la password	Rimuovi utente
	Contributore	Editor	Collaboratore	Lettore	Revisore	Amministratore del sito	Manager		
 Utente 1 (utente1)	☐	☐	☒	☐	☐	☐	☐	☐	☐

Applica le modifiche

Fig. 4.5: Il ruolo “Collaboratore” dato a tutti gli utenti

Il *Collaboratore* non è un ruolo speciale. Basi dati utente aggiuntive (LDAP, RDBMS) di solito forniscono questo stesso ruolo automaticamente. In pratica quando si vogliono dare poteri agli utenti autenticati nel sito Plone bisogna riferirsi a questo ruolo, che va preferito al ruolo *Autenticato* visto in precedenza.

Cosa succede se un utente non è Collaboratore?

Per quanto detto dell’*Autenticato* e del *Collaboratore* si può concludere che è possibile avere utenti del sito sprovvisti del ruolo *Collaboratore* (non è possibile il contrario invece).

Plone continua a funzionare a dovere (ci sono in effetti piccole differenze, funzionalità che in questa configurazione avrebbe solo il *Collaboratore*).

Può servire una simile impostazione? In effetti sarebbe possibile definire in questo modo utenti del sito di primo e di secondo livello, dove gli utenti con ruolo *Autenticato* hanno minori poteri.

Tenete sempre presente che si sta comunque parlando di due ruoli di basso livello (non creano contenuti, non gestiscono documenti, ...).

La possibilità c’è.

Lettore (Reader)

Nel significato che Plone dà al ruolo **Lettore** c'è il poter “vedere”, che si traduce (di base) in poter accedere a contenuti normalmente non visibili. Va usato per assegnare ad utenti del sito un'anteprima di un lavoro in corso o l'accesso permanente ad un'area privata. Tutto questo senza fornire poteri di modifica di nessun tipo.

Il lettore è un ruolo interessante ed utile, ma non è detto che sia necessario nel vostro portale. Dal punto di vista della “scala dei poteri” questo ruolo è appena sopra la coppia *Autenticato/Collaboratore*.

Revisore (Reviewer)

Il **Revisore** assume importanza solo in presenza di un processo di pubblicazione. Il *Revisore* normalmente non crea contenuti ma lavora sui contenuti altrui: li revisiona.

Ha di solito il potere di accettare il lavoro svolto (di solito: la richiesta di pubblicazione) o rifiutarlo: agisce sul **workflow**.

Un altro potere che (normalmente) gli viene assegnato è la **gestione delle parole chiave**.

Anche questo ruolo potrebbe non servire nel vostro sito: come tutto in Plone, dipende dal vostro ambiente e dai vostri scopi.

Amministratore del sito (Site Administrator)

Questo ruolo è stato introdotto con Plone 4.1, e per ottimi motivi. Il suo scopo è dare poteri assoluti agli utenti Plone, senza dar loro poteri definiti “di programmazione” (che si traduce normalmente con l'accesso alla ZMI).

Di questo ruolo se ne sentiva la mancanza. È normale che il vostro cliente, l'azienda che vi ha commissionato un'applicazione basata su Plone voglia avere utenti con “poteri assoluti” (per l'appunto gli “amministratori del sito”).

Il problema un tempo era non dare poteri inutilmente pericolosi: Alla ZMI deve avere accesso solo un utente che ne abbia effettivamente bisogno.

Attualmente: un ruolo poco supportato

Spero che questo paragrafo diventi velocemente deprecato ma al momento le cose vanno così: molti prodotti vengono aggiornati senza fornire supporto al ruolo *Amministratore del sito*, oppure basandosi su permessi che questo ruolo non ha (ma che invece ha il *Manager*). Vedere la descrizione del permesso “*Manage portal*”.

Col tempo andrà meglio.

4.2 Come usare i ruoli

4.2.1 I ruoli globali

Il modo più facile per gestire i ruoli è direttamente dalla gestione “Utenti e gruppi”. Da queste pagine infatti è possibile vedere tutti i ruoli ed è la prima cosa che un amministratore vede dopo aver aggiunto un utente o creato un gruppo.

Questa “facilità” di lavoro trae in inganno e fa sì che gli amministratori *credano* che queste pagine siano il modo giusto di procedere.

No! Evitate i ruoli globali.

Elenco generale degli utenti

[Torna alle impostazioni di Plone](#)

Clicca sul nome di un utente per vedere e modificare i suoi dati. Da questo form puoi anche direttamente modificare gli indirizzi e-mail ed aggiungere o rimuovere utenti.

Nota che i ruoli vengono applicati direttamente all'utente. Il simbolo  indica un ruolo che eredita i permessi da un gruppo.

Aggiungi un nuovo utente

Ricerca collaboratore:

Nome utente	Ruoli							Azzera la password	Rimuovi utente
	Contributore	Editor	Collaboratore	Lettore	Revisore	Amministratore del sito	Manager		
 Utente (utente1)	☐	☐	☒	☐	☐	☐	☐	☐	☐
 Utente 2 (utente2)	☐	☐	☒	☐	☐	☐	☐	☐	☐

Fig. 4.6: La visione dei ruoli globali dal pannello di controllo degli utenti

I ruoli globali sono dannosi perché molto spesso nascondono una tra le più grandi funzionalità di Plone: **la condivisione di un contenuto o una sezione del sito**.

Per di più, i ruoli globali sono **assoluti** e non possono in nessun modo essere bloccati. Questo significa che se assegnate un ruolo globale ad un utente o un gruppo, quell'utente o gruppo avrà il potere assegnatogli in tutto il sito, senza eccezioni.

Per concludere: sconsiglio di usare i ruoli globali, soprattutto per i singoli utenti.

Eccezioni utili

Le eccezioni ci sono.

La prima eccezione è per l'assegnazione del ruolo di *Collaboratore* agli utenti, che in una configurazione normale diventa appunto una proprietà dell'utente che non ha limitazioni in nessuna sezione del sito: un utente del sito è utente del sito ovunque (nota bene: questo non significa che l'utente debba avere accesso a tutte le aree del sito).

La seconda eccezione vale per alcuni gruppi, come indicato quando si sono presentati i gruppi predefiniti di Plone. Ci sono alcuni gruppi che, per natura, definiscono poteri globali: l'ipotetico gruppo dei "Redattori Ufficio 5" non deve probabilmente avere nessun potere globale, ma per un gruppo come gli Amministratori del Sito la cosa è diversa.

L'unica eccezione che sconsiglio sempre è l'assegnazione di altri poteri che non siano quelli di *Collaboratore* a qualunque utente. Se ci possono essere eccezioni per i gruppi, per gli utenti no. Consiglio piuttosto di creare un gruppo dove porre questo utente e dare i poteri al gruppo.

4.2.2 I ruoli locali (condivisione)

Il modo che consiglio per gestire le assegnazioni dei ruoli nel vostro sito è il pannello della condivisione. Proseguiamo l'esempio mostrando la condivisione di una cartella del sito che dovrebbe essere l'area di lavoro dell'"Ufficio 5", all'interno di una macro-area che racchiude tutti gli uffici.

Fate particolare attenzione alle *briciole di pane* (breadcrumbs), che ci permettono sempre di comprendere la nostra posizione all'interno del sito.

Tu sei qui: [Home](#) / [Uffici](#) / [Ufficio 5](#)

Contenuti
Visualizza
Modifica
Regole
Condivisione

Condivisioni di “Ufficio 5”

Puoi controllare chi può visualizzare e modificare l'elemento usando l'elenco che segue.

Nome	Può aggiungere	Può modificare	Può revisionare	Può vedere
Administrators	☐	☐	☐	☐
Direzione	☐	☐	☐	☒
Utenti collegati	☐	☐	☐	☐
Site Administrators	☐	☐	☐	☐

☒ **Eredita i permessi dai livelli superiori**

Di norma, i permessi di questo elemento vengono ereditati dal contenitore. Se disabiliti questa opzione, verranno considerati solo i permessi di condivisione definiti esplicitamente. Nel sommario, il simbolo indica una impostazione ereditata. Analogamente, il simbolo indica un ruolo globale. Questi sono gestiti dall'amministratore del sito nelle impostazioni del portale.

Fig. 4.7: La vista della condivisione di un elemento

La descrizione “Puoi controllare chi può visualizzare e modificare l'elemento usando l'elenco che segue.” che leggete nell'immagine, di certo facilita a comprendere che cosa si può fare in questa vista ma è limitativa perché vale solo per la configurazione base di Plone.

Nella realtà da questo modulo si possono controllare tutti i ruoli, anche quelli non compresi in una installazione base.

Il pannello della condivisione mostra sempre una tabella riassuntiva sullo stato dei ruoli assegnati nel contesto. La lista può anche essere inizialmente vuota ma si popola automaticamente in presenza di impostazioni di condivisione, oppure non appena l'utente usa il campo di ricerca utenti e gruppi.

A questo punto l'utente che ha accesso a questo modulo può assegnare permessi semplicemente selezionando le caselle di spunta disponibili.

Come avrete notato, non tutte le spunte sono sempre attive, ma vengono a volte sostituite da icone. Il testo di aiuto in basso è molto utile a comprendere perché alcune spunte possono essere inattive.

I **ruoli globali** () sono quelli discussi alla sezione precedente. Se un dato utente o gruppo ha dei ruoli globali non avrebbe nessun effetto poter assegnare quello stesso ruolo anche nel contesto corrente, quindi l'azione è disabilitata.

I **ruoli ereditati** () verranno discussi meglio tra poco.

Ereditarietà dei ruoli locali

I ruoli assegnati agli utenti in Plone vengono di norma ereditati. Questo permette di fornire ruoli locali ad utenti in una sezione e (ovviamente) avere questi stessi ruoli in tutto il sotto-albero.

Nell'esempio di poco fa, il gruppo “Direzione” all'interno della cartella “Ufficio 5” ha un ruolo ereditato da un qualche livello superiore. Non possiamo sapere da questa pagina da quale livello si ottenga questa ereditarietà; la logica ci dice che molto probabilmente il gruppo ha un ruolo assegnato nella cartella padre (*Uffici*) ma questo non è importante.

Anche in questo caso, come succede per i ruoli globali, il controllo per assegnare il ruolo può essere inaccessibile e sostituito da un'icona, e questo per lo stesso motivo: non avrebbe effetto assegnare lo stesso ruolo ad un utente o un

gruppo che già lo possiede per effetto dell'ereditarietà

C'è però un comportamento molto interessante, che è il motivo scatenante per cui consiglio i ruoli locali a discapito dei ruoli globali: i ruoli locali possono essere bloccati.

Tu sei qui: [Home](#) / [Uffici](#) / Ufficio 5

[Contenuti](#) [Visualizza](#) [Modifica](#) [Regole](#) [Condivisione](#)

Condivisioni di "Ufficio 5"

Puoi controllare chi può visualizzare e modificare l'elemento usando l'elenco che segue.

Nome	Può aggiungere	Può modificare	Può revisionare	Può vedere
Direzione	☐	☐	☐	☐
Utenti collegati	☐	☐	☐	☐

☐ **Eredita i permessi dai livelli superiori**
Di norma, i permessi di questo elemento vengono ereditati dal contenitore. Se disabiliti questa opzione, verranno considerati solo i permessi di condivisione definiti esplicitamente. Nel sommario, il simbolo indica una impostazione ereditata. Analogamente, il simbolo indica un ruolo globale. Questi sono gestiti dall'amministratore del sito nelle impostazioni del portale.

Fig. 4.8: La vista della condivisione di un elemento con blocco dell'ereditarietà dei ruoli

La spunta "Eredita i permessi dai livelli superiori" ha proprio l'effetto descritto: se viene rimossa si viene ad annullare l'ereditarietà dei ruoli *locali* (e non globali) da un qualunque livello superiore.

A questo punto il gruppo "Direzione" diventa un gruppo come gli altri. Potremmo anche ri-assegnare lo stesso potere che aveva prima del blocco dell'ereditarietà e non sarebbe nemmeno un comportamento tanto bizzarro (perché magari era nel nostro interesse che il gruppo non avesse quel ruolo in altri uffici, ma non in questo).

Il blocco dell'ereditarietà permette di creare sezioni protette all'interno di aree del sito:

- una cartella altamente riservata, invisibile e inaccessibile a tutti gli utenti a cui abbiamo dato poter di poter vedere la nostra sezione della intranet
- un documento in sola lettura che nessun utente con potere di modificare possa toccare
- una sezione dove gli amministratori del sito posizionano documentazione relativa ad un gruppo di persone, ma non accessibile al gruppo stesso

Condivisione del documento predefinito

Un errore comune è quello di finire erroneamente nella condivisione di un documento usato come vista predefinita di una cartella e non nella cartella stessa.

Visto che nel 90% dei casi questo è un errore, Plone ci avverte del problema con un messaggio.

Questo comportamento potrebbe anche diventare un'opportunità, probabilmente legata al blocco dei ruoli locali descritti poco fa.

4.2.3 Ruoli locali come fossero globali

Verrà ora descritto come poter avere nel proprio sito Plone lo stesso comportamento relativo ai ruoli globali pur mantenendo la possibilità di bloccare l'ereditarietà.

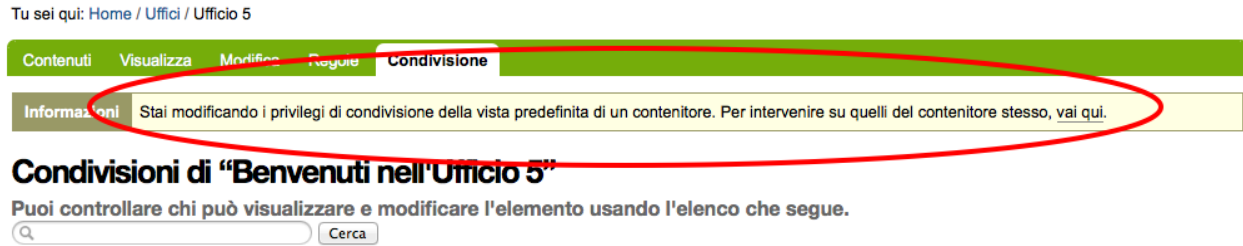


Fig. 4.9: Il messaggio di avvertimento in caso di condivisione dei permessi su una pagina predefinita

Quello che basta fare è usare la condivisione di Plone sulla radice del sito (come descritto alla sezione precedente: fate attenzione a non essere finiti in condivisione della pagina predefinita del sito). In questo modo avete il meglio dei due mondi:

- I ruoli sono assegnati ad utenti o gruppi in tutto il sito
- In qualunque momento potete bloccare l’ereditarietà dei ruoli in specifiche sezioni del sito

4.2.4 La “traduzione” dei ruoli locali

Fin’ora non abbiamo accennato nulla sul fatto che sembra esserci una grande differenza tra che cosa viene visualizzato nella gestione dei ruoli globali e cosa invece nella vista di condivisione per assegnare ruoli locali.

Avrete già notato come nella configurazione del sito vengano mostrati quasi tutti i ruoli che sono stati descritti nella relativa sezione. Sono esclusi tutti i ruoli definiti da Zope tranne *Manager* ma sono inclusi tutti i ruoli definiti a livello Plone. Questa vista ha quindi la particolarità di **mostrare automaticamente i nuovi ruoli** che potreste andare a definire.

Ruoli							Az
Contributore	Editor	Collaboratore	Lettore	Revisore	Amministratore del sito	Manager	da
☐	☐	☒	☐	☐	☐	☐	

Fig. 4.10: I ruoli globali, come sono presentati dalla gestione utenti e gruppi

Lo stesso non succede per la vista di condivisione, dove potrebbe addirittura sembrare che non siano mostrati *ruoli* ma *permessi*.

	Cerca				
	Può aggiungere	Può modificare	Può revisionare	Può vedere	
ti	☐	☐	☐	☐	

Fig. 4.11: I ruoli locali, come sono presentati dalla vista condivisione

In realtà questo non è vero. Sempre per semplificare la vista agli utenti che si avvicinano a Plone per la prima volta e per aumentare l'usabilità della pagina, dalla versione 3 di Plone la condivisione è stata modificata nel seguente modo:

- non mostra tutti i ruoli, ma solo quelli realmente utili per eseguire la condivisione
- non mostra i nomi dei ruoli, ma una generica descrizione di “cosa il ruolo fa”

Quindi:

- “**Può aggiungere**” è per il “*Contributore*”
- “**Può modificare**” è per “*Editor*”
- “**Può revisionare**” è per il “*Revisore*”
- “**Può vedere**” è per il “*Lettore*”

Rimane quindi sempre valida la regola: in Plone si assegnano ruoli, non permessi.

4.2.5 Quando non assegnare inutilmente ruoli

C'è un comportamento scorretto piuttosto comune che ho potuto vedere spesso (che si parli di ruoli globali o locali). Si ha la necessità di dare “*tutti i permessi*” (qualunque cosa significhi, ma è una frase ricorrente) ad un utente in una certa sezione e pigramente si sceglie di “*selezionare tutto*”, ossia assegnare all'utente tutti i permessi possibili... e non pensarci più.

Va invece tenuto presente che alcuni ruoli, per loro natura, “incapsulano” i poteri di altri ruoli. Nella configurazione base di Plone il problema si presenta spesso col ruolo *Lettore*.

È inutile assegnare ad un utente il permesso di *Lettore* se questo utente possiede già uno di altri ruoli quali *Contributore*, *Revisore* o *Editor* poiché questi ruoli per loro natura possiedono già i poteri del *Lettore*.

Ovviamente questo potrebbe non essere vero in presenza di modifiche ai workflow o con workflow particolari.

4.3 Creare nuovi ruoli: quando e perché

Nelle recenti versioni di Plone la necessità di avere nuovi ruoli è venuta largamente meno. Tutti le figure utili per quello che può essere un semplice sito, un enorme portale o una complessa intranet aziendale, sono forniti dall'installazione base di Plone.

4.3.1 Perché creare ruoli aggiuntivi è sconsigliato

La creazione di nuovi ruoli complica i workflow e la gestione dei permessi

Non si sono ancora affrontati i **workflow** o i **permessi** ma anticipiamo qualche cosa. Per ogni ruolo esistente in un sito Plone va considerato il suo effetto per ogni permesso e questo crea una specie di matrice (una tabella). Non c'è bisogno di immaginare questa tabella poiché esiste davvero.

Nell'immagine sopra troviamo in riga i permessi del sito (sono solo una piccola parte e non scenderemo nei dettagli per ora) e in colonna i ruoli. Potete ben immaginare che più la tabella diventa grande, più è difficile da gestire ma non ci è davvero possibile limitare i permessi (o solo in minima parte). Un'installazione base di Plone ha comunque un numero enorme di permessi, quindi dobbiamo rassegnarci ad avere una tabella con tantissime righe.

Capite quindi che aggiungere una colonna a questa tabella aumenta di molto il numero di permessi da gestire per questo ruolo. Nella maggior parte dei casi il valore predefinito del permesso andrà bene, ma particolare attenzione andrà ai permessi che sono poi gestiti tramite workflow... e questo ci obbliga anche a verificare i permessi gestiti dai workflow... per ogni stato.

	Anonymous	Authenticated	Contributor	Editor	Manager	Member	Owner	Reader	Reviewer	Site Administrator
ATContentTypes Topic: Add ATSimpleIntCriterion	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
ATContentTypes Topic: Add ATSimpleStringCriterion	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
ATContentTypes Topic: Add ATSortCriterion	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
ATContentTypes: Add Document	☐	☐	☒	☐	☒	☐	☒	☐	☐	☒
ATContentTypes: Add Event	☐	☐	☒	☐	☒	☐	☒	☐	☐	☒
ATContentTypes: Add File	☐	☐	☒	☐	☒	☐	☒	☐	☐	☒
ATContentTypes: Add Folder	☐	☐	☒	☐	☒	☐	☒	☐	☐	☒
ATContentTypes: Add Image	☐	☐	☒	☐	☒	☐	☒	☐	☐	☒
ATContentTypes: Add Large Plone Folder	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
ATContentTypes: Add Link	☐	☐	☒	☐	☒	☐	☒	☐	☐	☒

Fig. 4.12: Un'anteprima parziale della vista della sicurezza in ZMI

Se tutto questo non sembra ancora abbastanza chiaro, le cose miglioreranno dopo aver letto i rispettivi capitoli sui **permessi** e **workflow**.

Un altro motivo sono i **prodotti aggiuntivi**. È lecito pensare che la vostra installazione Plone utilizzerà alcuni tra le centinaia di add-on disponibili. I prodotti aggiuntivi non conoscono i vostri ruoli e contemporaneamente è possibile che aggiungano al vostro sito nuovi permessi; il prodotto quindi si prenderà in carico di configurare alcune impostazioni di sicurezza al momento dell'installazione.

Quali, se non i ruoli predefiniti, saranno presi in considerazione? Ecco perché molto spesso è meglio cambiare i poteri di un ruolo esistente piuttosto che crearne uno nuovo.

4.3.2 Quando *non* serve un nuovo ruolo

Molto spesso si crede che nel proprio sito Plone serva un nuovo ruolo quando invece serve una modifica ad un qualche workflow o alla sicurezza.

Il problema principale è che **creare nuovi ruoli è facile**, mentre modificare i workflow è una cosa più complessa; alle volte la scelta sbagliata viene presa per pigrizia.

Non è detto serva un nuovo ruolo Plone se serve che un utente debba fare “qualcosa di nuovo”.

Per semplicità seguono tre esempi di casi in cui *non* serve un nuovo ruolo.

Non serve un nuovo ruolo se... 1

Plone ti fornisce il ruolo di *Contributore* e *Editor* ma la tua installazione è semplice e senza fronzoli: i tuoi utenti devono poter creare contenuti e modificare quelli di tutti. Chiamiamolo *Redattore*.

La soluzione: dare entrambi i ruoli ai tuoi utenti.

Non serve un nuovo ruolo se... 2

Hai appena installato [Ploneboard](http://plone.org/products/ploneboard)⁸ e vuoi un nuovo ruolo che ti permetta di gestire i commenti: il *Moderatore*.

La soluzione: il moderatore non sarebbe più o meno il *Revisore* dell'area forum? Perché quindi non usare quel ruolo?

Quello che in questo caso ti serve è una modifica al workflow del forum o dei commenti e l'assegnazione di ruoli locali ai giusti utenti nell'area forum (bloccando ovviamente eventuali altri revisori del sito).

⁸ <http://plone.org/products/ploneboard>

Non serve un nuovo ruolo se... 3

Hai una speciale sezione del sito dove una nuova super razza *Revisori* non solo devono essere in grado di revisionare i contenuti, ma anche di modificarli: il “*Super Revisore*”.

La soluzione: se in quella sezione hai bisogno che tutti i *Revisori* diventino *Super Revisori*, allora quello che ti serve è semplicemente un nuovo workflow, e probabilmente installare il *supporto per le politiche di workflow* (o CMFPlacefulWorkflow⁹, presente nelle installazioni Plone ma di base non attivato).

4.3.3 Quando serve un nuovo ruolo

La creazione di nuovi ruoli è scoraggiata ma è inevitabile in vari casi.

Un ruolo diventa necessario quando un utente deve poter fare qualcosa che nessun altro ruolo (o combinazione di ruoli) sia in grado di fare in quel contesto.

Ricollegiamoci all’ultimo caso appena affrontato (l’ipotesi del *Super Revisore*).

Se in quella speciale area del sito la richiesta fosse stata di mantenere *anche* il ruolo di *Revisore* (col suo funzionamento predefinito, accettare/rifiutare i contenuti), allora il *Super Revisore* (che in più modifica) sarebbe stato per forza un nuovo ruolo.

In presenza di una simile richiesta c’è poco da fare, se non tentare di far ragionare il committente, chiedergli se *davvero* c’è la necessità di una simile presenza di due diverse figure di revisori.

Seguono tre esempi di casi in cui la creazione di un nuovo ruolo è inevitabile. Sono tutti e tre casi reali (decontestualizzati) che ho potuto vedere in questi anni.

Serve un nuovo ruolo se... 1

Hai necessità di un meccanismo di revisione a due livelli: il normale *Revisore* approva i contenuti ma una seconda figura ha voce in capitolo per un’approvazione di secondo livello.

Serve un nuovo ruolo se... 2

Ti viene chiesto che un certo gruppo di utenti debba poter gestire le portlet (riquadri) del sito.

Le portlet sono un’attività ad oggi sotto il controllo dei *Manager* e degli *Amministratori del Sito* e, a meno che la richiesta non sia di dare questo potere a tutti i *Revisori* del sito, l’unica soluzione è creare un ruolo di *Gestore portlet*.

Serve un nuovo ruolo se... 3

La tua installazione Plone è in realtà un applicativo di gestione ordini (diciamo un DMS), dove in poche cartelle sono contenute decine di migliaia di ordini di acquisto. In più l’azienda che utilizza l’applicativo ha un enorme numero di ruoli interni e tutti devono mettere voce nell’approvazione dell’ordine per passare dalla sua fase iniziale all’evasione finale.

In questo caso siamo in presenza di una struttura del sito molto semplice ma anche di un organigramma molto complesso. L’unica soluzione è davvero quella di creare tutti i ruoli necessari.

⁹ <http://pypi.python.org/pypi/Products.CMFPlacefulWorkflow>

4.4 Come creare nuovi ruoli

Warning: Quando segue, sebbene sia una descrizione di come creare nuovi ruoli tramite ZMI, non è solitamente il comportamento corretto da tenere, soprattutto se si ha accesso al server dove è installato Plone e si ha possibilità quindi di poter aggiungere prodotti.

Per replicare la creazione di un nuovo ruolo con un prodotto, vedere la sezione *Portare quanto fatto in un prodotto*.

La creazione di nuovi ruoli è semplice, basta accedere alla ZMI del proprio sito Plone alla gestione della sicurezza (scheda **security**) il che ci porta alla pagina `/manage_access`.



Fig. 4.13: Link per andare alla gestione della sicurezza del sito Plone, da ZMI

In fondo a questa pagina trovare il **form per aggiungere nuovi ruoli** (ed eliminarli).

You can define new roles by entering a role name and clicking the "Add Role" button.

User defined roles

Super Revisore Add Role

Contributor Delete Role

Fig. 4.14: Form per la creazione di nuovi ruoli dalla gestione della sicurezza del sito Plone, da ZMI

Basta scegliere un nome per il nostro ruolo e premere *Add Role*. Immediatamente possiamo vederne gli effetti nella pagina stessa.

Roles											
Anonymous	Authenticated	Contributor	Editor	Manager	Member	Owner	Reader	Reviewer	Site Administrator	Super Revisore	
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Fig. 4.15: Il nuovo ruolo appena creato, inserito tra le colonne della matrice

4.4.1 Comportamenti di un ruolo appena creato

Quando viene creato un nuovo ruolo, Plone non sa come gestirlo, quindi non gli viene assegnato nessun permesso (la colonna dei checkbox sarà inizialmente vuota). Sta a voi passare in rassegna tutti i permessi e fornire al ruolo tutti i permessi necessari, tenendo presente che:

- alcuni permessi potrebbero essere gestiti dai workflow

- gli utenti potrebbero avere altri ruoli oltre al nuovo venuto

Date al ruolo il minor numero di permessi possibile, concentratevi su quello che il ruolo dovrà fare.

Manteniamoci sull'esempio proposto in precedenza: il *Super Revisore* deve avere gli stessi poteri del *Revisore* ma poter modificare i contenuti pubblicati. Il primo passo sarà quello di copiare *tutti* i permessi associati al ruolo di *Revisore*, poi concentrarsi sulle differenze. Per poter modificare i documenti pubblicati sarà necessario lavorare col workflow.

Non assegnate altri permessi se non sono necessari.

4.4.2 Assegnare il nuovo ruolo

Lasciamo la ZMI e torniamo all'interfaccia di Plone.

Come fatto in precedenza, partiamo dalla gestione utenti e gruppi del sito.

Utenti **Gruppi** Impostazioni Registrazione utenti

Elenco generale dei gruppi

[Torna alle impostazioni di Plone](#)

I gruppi sono insiemi logici di utenti che condividono una certa caratteristica, come un dipartimento, o un ufficio. Non sono direttamente legati ai permessi, in generale: normalmente per quello scopo si usano i ruoli, assegnandoli poi ai vari gruppi. Il simbolo indica un ruolo che eredita i permessi da un gruppo

Aggiungi un nuovo gruppo

Ricerca gruppi Cerca Mostrali tutti

Nome del gruppo	Ruoli								Rimuovi gruppo
	Contributore	Editor	Collaboratore	Lettore	Revisore	Amministratore del sito	Super Revisore	Manager	
Administrators (Administrators)	☐	☐	☐	☐	☐	☐	☐	☒	☐
Authenticated Users (Virtual Group) (AuthenticatedUsers)	☐	☐	☐	☐	☐	☐	☐	☐	☐
Direzione (Direzione)	☐	☐	☐	☐	☐	☐	☐	☐	☐
Reviewers (Reviewers)	☐	☐	☐	☐	☒	☐	☐	☐	☐
Site Administrators (Site Administrators)	☐	☐	☐	☐	☐	☒	☐	☐	☐

Fig. 4.16: Il nuovo ruolo appena creato, visto dalla gestione utenti e gruppi di Plone

In questo caso possiamo vedere la facilità con cui saremmo in grado di assegnare questo ruolo in modo globale. Come già discusso, questo può essere giusto o sbagliato; magari nel vostro sito il gruppo **Direzione** deve possedere questo ruolo in modo globale e senza eccezioni e sarebbe quindi giusto fornirgli questo ruolo da questa pagina.

L'esempio con cui abbiamo introdotto il concetto di *Super Revisore* parlava però di una specifica cartella del sito dove questi utenti dovevano poter lavorare (ammettiamo che questo avvenga nella cartella *News*). Per ottenere questo avrete capito che si sta invece parlando di ruoli locali, quindi andiamo a condividere il nuovo ruolo su quella cartella.

Che succede? Avremmo bisogno di vedere il nuovo ruolo nella condivisione ma questo non compare!

Avevamo detto in precedenza come la condivisione non mostra tutti i ruoli, ma solo quelli realmente utili al funzionamento della convisidione di Plone. Il problema ora è che il nostro ruolo *dovrebbe* comparire in questa lista e siamo

Tu sei qui: [Home](#) / [Notizie](#)

Contenuti
Visualizza
Modifica
Regole
Condivisione

Condivisioni di “Notizie”

Puoi controllare chi può visualizzare e modificare l'elemento usando l'elenco che segue.

Nome	Può aggiungere	Può modificare	Può revisionare	Può vedere
Utenti collegati	☐	☐	☐	☐

Fig. 4.17: Dalla condivisione della cartella “News” manca il nuovo ruolo appena definito

quindi costretti a dire esplicitamente a Plone di voler inserire il nuovo ruolo.

Per fare questo purtroppo dobbiamo per la prima volta sporcarci davvero le mani con qualche riga di codice.

Note: Ad oggi non c'è nessun modo di scegliere da interfaccia Plone o ZMI quali ruoli mostrare nella pagina della condivisione.

Questo viene ottenuto in modo piuttosto semplice, utilizzando uno dei passi di un profilo di *Generic Setup* (che si occupa di gestire per noi una *local utility*).

4.4.3 Mostrare i nuovi ruoli nella condivisione

La prima domanda che probabilmente ci si porrà: dove inserire il codice?

Il codice di Plone, quando non fa parte del core, va in prodotti aggiuntivi. Non è necessario siano uno dei prodotti liberamente scaricabili dal sito ufficiale¹⁰, ma può essere un vostro prodotto ad uso interno.

In quale prodotto inserire questo codice?

In questa guida non si vuole affrontare nel dettaglio come creare nuovi prodotti in Plone ma una risposta va comunque quantomeno accennata.

Il vostro nuovo ruolo è parte integrante di un prodotto che aggiunge una qualche funzionalità a Plone? In quel caso la risposta è semplice: il prodotto stesso dovrebbe fornirvi anche il nuovo ruolo.

Se il vostro ruolo è necessario per il funzionamento di un workflow personalizzato è molto probabile che anche il workflow diventerà prima o poi parte di un prodotto, quindi tanto vale inserire in un ipotetico prodotto chiamato “miaazienda.workflow” la registrazione del permesso.

Questo stesso prodotto potrebbe poi essere necessario per ospitare anche le modifiche ai permessi del sito.

Ammettiamo che l'azienda per cui stiamo sviluppando questo workflow si chiami **Lorem Ipsum S.r.L.** Ecco quindi che il buon nome per questo prodotto potrebbe diventare **loremipsum.workflow** (la convenzione dei nomi di prodotti Zope&Plone di solito rispecchia il namespace Python della libreria) e la sua realizzazione ci accompagnerà nei prossimi capitoli.

¹⁰ <http://plone.org/products/>

La registrazione del ruolo

La prima cosa che ci serve è avere un prodotto che registri un profilo di *Generic Setup*, in pratica la definizione di una serie di passi che vengano eseguiti quando il prodotto verrà poi attivato.

Il profilo principale è di solito contenuto nella cartella `profiles/default` di un qualunque prodotto. Per maggiori informazioni riferirsi alla sezione “*Portare quanto fatto in un prodotto*”.

La definizione del ruolo locale va fatta utilizzando un file `sharing.xml`¹¹ così composto ([sorgente online](#)¹²):

```
<?xml version="1.0"?>
<sharing xmlns:il8n="http://xml.zope.org/namespaces/il8n">
  <role
    id="Super Revisore"
    title="Può super-revisionare"
    permission="Sharing page: Delegate roles"
  />
</sharing>
```

Sebbene possiamo ignorare come il codice sopra venga eseguito, capirlo risulterà piuttosto semplice: l’`id` non è altro che il nome del ruolo, l’attributo `title` rappresenta come questo ruolo deve essere tradotto nell’interfaccia di condivisione. Come ricorderete, la pagina di condivisione “traduce” i ruoli in azioni; ammetto che il nome scelto “*Può super revisionare*” sia assolutamente poco sensato, ma ci farà capire al volo a quale ruolo ci stiamo riferendo.

L’ultimo attributo (`permissions`) verrà spiegato tra poco.

Ora torniamo all’interfaccia Plone.

Tu sei qui: [Home](#) / [Notizie](#)

The screenshot shows the 'Condivisione' (Sharing) tab selected in the Plone interface. Below the tab, the title 'Condivisioni di "Notizie"' is displayed. A message states: 'Puoi controllare chi può visualizzare e modificare l'elemento usando l'elenco che segue.' Below this is a search bar with the text 'Cerca'. A table lists permissions for 'Utenti collegati' (Connected users). The table has six columns: 'Nome', 'Può aggiungere', 'Può modificare', 'Può revisionare', 'Può super-revisionare', and 'Può vedere'. The 'Può super-revisionare' column is circled in red. The 'Utenti collegati' row shows checkboxes for each permission, all of which are currently unchecked.

Nome	Può aggiungere	Può modificare	Può revisionare	Può super-revisionare	Può vedere
Utenti collegati	☐	☐	☐	☐	☐

Fig. 4.18: Dalla condivisione della cartella “News” finalmente si vede il nuovo ruolo

Sembra che il nostro scopo sia raggiunto! Assegnando il nuovo ruolo ad utenti e gruppi sulla cartella possiamo finalmente imbastire dei *Super Revisori* locali.

4.5 Chi può assegnare i ruoli, e quali

In questa sezione analizzeremo un altro aspetto spesso tralasciato della configurazione della sicurezza: limitare (o per lo meno: conoscere) chi può fornire nuovi ruoli... e quali.

¹¹ <http://plone.org/documentation/manual/developer-manual/generic-setup/reference/sharing-ref>

¹² <https://github.com/keul/loremipsum.workflow/blob/de8abb19c0006246c881a9a397510ffa67fa0db5/loremipsum/workflow/profiles/default/sharing.xml>

4.5.1 L'accesso alla pagina di condivisione

In una installazione base di Plone la pagina di condivisione non è visibile a tutti gli utenti:

- Il *Lettore* non accede alla vista
- Il *Contributore* non accede alla vista
- Il *Possessore* accede alla condivisione e può dare ad altri i ruoli di *Contributore*, *Editor* e *Lettore*
- Il *Contributore* non ha accesso alla scheda, se non sui contenuti da lui creati (poiché vale la regola del *Possessore* sopra descritta)
- L'*Editor* accede alla condivisione e può fornire il ruolo di *Editor* e *Lettore*
- Il *Revisore* accede alla condivisione e può fornire il ruolo di *Revisore* e *Lettore*
- *Manager* e *Amministratore del sito* accedono, ed ovviamente possono fornire tutti i ruoli

Leggendo l'elenco qui sopra potreste avere molte obiezioni ma è probabilmente impossibile in questo caso trovare una configurazione di base che vada bene a tutti.

Una delle cose contro cui mi sono trovato più spesso a modificare è il comportamento del *Possessore*: solo perché un utente ha creato un documento, magari in una sezione privata, gli dà il diritto di fornire l'accesso ad altri utenti?

Altro esempio: è giusto che ogni utente con un determinato ruolo che ha accesso alla vista possa fornire il ruolo stesso? Magari un amministratore ha fornito all'utente A il ruolo di *Revisore*, e questi "subappalta" lo stesso ruolo ad altre persone!

Per fortuna questa è solo una configurazione di base che può essere modificata.

L'accesso alla pagina è controllato da un singolo permesso, ma il poter assegnare o meno un determinato ruolo è controllato da uno specifico **permesso di delega**, diverso per ogni ruolo.

Questi permessi verranno analizzati e spiegati nell'*apposita sezione*.

4.5.2 Comportamento del Super Revisore nella pagina di condivisione

Non abbiamo ancora parlato di chi possa fornire il ruolo di *Super Revisore* ma capirete facilmente che questo dipende dall'attributo `permissions` che poco sopra abbiamo ignorato.

Non essendo ancora in possesso di un permesso specifico per delegare il nostro ruolo, ci siamo limitati ad importare il permesso più generale ossia "*Sharing page: Delegate roles*", che è lo stesso permesso che protegge l'accesso generale alla pagina di condivisione.

In pratica stiamo dicendo che chiunque acceda alla condivisione può delegare questo ruolo.

Non è assolutamente una soluzione ottimale, quindi in vedremo come sistemare questo problema.

4.6 Verificare i ruoli di un utente

Quando si disegnano complessi workflow o si ricevono segnalazioni dagli utenti del tipo "*non riesco ad accedere alla sezione*" oppure "*pare io non possa fare quest'operazione, perché?*" vi troverete nella situazione di dover capire che cosa non funziona.

Recentemente Plone ha introdotto un utilissimo form che permette di verificare quali ruoli (e permessi) un utente possieda in un certo contesto.

Il form è accessibile dalla scheda *Security* della ZMI.

When a role is assigned to a permission, users with the given role will be able to perform tasks associated with the permission on this item. When the *Acquire permission settings* checkbox is selected then the containing objects's permission settings are used. Note: the acquired permission settings may be augmented by selecting Roles for a permission in addition to selecting to acquire permissions.

Username: [Show me the user permissions and roles in the context of the current object](#)

Fig. 4.19: Il form che permette di visualizzare ruoli e permessi nel contesto

Ora ci concentriamo sui ruoli, ma come potrete vedere questo utilissimo form mostra anche i permessi. Per usarlo basta inserire lo *userid* di un utente esistente ed otterremo qualcosa del genere:

Contents

Components

View

Properties

Security

Undo

Plone Site at /Plone

This listing shows the permissions and roles for particular user in the context of the current object.

User account : utente2
User account defined in: /book/acl_users

Roles	Roles in context
• Authenticated • Member	• Authenticated • Member

Fig. 4.20: Il risultato della ricerca di ruoli e permessi dell'utente

Da notare come vengano mostrati i “ruoli” (**Roles**) ossia i ruoli globali e i ruoli nel contesto specifico (**Roles in context**).

La prima osservazione dopo aver visto questo form potrebbe essere relativa alla sua effettiva utilità, in quanto da ZMI la pagina “*Security*” è visibile solo nella radice del sito, mentre invece sarebbe estremamente utile avere questo strumento sul singolo contenuto o su una cartella del nostro sito.

C'è un trucco. In pratica la pagina *Security* è accessibile ovunque, su qualunque contenuto ma è stata di recente nascosta nelle recenti versioni di Plone; è però ancora richiamabile manualmente così:

http://urldelsito/percorso/al/contesto/manage_access

Warning: Modificare le impostazioni di sicurezza via ZMI in sezioni che non siano la radice del sito Plone può portare a problemi difficili da capire.
Le modifiche qui fatte potrebbero poi essere sovrascritte.

Questo è anche messo in evidenza da un ben visibile avvertimento ad inizio pagina.

ATFolder at /Plone/news

Attention!

Any security settings for Plone objects changed here are liable to be overwritten without warning. To assign local roles use the "Sharing" tab in Plone. More complex changes should be made using workflows where appropriate.

Fig. 4.21: La pagina di gestione ruoli e permessi in sezioni diverse dalla radice del sito Plone

Quindi: limitatevi all'uso del form per trovare ruoli e permessi degli utenti a meno che non sappiate davvero quello che fate!

4.7 Portare quanto fatto in un prodotto

Anche se è possibile effettuare alcune modifiche al proprio sito via ZMI, questo non vuole assolutamente dire che sia giusto farlo.

Note: È sconsigliato agire via ZMI poiché diventa problematico rendere *replicabili* le operazioni svolte.

Se le vostre configurazioni fossero da replicare in un sito gemello di quello che state impostando, o se voleste rendere disponibile ad altri il vostro lavoro, obblighereste queste persone a ripetere manualmente i passi da voi eseguiti. Alle volte il problema diventa anche ricordarsi tutto quello che è stato fatto.

I puristi dicono che tutto deve essere fatto tramite installazione di prodotti o **esecuzione di profili di Generic Setup**. Lasciatemi dire che i puristi questa volta hanno ragione. Ciononostante è molto più facile imparare a configurare Plone via Web, poi sistemare le cose non appena la nostra modifica diventa definitiva.

4.7.1 Creare automaticamente il nuovo ruolo

L'unica configurazione non ancora coperta da codice che abbiamo eseguito è la creazione del nuovo ruolo *Super Revisore* fatta da ZMI.

Il nostro *loremipsum.workflow* ha bisogno di avere un profilo di installazione e diventare quindi un prodotto Plone installabile dal pannello di gestione dei prodotti aggiuntivi.



Fig. 4.22: Col profilo definito il nostro prodotto è installabile

Per fare questo è necessario che il prodotto registri un profilo di **Generic Setup**, il modo in cui i prodotti Plone eseguono operazioni standard usando file in formato XML.

La registrazione del profilo avviene tramite una modifica al file `configure.zcml` già visto in precedenza ([sorgente online](#)¹³):

```
...
<genericsetup:registerProfile
  name="default"
  title="loremipsum.workflow"
  directory="profiles/default"
  description="Installazione del workflow della Lorem Ipsum S.r.L."
  provides="Products.GenericSetup.interfaces.EXTENSION"
/>
...
```

A questo punto Zope si aspetta di trovare un folder `profiles` (che ospiterà tutti i profili del prodotto, se più di uno) con all'interno un folder `default`. Nella convenzione, "default" è usato per il profilo predefinito.

¹³ <https://github.com/keul/loremipsum.workflow/blob/0a6ba2d069eae8174d6ace5b5c48657be3f74246/loremipsum/workflow/configure.zcml>

Per registrare il nostro ruolo il profilo deve contenere un file `rolemap.xml` così fatto ([sorgente online](#)¹⁴):

```
<?xml version="1.0"?>
<rolemap>
  <roles>
    <role name="Super Revisore"/>
  </roles>
  <permissions>

  </permissions>
</rolemap>
```

Notate come la sezione `roles` serva a definire nuovi ruoli, mentre la sezione `permissions` sia riservata per creare permessi (anche se vuota, deve essere presente).

A questo punto se il nostro prodotto venisse installato in un sito dove il ruolo *Super Revisore* non fosse presente, questo verrebbe creato.

¹⁴ <https://github.com/keul/loremipsum.workflow/blob/0a6ba2d069eae8174d6ace5b5c48657be3f74246/loremipsum/workflow/profiles/default/rolemap.xml>

I Permessi

Al capitolo precedente è stata data la definizione di ruolo e si è visto come questo sia associato direttamente con i **permessi**. Si è anzi detto come *il ruolo non sia altro che un raggruppamento di permessi*.

Se un utente con ruolo di *Editor* può fare alcune delle cose possibili anche al *Revisore* (accedere alla vista dei contenuti di una cartella, modificare un documento, ...) è dovuto al fatto che condividono uno o più permessi.

I permessi sono il vero cuore della sicurezza di Plone, poiché controllano una singola azione o un comportamento puntuale del CMS.

È bene chiarire che agiscono a basso livello; fin'ora ci siamo abituati a lavorare sull'interfaccia di Plone, per poi muoverci brevemente a livelli più bassi (in ZMI) e abbiamo visto poco codice. I permessi invece *non sono visibili o gestiti a livello Plone* (per questo motivo non sono nemmeno tradotti).

In questo capitolo non scenderemo ad un livello di dettaglio eccessivo poiché non risulterebbe utile, a meno che la vostra intenzione non sia diventare uno sviluppatore di prodotti Plone (il che esula dallo scopo di questo libro).

Vi basti sapere che la singola chiamata ad un metodo di una classe Python potrebbe essere protetta da un permesso. Questo significa che quando quel metodo viene chiamato per reagire ad un'azione di un utente, viene verificato se l'utente possiede il permesso richiesto. In caso contrario viene lanciata un'eccezione speciale: **Unauthorized (Non autorizzato)** che, di solito, genera la classica pagina di permessi insufficienti di Plone.

Tu sei qui: [Home](#)

Permessi insufficienti

Non disponi dei privilegi sufficienti per visualizzare questa pagina. Se ritieni che questo sia sbagliato, contatta l'[amministratore del sito](#).

Fig. 5.1: La classica pagina di errore di Plone per permessi insufficienti

Ma i permessi non si limitano solo a generare errori di mancanza di privilegi: alcuni comportamenti del CMS controllabili da ZMI (come ad esempio l'accesso a vari aspetti dell'interfaccia grafica di Plone) sono regolati da permessi: avere il permesso richiesto determina se l'elemento grafico compaia o meno.

5.1 La gestione dei permessi

La gestione dei permessi avviene da ZMI, dalla radice del sito, dalla stessa pagina da cui abbiamo creato in precedenza un nuovo ruolo: la scheda **Security**.

Un accesso diretto alla pagina (che permette anche di non aprirla nel solito frame HTML usato dalla ZMI) è ottenibile richiamando manualmente `/manage_access` sul contesto del sito Plone.

Ad esempio: se state facendo test su un sito locale dovreste probabilmente digitare:



Fig. 5.2: Link per andare alla gestione della sicurezza del sito Plone, da ZMI

http://localhost:8080/Plone/manage_access

Quello che vi troverete davanti è una griglia la cui logica è riassunta nello schema seguente.

Permission		Roles		
		Ruolo 1	Ruolo 2	Ruolo 3
Acquire permission settings?				
☒	Permesso A	☐	☐	☐
☒	Permesso B	☐	☐	☐
☒	Permesso C	☐	☐	☐

Fig. 5.3: Schema generale della gestione della sicurezza del sito

In **riga** avrete disponibili i permessi, e come potete vedere in un sito Plone possono essere davvero molti.

In **colonna** ci sono i ruoli, tutti i ruoli definiti, che siano Zope, Plone o applicativi (anche il nostro *Super Revisore* si trova qui).

Preso come riferimento un qualunque permesso e un qualunque ruolo, trovate all'incrocio della riga e della colonna un checkbox:

- se il checkbox è *selezionato* quel ruolo ha il relativo permesso nel contesto (il sito Plone).
- Se il checkbox è *deselezionato* quel ruolo non ha il permesso.

5.1.1 Capire “Acquire permission settings?”

Avrete notato la presenza di una serie di checkbox in prima colonna con intestazione “Acquire permission settings?”.

Il loro significato è estremamente importante e diventerà vitale per la realizzazione di buoni workflow.

Noterete infatti come per tantissimi permessi non ci sia nessuno dei checkbox della griglia selezionati ma solo quello dell'acquisizione (che è invece quasi sempre selezionato in ogni permesso).

Il suo significato è “*acquisisci permessi dal livello superiore/dal contenitore*”.

Ci si potrebbe chiedere quale possa essere il “contenitore” del sito Plone e la risposta è: la **radice di Zope**. Anche questa infatti è una specie di cartella, dove i siti Plone diventano dei semplici contenuti e da dove è possibile ancora una volta accedere alla scheda “Security”.

Per accedere alla radice di Zope è necessario avere un utente con i poteri di *Manager* sull'intera installazione (di solito: l'unico disponibile è l'utente predefinito *admin*). Mantenendo l'esempio precedente, l'URL di accesso del vostro sito di test dovrebbe quindi essere:

http://localhost:8080/manage_access

Innanzitutto va notato come da questa pagina non sia presente un'ulteriore serie di checkbox per l'acquisizione dei permessi (siamo giunti davvero alla radice).

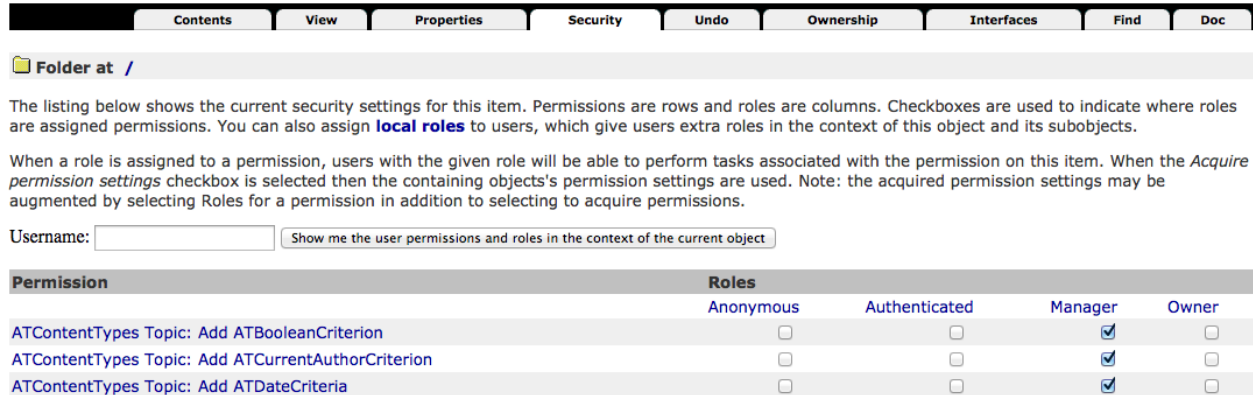


Fig. 5.4: Come si presenta la gestione della sicurezza sulla radice di Zope

I permessi che troverete sono gli stessi del sito Plone, l'unica differenza sta nei *ruoli*: qui troverete solo i ruoli predefiniti di Zope e non quelli Plone (quindi nemmeno il nostro *Super Revisore*).

A parità di permesso, le impostazioni di sicurezza definite qui si vanno a sommare a quelle definite nello stesso permesso del sito Plone *se* il checkbox “Acquire” è selezionato.

Se l'acquisizione del permesso nel sito Plone è deselezionata, le impostazioni al livello superiore vengono ignorate.

5.1.2 Modifiche ai permessi al di fuori della radice del sito

La modifica dei permessi sulla radice del sito è normale amministrazione del lavoro con Plone per personalizzare per i propri bisogni la sicurezza.

La modifica dei permessi nella radice di Zope è meno comune ma comunque possibile e tutto sommato lecita (consiglio comunque di evitarla, ed accedervi solo in consultazione).

Nella sezione “*Verificare i ruoli di un utente*” abbiamo visto come la pagina di modifica della security sia accessibile anche al di fuori della radice del sito (anche se nascosta).

L'avvertimento dato in precedenza è talmente importante che vale la pena ripeterlo:

Warning: Modificare le impostazioni di sicurezza via ZMI in sezioni che non siano la radice del sito Plone può portare a problemi difficili da capire.

5.2 Il funzionamento dei permessi nei contenuti

Pur tuttavia il cuore della sicurezza in Plone sta tutto qui: per sapere se un utente ha il potere di fare una certa azione in un dato contesto, viene verificato se è in possesso di uno specifico permesso e nella maggior parte dei casi questo permesso è **controllato sul contesto stesso**.

Vediamo ad esempio cosa succede se accediamo alla gestione della sicurezza di un contenuto news in stato *privato*.

Noterete come ci siano varie impostazioni personalizzate e non solo una serie infinita di “Acquire”.

Per rendere le cose semplici ci concentreremo solo su un permesso: *View*, ossia il permesso che determina se il contenuto può essere visto o meno (verrà trattato nel dettaglio in seguito).

Qualcosa ha determinato che quel contenuto (la news) sia visibile (e quindi accessibile) solo dai ruoli *Contributore*, *Editor*, *Manager*, *Possessore*, *Lettore* e *Amministratore del sito*.

HistoryViewInterfacesDoc

ATNewsItem at /Plone/news/news-personale

Attention!

Any security settings for Plone objects changed here are liable to be overwritten without warning. To assign local roles use the "Sharing" tab in Plone. More complex changes should be made using workflows where appropriate.

The listing below shows the current security settings for this item. Permissions are rows and roles are columns. Checkboxes are used to indicate where roles are assigned permissions. You can also assign **local roles** to users, which give users extra roles in the context of this object and its subobjects.

When a role is assigned to a permission, users with the given role will be able to perform tasks associated with the permission on this item. When the *Acquire permission settings* checkbox is selected then the containing objects's permission settings are used. Note: the acquired permission settings may be augmented by selecting Roles for a permission in addition to selecting to acquire permissions.

Username: Show me the user permissions and roles in the context of the current object

Permission	Roles
Acquire permission settings?	Anonymous Authenticated Contributor Editor Manager Member Owner Reader Reviewer Site Administrator Super Revisore
☒ ATContentTypes: View history	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☐ Access contents information	☐ ☐ ☒ ☒ ☒ ☐ ☒ ☒ ☐ ☐ ☒ ☐
☒ Change permissions	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☒ Copy or Move	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☒ Delete objects	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☒ FTP access	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☒ Manage WebDAV Locks	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☒ Manage portal	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☒ Manage properties	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☐ Modify portal content	☐ ☐ ☐ ☒ ☒ ☐ ☒ ☐ ☐ ☐ ☒ ☐
Acquire?	Anonymous Authenticated Contributor Editor Manager Member Owner Reader Reviewer Site Administrator Super Revisore
☒ Modify view template	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☒ Take ownership	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☒ Undo changes	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☐ View	☐ ☐ ☒ ☒ ☒ ☐ ☒ ☒ ☐ ☐ ☒ ☐
☒ View History	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☒ View management screens	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☒ WebDAV Lock items	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☒ WebDAV Unlock items	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
☒ WebDAV access	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Save Changes

You can define new roles by entering a role name and clicking the "Add Role" button.

User defined roles

Add Role

Fig. 5.5: Come sono impostati i permessi di una news privata

Per questo motivo chiunque sia sprovvisto di questi ruoli nel contesto della news, non potrà accedervi (ed otterrà l'errore permessi insufficienti).

Chi però governa questi permessi sulla news è il **workflow ad essa associato**.

5.2.1 L'importanza del contesto

Il concetto di **contesto** è vitale per comprendere appieno i permessi o per realizzare buoni workflow.

Potenzialmente tutti i permessi possono essere verificati sul *contesto corrente* (che identifica sempre il documento che l'utente sta visitando, o la radice del sito Plone nel caso si sia posizionati proprio su quest'ultima) ma alcuni di questi sono nei fatti verificati solo sulla radice del sito (questo dipende dallo scopo del permesso).

5.3 Analisi dei permessi esistenti

Se fin'ora vi siete spaventati di fronte alla grande quantità di permessi che Plone offre e alla mancanza di una descrizione dettagliata sul loro significato, sappiate che le cose non stanno così male.

Molti dei permessi che vedete sono definiti dagli strati software più bassi (CMF, Zope, ...) e **non serve gestirli in Plone** o tanto meno comprenderne il significato. Per questi permessi potete lasciare il valore predefinito e dimenticarvi di loro (e così faremo qui).

Rimane però vera la seconda osservazione: non ci sono descrizioni del funzionamento dei permessi ma per alcuni è importante sapere a cosa servono.

Di seguito analizzeremo una piccola serie di permessi che sono davvero molto importanti per il funzionamento di Plone e che necessitano di essere compresi.

Se state cercando una **lista completa dei permessi utilizzati da Plone** potete trovarla andando all'[Appendice A](#).

5.3.1 ATContentTypes: Add tipo di contenuto

Questa serie di permessi controlla il potere di **poter aggiungere un tipo di contenuto** e ne esiste uno per ognuno dei tipi base di Plone.

Il prefisso *ATContentTypes* identifica uno dei prodotti Plone centrali che è per l'appunto [Products.ATContentTypes](#)¹⁵. Questo prodotto è quello che fornisce attualmente i tipi base di Plone basati sul framework [Archetypes](#)¹⁶. Nelle prossime versioni di Plone il framework di riferimento cambierà, sostituito da [Dexterity](#)¹⁷ (e quindi dal prodotto [plone.app.contenttypes](#)¹⁸ di cui al momento non esiste una release stabile).

Segue uno ad uno l'elenco dei permessi e una brevissima spiegazione.

ATContentTypes: Add Document Aggiunta di una **Pagina** (*Document* è il vecchio nome della *Pagina* ma era considerato troppo generico e per questo cambiato).

ATContentTypes: Add Event Aggiunta di un **Evento**.

ATContentTypes: Add File Aggiunta di un **File**.

ATContentTypes: Add Folder Aggiunta di una **Cartella**.

ATContentTypes: Add Image Aggiunta di un' **Immagine**.

¹⁵ <http://pypi.python.org/pypi/Products.ATContentTypes>

¹⁶ <http://pypi.python.org/pypi/Products.Archetypes>

¹⁷ <http://plone.org/products/dexterity>

¹⁸ <https://github.com/plone/plone.app.contenttypes>

ATContentTypes: Add Large Plone Folder Aggiunta di una **Cartella capiente**.

Questo vecchio tipo di contenuto esisteva fino a Plone 4 escluso, dove c'era una differenza tra le cartelle semplici (e ordinabili) e quelle capienti che potevano contenere migliaia di oggetti senza problemi alle prestazioni (ma non ordinabili).

Con Plone 4 esiste [solo un tipo di cartella](#)¹⁹ con tutti i pregi e nessuno dei difetti dei precedenti due tipi.

ATContentTypes: Add Link Aggiunta di un **Collegamento**.

ATContentTypes: Add News Item Aggiunta di una **News**

Noterete come da questa lista sia assente la *Collezione*, poiché per ragioni storiche la sua aggiungibilità è gestita da altri permessi (vedere “[plone.app.collection: Add Collection](#)”).

Manipolare questi permessi si traduce letteralmente nel far sparire o apparire dal menù per l'aggiunta di nuovi elementi il tipo relativo. La differenza con la voce “*Restrizioni...*” dello stesso menù è sostanziale, poiché quella limitazione viene fatta per singola cartella.

Per impostazione predefinita i seguenti ruoli posseggono questi permessi:

- *Manager*
- *Amministratore del sito*
- *Possessore*
- *Contributore*

Note: Il fatto che in questa lista compaia il *Possessore* ci dice una cosa importante (e che molto spesso vale la pena modificare). Un utente che sia proprietario di una cartella (di solito: perché è stato lui a crearla) avrà il potere di inserirvi all'interno tutti i contenuti che vuole.

Vedere anche “[Add portal content](#)”.

5.3.2 Access contents information

Questo permesso è tanto difficile da spiegare quanto importante, letteralmente tradotto in “*accedere alle informazioni dei contenuti*”.

Il suo uso è sparso qua è là nel codice Plone senza che sia esattamente chiarito il suo scopo. Nella pratica è un permesso che solitamente viaggia a stretto contatto col più famoso permesso “*View*” e di solito viene assegnato e negato agli stessi ruoli negli stessi contesti.

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Possessore*
- *Contributore*
- *Lettore*
- *Editor*

¹⁹ <http://pypi.python.org/pypi/plone.app.folder>

5.3.3 Access inactive portal content

Questo permesso è quello che controlla il comportamento delle **date di scadenza e di pubblicazione dei contenuti**.

La sua impostazione modifica le ricerche di Plone e l'accesso alle viste dei contenuti delle cartelle.

Capire il suo funzionamento è molto importante poiché molti utenti credono che la scadenza di un contenuto abbia a che fare con il permesso di accedervi.

Fortunatamente ho già affrontato l'argomento in passato in un articolo piuttosto dettagliato (ed ancora valido): “[Data di Scadenza/Pubblicazione in Plone: la guida definitiva](#)²⁰”. La lezione più importante dell'articolo è la seguente: questo permesso può essere solo usato sulla radice del sito Plone (non può quindi funzionare o essere utilizzato nei workflow).

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Possessore* (il che, come spiegato nell'articolo sopra citato, non vuol dire nulla)

5.3.4 Add portal content

Note: È il permesso di riferimento del ruolo **Contributore**

Storicamente questo permesso era *il* permesso per aggiungere contenuti nel sito. Prima di Plone 2.1 esisteva solo questo permesso per controllare l'aggiungibilità dei contenuti, e controllava *tutti* i contenuti.

I limiti di un simile approccio si sono rivelati molto presto e si è poi arrivati ad avere un permesso per l'aggiunta di ogni contenuto, come descritto nella sezione “*ATContentTypes: Add tipo di contenuto*”.

Il permesso però rimane importante ancora oggi perché dovrebbe determinare il potere di “*poter aggiungere contenuti*” senza specificare quali. In passato non avere questo permesso determinava infatti l'impossibilità di poter aggiungere contenuti, ma questa caratteristica pare essere sparita in una qualche versione di Plone.

Ad ogni modo: il permesso è ancora usato per varie verifiche di sicurezza nel codice Plone quindi non va ignorato completamente.

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Possessore*
- *Contributore*

5.3.5 CMFEditions: ...

CMFEditions²¹ è uno dei componenti di Plone che si occupa del versionamento dei contenuti.

Usando Plone infatti, ogni volta che viene eseguita una modifica ad un contenuto definito “versionabile”, viene salvata la copia precedente, creando così una storia potenzialmente infinita del documento.

²⁰ <http://blog.keul.it/2011/08/data-di-scadenza-pubblicazione-in-plone.html>

²¹ <http://pypi.python.org/pypi/Products.CMFEditions>

Il prodotto è in qualche modo legato ad un altro dei componenti di Plone (disattivato di default ma presente in ogni installazione) che è il supporto alla [copia di lavoro](#)²² (*Working Copy*). Questo prodotto aggiunge numerose opzioni nel menù “Azioni”.

Va detto che il codice che si occupa del versionamento di Plone è piuttosto confuso e non sempre è facile capirne il funzionamento. Anche analizzando il codice si rischia spesso di trovarsi a verificare librerie sempre diverse, tutte in qualche modo collegate.

Note: Non va confusa la storia di un documento Plone con le transazioni dello ZODB. L'esecuzione dell'operazione di [pack dello ZODB](#)²³ di un sito Plone *non* interferisce col numero di versioni di un documento salvate ma solo con la possibilità di poter annullare (*undo*) le operazioni effettuate.

Il prodotto definisce quindi una serie di permessi aggiuntivi, tutti raccolti sotto il prefisso *CMFEditions*. A noi interessa analizzare solo un sotto-insieme di questi permessi poiché i rimanenti non sono nei fatti utili al funzionamento di Plone.

CMFEditions: Access previous versions

Questo permesso determina il potere dell'utente di accedere alla storia del documento e controlla la comparsa del link “Cronologia” e l'effettivo potere di utilizzarne le funzionalità.



Fig. 5.6: Il link alla “Cronologia” dal documento

CMFEditions: Apply version control

Questo permesso viene qui documentato solo perché *sembra* usato da uno dei metodi che si occupano del versionamento dei contenuti (`applyVersionControl`, nel tool `CopyModifyMergeRepositoryTool`). Dovrebbe essere utilizzato e verificato quando la storia del documento inizia (quindi alla sua creazione). In più un'installazione base di Plone imposta questo permesso ai ruoli *Contributore*, *Manager*, *Possessore*, *Editor*, *Revisore* e *Amministratore del sito*.

Leggendo il codice, *sembrerebbe* che una verifica di questo permesso venga fatta se il metodo di versionamento del contenuto è impostato su “Manuale” (una funzionalità di Plone usata piuttosto raramente).

Dopo una prova empirica: anche rimuovendo il permesso a tutti i ruoli non sembra esserci nessun effetto negativo sul comportamento del versionamento.

Il consiglio è: tenete i ruoli predefiniti ma per sicurezza assegnate questo permesso anche ad ipotetici nuovi ruoli che vorrete andare a creare e che possono avere poteri di modifica di qualunque tipo sui contenuti.

²² <http://pypi.python.org/pypi/plone.app.iterate>

²³ <http://plone.org/documentation/faq/how-do-i-pack-the-zodb>

CMFEditions: Checkout to location

Ci si potrebbe aspettare che questo permesso controlli la funzionalità del supporto alla copia di lavoro di effettuare il **checkout** (la creazione della copia di lavoro) in una certa posizione.

Sbagliato... questo permesso non fa assolutamente nulla. Eppure sono quasi certo che l'intenzione iniziale fosse esattamente questa.

Un permesso simile potrebbe essere “*iterate : Check out content*” (ma anche questo sembrerebbe inutilizzato).

CMFEditions: Revert to previous versions

Questo permesso è collegato alla possibilità di tornare alla versione precedente di un contenuto. Il problema è che nelle versioni moderne di Plone i template che controllano la storia sono cambiati.

Oggi il controllo delle versioni avviene tramite un moderno pop-up.

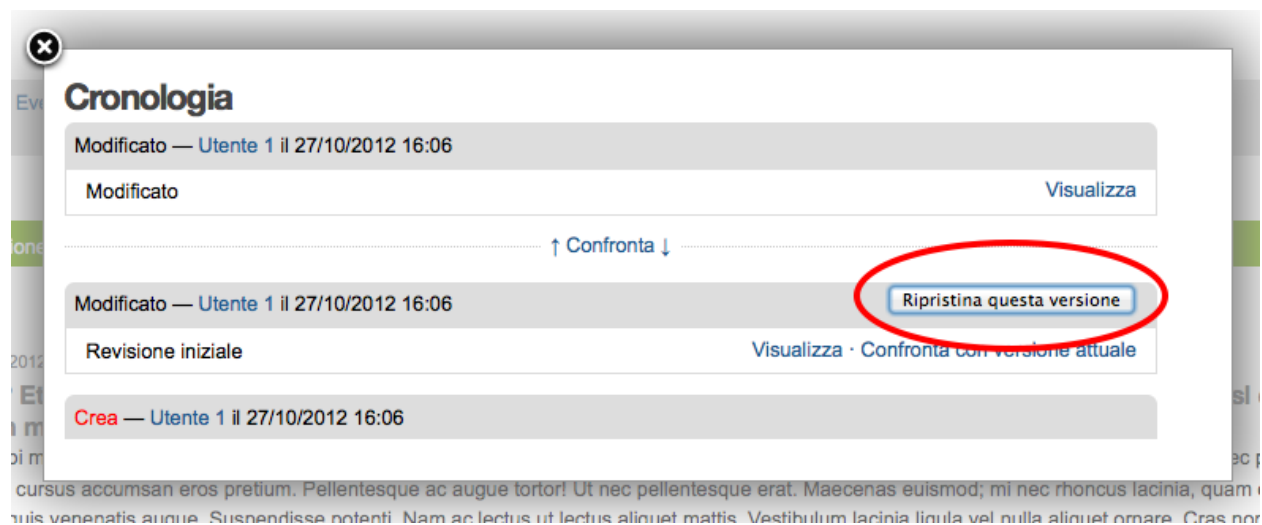


Fig. 5.7: Come compare oggi la storia del documento, dopo aver cliccato sul link “Cronologia”

Rimuovendo quel permesso agli utenti, visivamente non cambia nulla, il form rimane tale e quale. Premendo però il pulsante “*Ripristina questa versione*” si ottiene il permesso di permessi insufficienti.

Nei vecchi template di Plone, quando i controlli della versione del documento erano fatti tramite il tab aggiuntivo “*Storia*” (oggi disabilitato) le cose andavano meglio. La pagina è ancora oggi disponibile chiamando `/versions_history_form` sul contesto.

In questo vecchio template in assenza del permesso il pulsante “*Ripristina a questa versione*” sparisce (comportamento ovviamente migliore). Il comportamento attuale è molto probabilmente un piccolo bug, ma l'importante è che questo permesso controlli davvero questo potere.

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Possessore*
- *Editor*
- *Revisore*

Versioni di "Un documento"

creato da [Utente 1](#) — ultima modifica 27/10/2012 16:06

Versione	Effettuato da	Data e ora	Commento	Azioni
Copia di lavoro	utente1	27/10/2012 16:06	Modificato	Confronta con la versione precedente
0 (anteprima)	utente1	27/10/2012 16:06	Revisione iniziale	- Confronta con la versione corrente - Ripristina a questa versione

Fig. 5.8: Vecchia pagina della storia del documento

CMFEditions: Save new version

Questo permesso controlla il poter salvare una nuova versione di un documento, quindi in caso del semplice versionamento (automatico o manuale che sia) è un permesso necessario anche per salvare il documento. Se il prodotto per il supporto alla “Copia di lavoro” è attivo, questo permesso controlla anche il **checkin** del documento.

Nel caso del versionamento del contenuto Plone ha un comportamento che potrebbe non essere chiaro. Se l’utente corrente ha il potere di modificare il documento, egli può entrare nella pagina di modifica, ma se il versionamento è attivato e l’utente non possiede questo permesso, ottiene un errore al salvataggio (poiché salvando si sta tentando di creare anche una nuova versione). Forse la cosa andrebbe gestita in un altro modo (non creando una versione, oppure segnalando il problema all’utente in un modo alternativo).

Se l’estensione per la copia di lavoro è attiva e si tenta di eseguire il *checkin*, la cosa sembra funzionare ma non appena l’utente inserisce il commento alla modifica ottiene di nuovo l’errore di permessi insufficienti. Anche in questo caso il comportamento non è ottimale: sarebbe meglio che all’utente fosse inibita la voce di menù che scatena il *checkin*.

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Possessore*
- *Contributore*
- *Editor*
- *Revisore*

La presenza del ruolo *Contributore* è dubbia (perché il *Contributore* ha il diritto di generare una nuova versione di un documento quando potenzialmente non avrebbe i diritti di modificarlo?).

5.3.6 Change portal events

Questo permesso, per ragioni storiche, è **il permesso di modifica degli eventi**.

È da gestire allo stesso modo con cui viene usato il più famoso *Modify portal content*. È anche molto probabile che l’importanza di questo permesso venga meno non appena gli eventi di Plone verranno sostituiti dal prodotto [plone.app.event](http://pypi.python.org/pypi/plone.app.event)²⁴, nelle future versioni di Plone.

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*

²⁴ <http://pypi.python.org/pypi/plone.app.event>

- *Amministratore del sito*
- *Possessore*

5.3.7 Delete objects

Questo permesso controlla il potere di cancellare contenuti ma vista la sua complessità e il suo comportamento non sempre chiaro, c'è molto da dire.

Innanzitutto: in Plone ci sono due modi in cui è possibile eliminare un contenuto:

1. Dal **menù “Azioni”** (cancellazione del documento corrente)

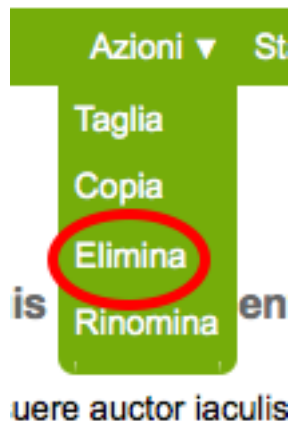


Fig. 5.9: Come eliminare il contesto corrente

2. Dalla **vista contenuti** (cancellazione di uno o più contenuti figli)

Nel primo caso il codice Plone richiama lo script `delete_confirmation.cpy` che a sua volta richiama il metodo di basso livello `manage_delObjects` sul padre dell'elemento che si vuole cancellare.

Nel secondo caso si passa invece per lo script `folder_delete.cpy` che, in modo indiretto, arriva sempre a richiamare lo stesso metodo `manage_delObjects` (questa volta: sul contesto corrente in quanto già padre degli elementi che si vogliono cancellare) fornendo una serie di id, che verranno tutti cancellati.

Anche gli elementi grafici dell'interfaccia Plone (la voce “*Elimina*” nel menù “*Azioni*” e il pulsante “*Elimina*” nella vista contenuti) sono mostrati o nascosti in presenza dello stesso permesso.

Il problema della cancellazione dei contenuti in Plone

Questo comportamento è a volte limitante e considerato inadatto: se un utente ha il potere di cancellare i contenuti di una cartella allora *può cancellarli tutti*. Non è possibile rendere cancellabili alcuni contenuti in base al loro stato di revisione del workflow poiché la verifica viene fatta comunque sul padre, è possibile solo determinare che, se il padre è in un certo stato di revisione, allora i suoi contenuti figli saranno o non saranno cancellabili.

Un comportamento che a mio avviso dovrebbe essere rispettato di base è che un utente non possa cancellare elementi che non è in grado di modificare (così come funziona un filesystem).

Tu sei qui: [Home](#)

Contenuti Visualizza Regole Condivisione

Plone: workflow e sicurezza

ultima modifica 15/09/2012 13:29

Selezione: **Tutti**

		Titolo	Dimensione	Modificato	Stato
⋮	☐	 Benvenuto in Plone ■	5.1 kB	15/09/2012 13:29	Pubblicato
⋮	☐	 Notizie	1 kB	26/10/2012 16:45	Pubblicato
⋮	☐	 Eventi	1 kB	15/09/2012 13:29	Pubblicato
⋮	☐	 Collaboratori	1 kB	15/09/2012 13:29	Pubblicato
⋮	☐	 Uffici	1 kB	06/10/2012 20:05	Pubblicato
⋮	☒	 Un documento	2.6 kB	28/10/2012 10:58	Privato

Fig. 5.10: Come eliminare i contenuti di una cartella

Per raggiungere questo obiettivo è necessario modificare parte del codice Plone (in realtà un'operazione fattibile direttamente da ZMI), oppure rimanere ad un livello superficiale: modificare solo l'interfaccia grafica.

Questa è quella che viene detta “sicurezza tramite oscuramento” (“[Security through obscurity](http://en.wikipedia.org/wiki/Security_through_obscurity)²⁵”) quindi non una vera e propria sicurezza: se l'utente infatti conosce il funzionamento di Plone, potrà comunque bypassare la vostra modifica.

In alcune situazioni (e.g. una intranet) è comunque una scelta tutto sommato accettabile.

5.3.8 List folder contents

Questo permesso è quello che permette agli utenti di vedere i contenuti di una cartella, quindi la sua modifica ha effetti solo sui contenuti di tipo simil-cartella, e controlla la presenza del tab “*Contenuti*”.

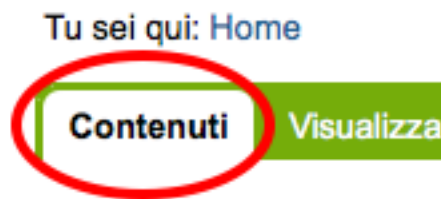


Fig. 5.11: Link al tab dei contenuti della cartella

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Possessore*
- *Contributore*
- *Editor*
- *Revisore*

In pratica tutti i ruoli che di solito hanno qualche tipo di potere dalla vista dei contenuti della cartella.

5.3.9 Manage portal

Note: È il permesso di riferimento del ruolo **Manager**

Questo permesso determina tantissimi poteri, tutti legati ad azioni che di solito può fare solo il ruolo Manager.

Ad oggi può creare problemi di incompatibilità col ruolo “*Amministratore del sito*” in presenza di prodotti che ancora non supportano quest'ultimo ruolo (vedere [la discussione relativa](#)).

Un esempio classico è l'**uso delle portlet**, che in Plone sono sempre state gestite dal *Manager* e di recente dal nuovo ruolo *Amministratore del sito*, ma è possibile ancora oggi trovare vecchi prodotti aggiuntivi che forniscono nuove portlet usando questo permesso, e sono quindi inutilizzabili dal nuovo ruolo. Un permesso più corretto sarebbe “*Portlets: Manage portlets*”.

²⁵ http://en.wikipedia.org/wiki/Security_through_obscurity

5.3.10 Modify portal content

Note: È il permesso di riferimento del ruolo **Editor**

A parte qualche eccezione degna di nota (vedere “*Change portal events*”), questo è il permesso che identifica il potere di modificare i contenuti.

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Possessore*
- *Editor*

L'importanza di questo permesso è altrove, gestito tramite l'uso dei **workflow**.

5.3.11 Portlets: Manage portlets

È il permesso che permette di accedere alla gestione delle portlet laterali ed è per questo motivo assegnato al *Manager* e all'*Amministratore del sito*.

In assenza di un permesso specifico per gestire una nuova portlet (magari in seguito all'installazione di un prodotto aggiuntivo), questo è il permesso che andrebbe utilizzato, anche se la soluzione migliore sarebbe sempre quella di avere un permesso per ogni tipo di portlet.

Purtroppo questo non succede: tutte le portlet predefinite di Plone sono gestite da quest'unico permesso, eccezione fatta per due casi:

- “*plone.portlet.collection: Add collection portlet*” (per gestire le **portlet collezione**)
- “*plone.portlet.static: Add static portlet*” (per gestire le **portlet statiche**)

5.3.12 Request review

È il permesso che identifica il potere di un utente di sottoporre un documento alla richiesta di revisione (di solito effettuata dal *Revisore*).

Di solito si traduce della presenza di una specifica voce nel menù di cambio di stato.

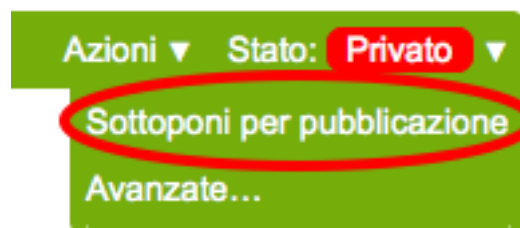


Fig. 5.12: La richiesta di sottoporre a revisione un documento, nel menù del workflow

È utilizzata in tutti i workflow base, ma se avete intenzione di creare un vostro workflow e vi serve questa funzionalità, tenete presente questo permesso prima di volerne creare altri.

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Editor*
- *Possessore*

5.3.13 Review portal content

Note: È il permesso di riferimento del ruolo **Revisore**

Questo permesso identifica il potere di revisionare un contenuto del sito, di solito legato ad una precedente richiesta di revisione ottenuta tramite uso di workflow.

Come già discusso per il permesso “*Request review*”, vale la pena riutilizzare il permesso anche in presenza di workflow personalizzati.

Di solito si traduce della presenza di voci aggiuntive nel menù di cambio di stato, una per pubblicare il contenuto (richiesta accettata) e un'altra voce per rifiutarlo.



Fig. 5.13: Pubblicazione o rifiuto del documento, nel menù del workflow

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Revisore*

5.3.14 Sharing page: ...

Questa serie di permessi controlla l'accesso alla **pagina di condivisione** e la possibilità di assegnare ad utenti e gruppi i singoli permessi disponibili in questa pagina.

Questi permessi sono già stati introdotti brevemente alla sezione “*L'accesso alla pagina di condivisione*” nel capitolo sui ruoli ma il loro comportamento necessita di maggiori delucidazioni.

Il permesso generale che determina la **possibilità di accedere alla pagina di condivisione** è “**Sharing page: Delegate roles**”.

Questo è il permesso più importante e viene verificato prima di tutti gli altri. Questo permesso è quindi assegnato a tutti gli utenti che possono assegnare qualche ruolo ad altri utenti del sito.

Nel nostro esempio del “*Super Revisore*” (vedere “*Comportamento del Super Revisore nella pagina di condivisione*”) ci eravamo limitati ad usare questo permesso e l’effetto ottenuto era quello di rendere possibile a tutti gli utenti in grado di condividere un documento, il potere di assegnare anche il ruolo.

Per i ruoli predefiniti di Plone (ed è quello che faremo anche per il nostro nuovo ruolo) esiste invece un permesso specifico per ogni ruolo.

Questi sono:

- **Sharing page: Delegate Contributor role**
- **Sharing page: Delegate Editor role**
- **Sharing page: Delegate Reader role**
- **Sharing page: Delegate Reviewer role**

Con questo meccanismo è possibile arrivare ad un livello di granularità estremo:

1. Si decide quali ruoli possono condividere il documento
2. Si decide quali ruoli è possibile fornire

In seguito vedremo come creare il nuovo permesso che al momento ci manca.

5.3.15 View

Note: È il permesso di riferimento del ruolo **Lettore**

Il permesso più semplice, eppure il più importante tra tutti i permessi. Determina il potere di vedere il contenuto.

Anche se, come tutti gli altri permessi, è gestibile nella radice del sito o alla radice di Zope, il suo scopo è quello di essere **gestito nei contenuti tramite workflow**.

Noterete infatti che il permesso, a livello di radice di Zope, è assegnato agli *Anonimi*, il che significa che *chiunque* deve poter accedere al sito Plone. Se state leggendo queste pagine perché volete disegnare una intranet, potreste pensare come questa impostazione sia qualcosa da cambiare, ma non è vero.

Disabilitando il permesso di *View* alla radice del sito non è il modo corretto. Gli utenti (anche gli anonimi) devono poter raggiungere il sito, per poi essere obbligati ad effettuare l’autenticazione.

Note: Togliere il permesso di *View* all’oggetto “Sito Plone” ha l’effetto di obbligare gli utenti ad eseguire un’autenticazione HTTP Basic, ma questa impostazione può portare a dei problemi difficili da gestire.

Non fatelo.

Da questo momento in poi parleremo del permesso sempre riferendoci alla sua presenza o assenza relativamente a contenuti.

Che cosa viene influenzato da “View”?

Il permesso influenza due comportamenti: la **ricerca** e l'**accesso diretto ai contenuti**.

Per ricerca si intende tutto ciò che in Plone si risolve con l'uso del **catalogo**, il che si traduce non solo nella ricerca tramite il campo di ricerca istantanea o la ricerca avanzata, ma anche l'uso delle collezioni, delle viste che mostrano i contenuti di una cartella, delle portlet, nel navigatore, etc.

In pratica la mancanza del permesso di *View* influenza tutto ciò che in Plone può generare liste dinamiche di contenuti. Deve essere chiaro che nel momento stesso in cui un utente perde il permesso di *View* relativamente ad un contenuto (di solito: in seguito ad un cambio di stato nel workflow), l'interfaccia di Plone reagisce facendo sparire per l'utente il contenuto.

Ma la sicurezza non è tutta qui. Se l'utente provasse comunque ad accedere al contenuto (magari tramite un link, un bookmark o semplicemente perché ne conosce l'URL) viene verificata la presenza del permesso per i ruoli dell'utente. In caso negativo, si viene rediretti alla pagina di permessi insufficienti.

“View” e il catalogo: `allowedRolesAndUsers`

Diciamo qualche parola in più sulle ricerche di Plone e le relazioni con il catalogo.

Il catalogo si occupa di *indicizzare* i contenuti in base a vari *indici* differenti e nel contempo di memorizzare alcuni dati del contenuto stesso.

Il motivo: l'accesso ad un contenuto Plone ha un certo costo in termini di consumo di risorse, costo irrisorio se si parla di accedere ad un singolo contenuto ma che può diventare grande se l'operazione richiesta necessitasse di accedervi centinaia... o migliaia.

Se non ci fosse il catalogo ed un utente si trovasse ad eseguire una ricerca per la parola “*Tasse*”, sarebbe necessario caricare uno ad uno tutti i contenuti del sito e poi controllare se la parola è compresa in uno dei campi del documento trovato. Impensabile.

Ma questo non basta. Se il catalogo ritornasse un set di risultati con 100 contenuti che parlano di *Tasse* e questi venissero direttamente mostrati all'utente, potrebbero esserci problemi di sicurezza: va verificato se l'utente ha i diritti (il permesso di *View*) per accedere al contenuto.

Per fare questo sarebbe comunque necessario caricare i contenuti prima di riportarli come risultato all'utente, invalidando in buona parte i benefici del catalogo.

Per questo esiste uno speciale indice: `allowedRolesAndUsers`. Questo permesso memorizza per ogni contenuto del sito i ruoli, gli utenti e i gruppi che possono accedervi (quindi verificandone il permesso di *View*). L'uso di questo indice è sempre aggiunto a qualunque tipo di ricerca in modo trasparente all'utente.

Quindi in Plone è possibile chiedere al catalogo se un certo utente ha il permesso di *View* su un certo contenuto, cosa che non è possibile con nessun altro permesso.

Un buon esempio dell'approccio è il prodotto [collective.portlet.truereview](http://pypi.python.org/pypi/collective.portlet.truereview)²⁶, un componente (non molto conosciuto) che aggiunge a Plone una nuova portlet di revisione. Questa portlet a differenza di quella originale fornita col CMS (che in alcuni casi può diventare estremamente lenta, proprio perché non può usare il catalogo) utilizza lo stesso approccio dell'indice che abbiamo introdotto, applicando lo stesso principio con un nuovo indice: `reviewerRolesAndUsers`.

“View” e i documenti scaduti

Dei documenti scaduti si è già parlato in relazione del permesso “*Access inactive portal content*”.

²⁶ <http://pypi.python.org/pypi/collective.portlet.truereview>

Ripetiamo qui una precisazione: è possibile che un documento scaduto sia “*visibile*” ad un certo utente (qui inteso come: “l’utente ha il permesso di *View* sul documento”) eppure che non riesca a trovarlo, perché senza il permesso per vedere contenuti scaduti.

In questo caso l’accesso diretto non mente: “*Access inactive portal content*” influenza solo le ricerche ma l’utente può accedere al contenuto andando direttamente all’URL.

5.3.16 plone.app.collection: Add Collection

Questo permesso è stato introdotto con le nuove *Collezioni* ed è relativo al potere di aggiungere collezioni nel sito.

Vale quanto detto per i *permessi di aggiungibilità dei contenuti*.

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*

5.4 Dove i permessi incidono sull’interfaccia Plone

Note: Per tutti gli esempi seguenti, vale sempre la regola dell’uso della ZMI per effettuare modifiche.

Modificare le impostazioni via ZMI e non esportare le modifiche rende la vostra configurazione difficile da replicare, o eseguirne il debug se qualcosa va storto.

Segue una serie di punti da dove è possibile modificare le impostazioni dell’uso dei permessi tramite ZMI e le cui modifiche hanno immediati effetti sul comportamento di Plone.

5.4.1 Il tool `portal_actions`

Il primo elemento di ZMI che andiamo a visitare è anche il più ricco in assoluto di impostazioni. È il **`portal_actions` tool**, accessibile tramite la ZMI di ogni sito Plone.

Si occupa di gestire la presenza di elementi dell’interfaccia Plone, solitamente sotto forma di link, o pulsanti di form.

Entrati nel tool vengono mostrate una serie di elementi “**CMF Action Category**”, che non sono altro che gruppi di *azioni* (**CMF Action**).

Il funzionamento generale è il seguente: per ogni categoria ci possono essere una serie di una o più azioni. Prodotti aggiuntivi potrebbero creare nuove tipologie di azioni (raro, ma non impossibile poiché questo tool è ottimo per configurare URL da usare nell’interfaccia Plone).

Andando in creazione o in modifica di una nuova azione all’interno di una categoria, ci si trova di fronte ad uno spettacolo del genere:

Non ci soffermeremo sull’intero form mostrato, ma solo sulla sezione “*Permissions*”. Questa permette di configurare l’azione con un filtro che richieda un permesso specifico nel contesto su cui l’azione deve poi essere utilizzata.

L’utente deve avere almeno uno dei permessi selezionati per poter vedere l’azione. Non è possibile specificare più permessi in “*AND booleano*” (verificare se l’utente ha tutti i permessi di un certo insieme). La selezione del permesso non è obbligatoria; non selezionando nessun permesso rende dittiva la verifica (di solito comunque viene sempre indicata la presenza del permesso “*View*”).

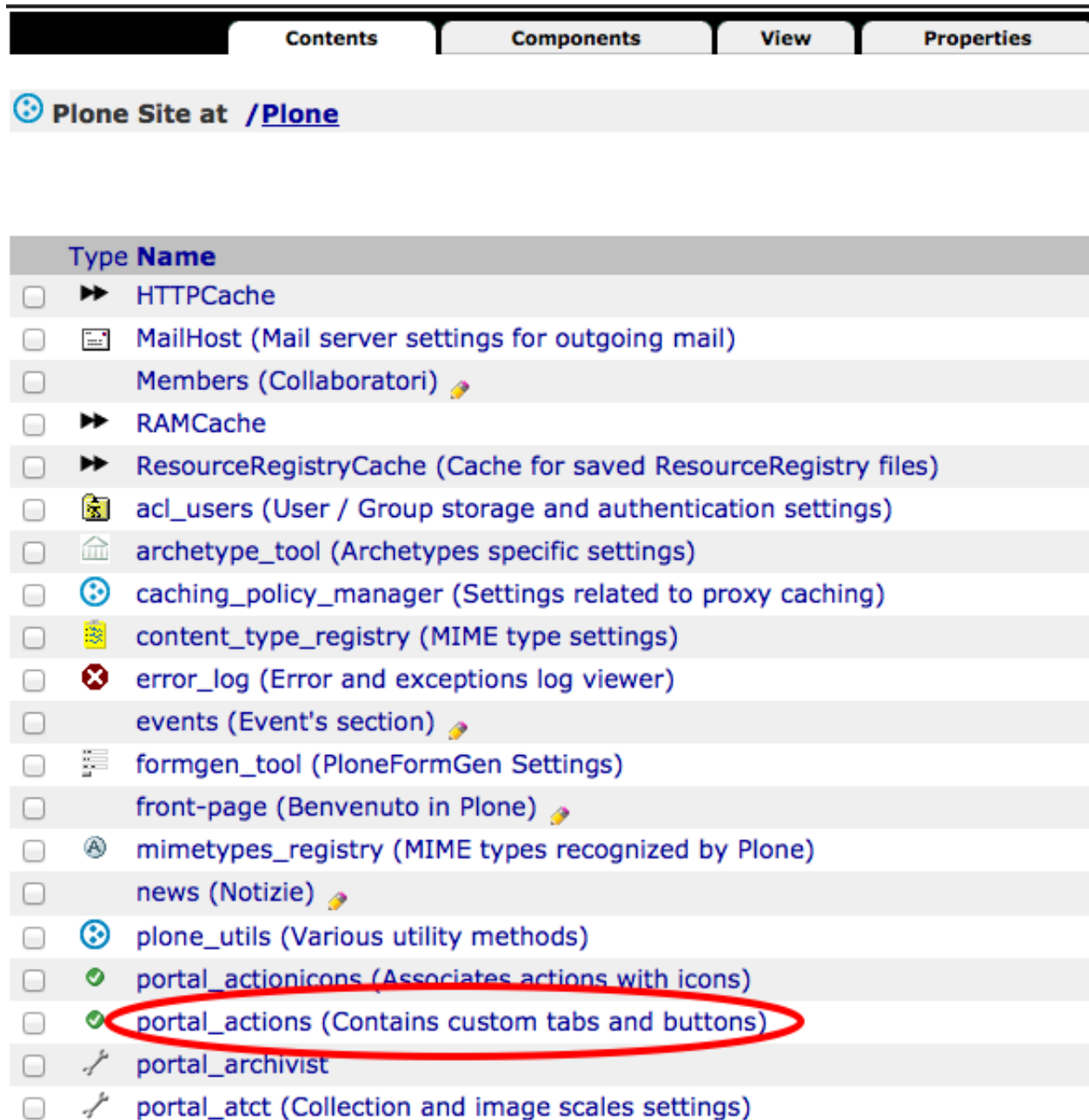


Fig. 5.14: Il tool `portal_actions` visto dalla radice del sito Plone, in ZMI

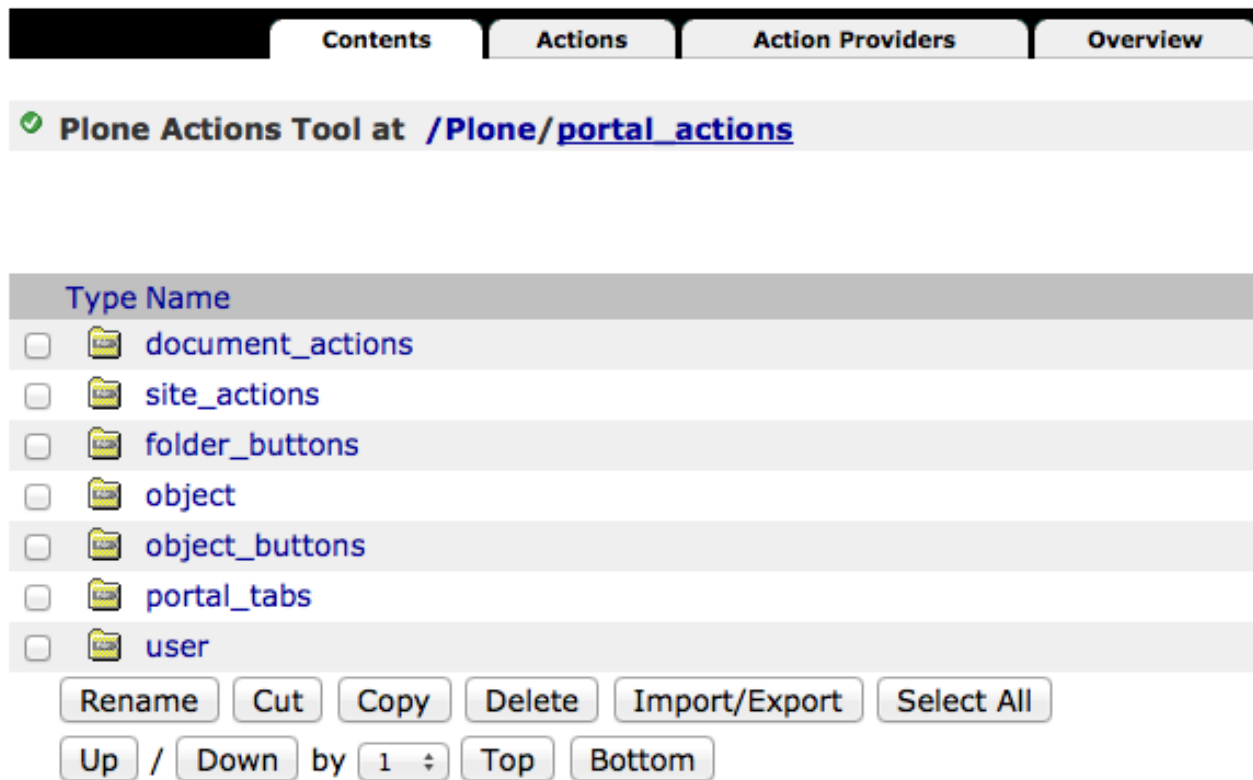


Fig. 5.15: Come si presenta il `portal_actions` tool in un sito Plone

Properties

Undo

Ownership

CMF Action at /book/portal_actions/foo/baz

Properties allow you to assign simple values to Zope objects. To change property values, edit the values and click "Save Changes".

Name	Value	Type
Title		string
Description		text
I18n Domain		string
URL (Expression)		string
Link Target		string
Icon (Expression)		string
Condition (Expression)		string
Permissions	ATContentTypes Topic: Add ATBooleanCriterion ATContentTypes Topic: Add ATCurrentAuthorCriterion ATContentTypes Topic: Add ATDateCriteria ATContentTypes Topic: Add ATDateRangeCriterion ATContentTypes Topic: Add ATListCriterion ATContentTypes Topic: Add ATPathCriterion ATContentTypes Topic: Add ATPortalTypeCriterion	multiple selection
Visible?	☒	boolean

Save Changes

Delete

Fig. 5.16: La creazione di una nuova CMF Action all'interno del portal_actions tool

Per avere invece la verifica di più permessi si ricorre spesso all'uso della voce “*Condition (Expression)*”, che permette di scrivere un'espressione Python per eseguire una condizione arbitraria (tra cui anche la verifica di permessi).

Se la necessità fosse verificare due permessi, si potrebbe verificare un primo permesso nel modo canonico e un secondo permesso tramite l'uso di un'espressione.

Segue una forma standard per ottenere questo tipo di espressioni:

```
python:checkPermission("nome del permesso", object)
```

Qui sopra viene verificato tramite un'espressione Python (con l'uso della funzione `checkPermission`), che l'utente corrente abbia il permesso passato come stringa, sul contesto corrente (identificato da `object`).

Se fosse necessario verificare due (o più) permessi tramite l'espressione:

```
python:checkPermission("permesso1", object) and checkPermission("permesso2", object)
```

Vediamo ora le azioni più importanti e il loro impatto sull'interfaccia. Nell'elenco che segue salteremo varie categorie di azioni, poiché usano di solito sempre il permesso *View*; ciò non toglie che l'utente possa aggiungere nuovi azioni in queste categorie, proteggendole con altri permessi.

folder_buttons

Questa categoria viene utilizzata per popolare i pulsanti che vengono mostrati nella vista dei contenuti di una cartella.



Fig. 5.17: I pulsanti mostrati nella vista dei contenuti di una cartella, popolati grazie alla categoria *folder_buttons*

copy (Copia) Controlla la presenza del pulsante di “Copia” di uno o più contenuti ed è controllato dal permesso “*Copy or Move*”.

cut (Taglia) Controlla la presenza del pulsante per eseguire il “Taglia” di uno o più contenuti.

Vista la particolarità delle operazioni di taglio (che necessitano anche della cancellazione del contenuto dalla cartella corrente) vengono verificati due permessi: “*Copy or Move*” e “*Delete objects*”.

rename (Rinomina) Controlla la presenza del pulsante di “*Rinomina*” di uno o più contenuti.

Rinominare un contenuto è visto in qualche modo come un re-inserirlo nella cartella (con un nome diverso) quindi il pulsante è controllato dal permesso “*Add portal content*”.

paste (Incolla) Controlla la presenza del pulsante di “*Incolla*”, per inserire nella cartella uno o più contenuti.

Dovendo inserire nuovi contenuti nella cartella, viene verificato il permesso “*Add portal content*”.

delete (Elimina) Controlla la presenza del pulsante di “*Elimina*”, per cancellare uno o più contenuti dalla cartella.

Come spiegato nella sezione “*Il problema della cancellazione dei contenuti in Plone*”, il permesso utilizzato è solo “*Delete objects*” (sulla cartella stessa).

change_state (Cambia lo stato) Permette di controllare il pulsante “*Cambia lo stato*”, che porta l’utente alla vista “*Processo di pubblicazione*”.

Da questa pagina è possibile modificare lo stato di revisione di tutti i contenuti selezionati (potendo anche inserire un **commento di revisione** unico per tutti i contenuti) e modificarle le date di pubblicazione e scadenza (un’accoppiata di funzionalità non facili da giustificare).

Si arriva a questa stessa pagina anche dal menù “*Stato*” che controlla i workflow (voce “*Avanzate...*”).

Non è semplice capire con che permesso rendere disponibile questo pulsante, viste le funzionalità differenti che offre. È quindi protetto dal permesso di “*View*”, ma l’espressione verifica invece altri due permessi: “*Modify portal content*” e “*Review portal content*”.

object

La categoria **object** racchiude una serie di link che vengono visualizzati in tutti i contenuti del sito tramite **tab** agli autenticati.



Fig. 5.18: I tab mostrati sui contenuti, con evidenza a quelli forniti dalla categoria “object”

folderContents (Contenuti) Controlla la comparsa del tab “**Contenuti**”, che mostra i contenuti della cartella corrente. Per questo motivo è protetto dal permesso “*List folder contents*”.

syndication (Distribuzione) Un vecchio tab deprecato ed ora disabilitato, che controllava l’accesso al form “*Proprietà della distribuzione*”.

contentrules (Regole) Controlla la comparsa del tab “**Regole**” per accedere al form di controllo delle regole di contenuto. È controllato dal permesso “*Content rules: Manage rules*”.

local_roles (Condivisione) Controlla la comparsa del tab “**Condivisione**” per accedere alla condivisione dell’elemento corrente. È controllato dal permesso “*Sharing page: Delegate roles*”.

object_buttons

La categoria **object_buttons** può erroneamente far pensare a “bottoni”, ma è invece usata per popolare il contenuto del menù “Azioni”.

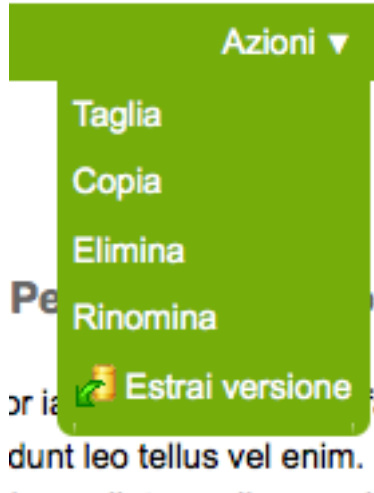


Fig. 5.19: Il menù “Azioni” con tutte le opzioni predefiniti e col supporto alla copia di lavoro installato

cut (Taglia) Controlla la presenza della funzionalità di “Taglia” sul contenuto.

Vista la particolarità delle operazioni di taglio (che necessitano anche della cancellazione del contenuto dalla cartella corrente), tramite una combinazione di uso dei permessi dell’azione ed espressione di controllo vengono verificati due permessi: “*Delete objects*” (sul contenuto e sul suo contenitore) e “*Copy or Move*” (sul contenitore).

copy (Copia) Controlla la presenza dell’azione di “Copia” del contenuti. Per qualche motivo è protetto dal permesso di “View”, mentre la verifica del permesso reale è lasciata all’espressione di controllo (che verifica la presenza di “*Copy or Move*”).

paste (Incolla) Controlla la presenza dell’azione di “Incolla”, per aggiungere il contenuto precedentemente copiato/tagliato nella cartella che contiene l’elemento corrente.

Dato che il contesto corrente non ha nulla a che fare con il nuovo elemento che si va a copiare, viene verificato il permesso di “View” e la presenza del pulsante è lasciata ad un’espressione che verifica se ci sono dati validi da incollare.

Tutto questo sembra molto permissivo (e lo è... perché non viene invece verificato il permesso “*Add portal content*” sul contesto del padre?) ma se poi l’utente non ha nei fatti i poteri per incollare, ottiene un messaggio di errore.



Fig. 5.20: Il messaggio di errore mostrato se non si hanno permessi per incollare elementi

La sicurezza è quindi rispettata, ma sarebbe a mio avviso più corretto non far comparire il pulsante.

delete (Elimina) Controlla la presenza dell’azione di “Elimina” del contenuto corrente.

Perché il controllo compaia viene verificato il permesso “*Delete objects*” sia sul contenuto che sul suo contenitore.

rename (Rinomina) Controlla la presenza dell'azione di “*Rinomina*” del contenuto.

Rinominare un contenuto è visto in qualche modo come un re-inserirlo nella cartella (con un nome diverso). In questo caso viene fatta una complessa lista di verifiche:

- “*Add portal content*” sul contenuto
- “*Delete objects*” sul contenitore
- “*Copy or Move*” sul contenitore
- “*Add portal content*” sul contenitore

Segue una lista di altre tre azioni, disponibili solo se viene attivato il componente opzionale per il supporto alla *copia di lavoro* (Working Copy).

I limiti attuali dei permessi di questo prodotto sono stati introdotti quando si è parlato dei “*permessi relativi a CMFEditions*”.

iterate_checkout (Estrai versione) L'azione che permette di creare una nuova copia di lavoro, partendo dal contenuto corrente.

L'azione è protetta dalla presenza del permesso di “*View*”, perché tutta la logica è racchiusa nella chiamata ad un metodo `checkout_allowed`.

iterate_checkin (Crea versione) L'azione compare solo sulle copie di lavoro di altri contenuti. Permette di far “rientrare” il documento corrente nel documento principale, come nuova versione di quest'ultimo.

L'azione è protetta dalla presenza del permesso di “*View*”, perché tutta la logica è racchiusa nella chiamata ad un metodo `checkin_allowed`.

iterate_checkout_cancel (Annulla il check-out) L'azione compare solo sulle copie di lavoro di altri contenuti. Permette di annullare la copia di lavoro (nei fatti eliminando il contenuto).

L'azione è protetta dalla presenza del permesso “`ref:section-permission-modify-portal-content`” (perché non il permesso per cancellare?) e dalla verifica alla chiamata del metodo `cancel_allowed`.

portal_tabs

I **tab del portale** identificano quella zona che normalmente racchiude i link sotto alla testata del sito.

Questa zona è popolata dalle azioni definite in questa categoria, ma anche da tutti i contenuti nella radice del sito (questo se nella configurazione del sito, nelle impostazioni della Navigazione è stata selezionata la voce “*Genera automaticamente le schede*”).

I tab del portale hanno una particolarità: non si riferiscono al contesto corrente ma sempre alla radice del sito (i permessi sono quindi verificati sul sito Plone). Questo è corretto, anche se limita notevolmente l'utilizzo di questa sezione.

Di base esiste una sola voce: **index_html** (Home) che è un link alla home del sito (quindi protetto dal semplice permesso “*View*”).

Tutto questo potrebbe far sembrare questa categoria di azioni poco importante, ma nel complesso è invece una tra le aree più sfruttabili. Quest'area può comunque essere usata per mostrare altri link utili, magari a siti esterni, che non coincidono per forza con contenuti del sito (o contenuti del sito nella radice di questo).

users

Questa categoria raccoglie tutte le azioni presenti nel menù a tendina riservato agli utenti autenticati; se l'utente è un anonimo e per impostazioni ha più di un'azione a disposizione (di solito *Fatti riconoscere* e *Iscriviti*) le azioni vengono mostrate come una serie di link affiancati.

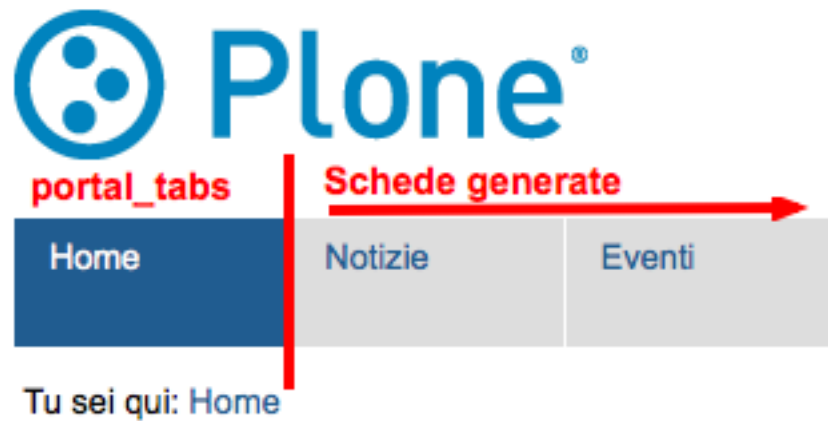


Fig. 5.21: La separazione tra la sezione dei tab del portale e le schede generate automaticamente



Fig. 5.22: Il menù degli strumenti personali

Questo menù di base è fortemente influenzato dal ruolo dell'utente. Se l'utente è autenticato, viene mostrato il suo nome come voce principale del menù ed espandendolo vengono mostrate le altre opzioni.

Anche in questo caso: i permessi sono sempre verificati sulla radice del sito, il che limita notevolmente la manipolazione dei permessi.

mystuff (Cartella personale) È il link alla cartella personale degli utenti (se abilitata).

È protetto dal semplice permesso di “View” (in pratica: non è usato nessun permesso) ma compare solo se la cartella personale dell'utente esiste (grazie ad un'espressione di controllo).

Vedere anche “*Le cartelle personali*”.

dashboard (Dashboard) È il link alla **dashboard** personale dell'utente ma per qualche motivo non è controllato dal permesso “*Portlets: View dashboard*” quanto invece dal permesso “*Portlets: Manage own portlets*”.

preferences (Preferenze personali) Il link alle preferenze personali dell'utente.

Non è protetto da un permesso specifico (viene usato *View*) ma compare automaticamente per ogni utente autenticato.

plone_setup (Configurazione del sito) Il link al pannello generale della configurazione del sito.

Per questo motivo controllato dal permesso “Plone Site Setup: Overview” (vedere *l'apposita sezione*).

login (Fatti riconoscere) È il link che permette l'autenticazione nel sito Plone (un nome migliore sarebbe probabilmente mantenere la forma inglese **log in**).

Non è protetto da nessun permesso particolare se non *View* ma compare solo agli utenti anonimi grazie ad un'espressione di controllo.

join (Iscriviti) Se nelle *impostazioni di sicurezza* è selezionata al voce **Consenti l'auto-registrazione** questa azione compare a tutti gli utenti anonimi e permette di crearsi autonomamente un account nel sito.

È protetto dal permesso “*Add portal member*”.

undo (Annulla) Controlla la presenza dell'azione che permette l'accesso al modulo “*Annulla azioni*” per effettuare l'annullamento di operazioni effettuate e tornare ad uno stato precedente del sistema.

Le operazioni di *undo* in Plone sono piuttosto delicate (non inteso come “pericolose”, ma molto spesso non possono essere effettuate e falliscono senza riuscire a dare all'utente una spiegazione ragionevole) quindi la voce è di solito disabilitata.

È controllata dal permesso “*List undoable changes*”.

review-comments (Moderazione commenti) Questo permesso controlla l'accesso alla pagina “*Moderazione commenti*”. Nel caso ci siano commenti da moderare sparsi per il sito, questi sono riassunti in questa pagina.

La voce è controllata dal permesso “*Review comments*” ma compare solo se ai commenti del sito è stato associato un workflow (come è spiegato alla sezione “*Commenti*” della configurazione del sito).

logout (Esci) Controlla la presenza del link che permette di uscire dalla sezione corrente (eseguendo appunto il log-out).

Non c'è un permesso particolare per questo contenuto, la sua presenza viene controllata dall'espressione.

5.4.2 Il tool `portal_types`

In passato Plone aveva vari tool della ZMI offrivano delle azioni che potevano influenzare l'interfaccia. A parte il tool principale appena descritto (*portal_actions*) erano presenti altri *action providers* secondari ma di questi ad oggi è rimasto solo il tool *portal_types*.

Lo scopo principale del **portal_types** tool non è direttamente legato all'interfaccia o alle azioni, ma racchiude la **registrazione di tutti i tipi** di contenuto del CMS.



Fig. 5.23: Il tool `portal_types` visto dalla radice del sito Plone, in ZMI

In ogni tipo di contenuto avete quindi a disposizione un familiare tab “Actions”:

Questo ci porta ad un form dove è possibile gestire un'altra categoria di azioni. La grossa differenza sta nel contesto: queste azioni sono solo legate al tipo di contenuto in esame.

Warning: Un chiaro messaggio in questo caso avverte che la funzionalità è in via di dismissione e che è sconsigliato aggiungere azioni in questo tool.

Rimane il fatto che ad oggi è ancora il posto più semplice da utilizzare per aggiungere azioni specifiche di un tipo di contenuto.

Le azioni qui definite vengono utilizzate nell'interfaccia grafica di Plone allo stesso modo con cui vengono mostrate i link nella categoria *object* del `portal_action` tool.

La differenza è sostanziale. Le azioni nella categoria *object* sono globali ed incidono contemporaneamente su tutti i tipi di contenuto; in caso negativo è necessario ricorrere a delle espressioni di controllo più o meno complesse.

Le azioni definite nel `portal_types` tool sono invece limitate al tipo specifico.

Ad oggi tutti i tipi di contenuto base di Plone definiscono due semplici azioni: *view* ed *edit* per determinare il link della scheda “Visualizza” e “Modifica”.

Le due azioni sono controllate (ovviamente) dai due permessi associati: *View* e *Modify portal content*.

Vale la pena notare come a prima vista sembrerebbe che queste due azioni, essendo uguali per tutti i contenuti del sito, possano essere spostate nel `portal_action` tool, come già accade per azioni quali *Contenuti* e *Condivisione*. Questo è probabilmente quello che presto succederà, ma ad oggi ci sono però piccole sfumature che rendono ancora comodo avere ed usare questo tool:

- Il link alla vista di un contenuto dipende dal tipo di contenuto. I contenuti di tipo *File* infatti vogliono che l'URL della loro vista principale termini con “/view” o il file viene invece scaricato direttamente.
- Le vecchie collezioni hanno un link aggiuntivo “*Criteri*”, che viene quindi controllato dal tipo stesso.

5.5 Creare nuovi permessi

Come accennato all'inizio del capitolo, ricordiamo che la creazione di nuovi permessi è un'operazione di programmazione.

Va anche ricordato come la creazione di nuovi permessi vada limitata il più possibile, nella maggior parte dei casi il permesso che *credete* di dover creare potrebbe essere già presente in Plone. Ad ogni modo il voler aggiungere nuovi permessi è quasi sempre legata alla presenza di codice (da voi sviluppato, o codice di terze parti che avete installato nel vostro ambiente Plone).

Il permesso in quanto tale non arricchisce Plone in nessun modo.

	Contents	Aliases	Actions	Overview	View																					
 Plone Types Tool at /Plone/portal_types																										
<table border="1"> <thead> <tr> <th>Type Name</th> </tr> </thead> <tbody> <tr><td>☐  ATBooleanCriterion (Boolean Criterion)</td></tr> <tr><td>☐  ATCurrentAuthorCriterion (Current Author Criterion)</td></tr> <tr><td>☐  ATDateCriteria (Friendly Date Criteria)</td></tr> <tr><td>☐  ATDateRangeCriterion (Date Range Criterion)</td></tr> <tr><td>☐  ATListCriterion (List Criterion)</td></tr> <tr><td>☐  ATPathCriterion (Path Criterion)</td></tr> <tr><td>☐  ATRelativePathCriterion (Relative Path Criterion)</td></tr> <tr><td>☐  ATPortalTypeCriterion (Portal Types Criterion)</td></tr> <tr><td>☐  ATReferenceCriterion (Reference Criterion)</td></tr> <tr><td>☐  ATSelectionCriterion (Selection Criterion)</td></tr> <tr><td>☐  ATSimpleIntCriterion (Simple Int Criterion)</td></tr> <tr><td>☐  ATSimpleStringCriterion (Simple String Criterion)</td></tr> <tr><td>☐  ATSortCriterion (Sort Criterion)</td></tr> <tr><td>☐  Discussion Item (Comment)</td></tr> <tr><td>☐  Document (Page)</td></tr> <tr><td>☐  Event (Event)</td></tr> <tr><td>☐  File (File)</td></tr> <tr><td>☐  Folder (Folder)</td></tr> <tr><td>☐  Image (Image)</td></tr> <tr><td>☐  Link (Link)</td></tr> </tbody> </table>						Type Name	☐  ATBooleanCriterion (Boolean Criterion)	☐  ATCurrentAuthorCriterion (Current Author Criterion)	☐  ATDateCriteria (Friendly Date Criteria)	☐  ATDateRangeCriterion (Date Range Criterion)	☐  ATListCriterion (List Criterion)	☐  ATPathCriterion (Path Criterion)	☐  ATRelativePathCriterion (Relative Path Criterion)	☐  ATPortalTypeCriterion (Portal Types Criterion)	☐  ATReferenceCriterion (Reference Criterion)	☐  ATSelectionCriterion (Selection Criterion)	☐  ATSimpleIntCriterion (Simple Int Criterion)	☐  ATSimpleStringCriterion (Simple String Criterion)	☐  ATSortCriterion (Sort Criterion)	☐  Discussion Item (Comment)	☐  Document (Page)	☐  Event (Event)	☐  File (File)	☐  Folder (Folder)	☐  Image (Image)	☐  Link (Link)
Type Name																										
☐  ATBooleanCriterion (Boolean Criterion)																										
☐  ATCurrentAuthorCriterion (Current Author Criterion)																										
☐  ATDateCriteria (Friendly Date Criteria)																										
☐  ATDateRangeCriterion (Date Range Criterion)																										
☐  ATListCriterion (List Criterion)																										
☐  ATPathCriterion (Path Criterion)																										
☐  ATRelativePathCriterion (Relative Path Criterion)																										
☐  ATPortalTypeCriterion (Portal Types Criterion)																										
☐  ATReferenceCriterion (Reference Criterion)																										
☐  ATSelectionCriterion (Selection Criterion)																										
☐  ATSimpleIntCriterion (Simple Int Criterion)																										
☐  ATSimpleStringCriterion (Simple String Criterion)																										
☐  ATSortCriterion (Sort Criterion)																										
☐  Discussion Item (Comment)																										
☐  Document (Page)																										
☐  Event (Event)																										
☐  File (File)																										
☐  Folder (Folder)																										
☐  Image (Image)																										
☐  Link (Link)																										

Fig. 5.24: Il contenuto del tool `portal_types` visto da ZMI

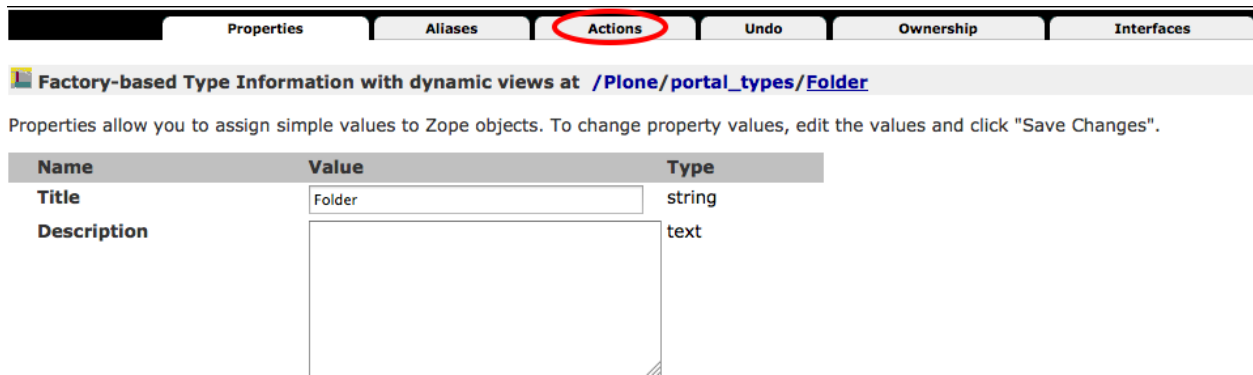


Fig. 5.25: Il link alle azioni di un tipo di contenuto

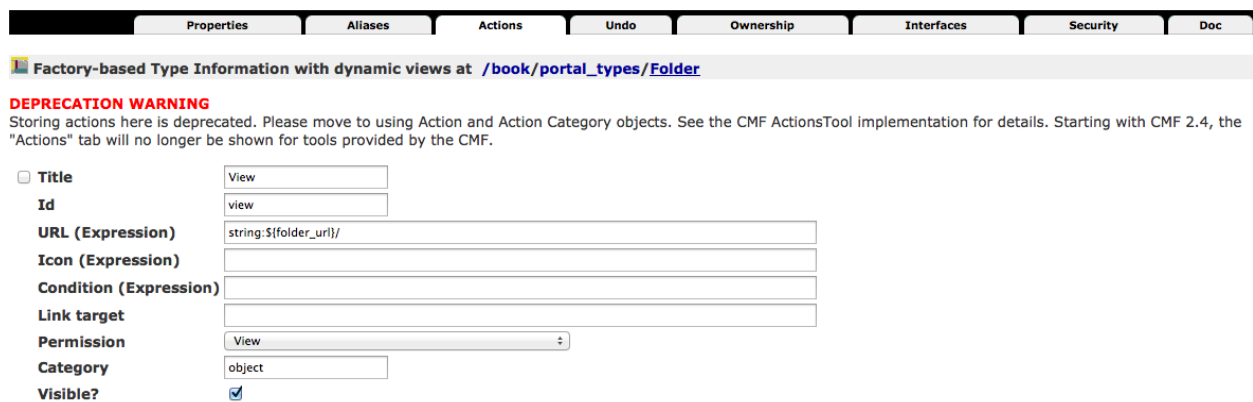


Fig. 5.26: Le azioni di un tipo di contenuto, e il messaggio che avverte che la funzionalità è in dismissione



Fig. 5.27: Le due azioni "Visualizza" e "Modifica", controllate dal portal_types tool

Crediti

Questo libro è distribuito con licenza **Creative Commons**. Attribuzione - Condividi allo stesso modo 3.0 Unported (CC BY-SA 3.0). Tu sei libero:



- di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare quest'opera
- di modificare quest'opera
- di usare quest'opera per fini commerciali

Il sorgente del libro è liberamente scaricabile al seguente indirizzo: <https://github.com/keul/plone-workflow-security-book/>

6.1 Autore

Realizzato da Luca Fabbri.

6.2 Segnalazioni e contatti

Per domande o segnalazioni di errori, non esitate a contattarmi all'indirizzo **luca@keul.it**.

Strettamente per segnalazioni, siete i benvenuti anche ad usare l'issue tracker del progetto: <https://github.com/keul/plone-workflow-security-book/issues>

6.3 Plone e il logo Plone

Plone® e il logo di Plone sono marchi registrati dalla [Fondazione Plone](http://plone.org/foundation/)²⁸.

Plone® and the Plone Logo are registered trademarks of the [Plone Foundation](http://plone.org/foundation/)²⁹.

²⁷ <http://creativecommons.org/licenses/by-sa/3.0/deed.it>

²⁸ <http://plone.org/foundation/>

²⁹ <http://plone.org/foundation/>

Appendice A - tutti i permessi

In questo capitolo verranno descritti *tutti* i permessi Plone che sono utilizzati dal CMS e che può valere la pena conoscere per modificarne i comportamenti.

Verranno anche analizzati alcuni permessi che erano utilizzati in versioni precedenti del CMS e che oggi sembrano aver perso prestigio, ma che può valere la pena conoscere in presenza di vecchi prodotti.

7.1 ATContentTypes Topic: Add ...*Criterion*

Questa grande serie di permessi è storicamente collegata alle **vecchie collezioni**, ancora presenti in Plone ma disabilitate e sostituite con una nuova versione a partire da Plone 4.2.

Se vi ritrovate a gestire versioni di Plone più vecchie di questa o se siete di fronte ad un sito Plone migrato da una vecchia versione (le vecchie collezioni non vengono trasformate nelle nuove versioni nel processo di migrazione) vale la pena continuare la lettura.

Questi permessi controllavano il potere di un utente di poter usare uno specifico criterio. Per fortuna ora non serve più occuparsene.

Per impostazione predefinita: solo *Manager* e *Amministratore del sito* posseggono questi permessi.

7.2 Add portal member

È il permesso che controlla il potere di creare nuovi utenti nel sito.

Oltre al *Manager* e all'*Amministratore del sito* se viene aggiunto anche il ruolo *Anonimo* si abilita la libertà dei visitatori di iscriversi al sito.

Oggi è raramente manipolato manualmente poiché è stato aggiunto un controllo specifico nella sezione “*Sicurezza*” della configurazione del sito.

7.3 Add portal topics

È il permesso che determina il potere di aggiungere le vecchie **Collezioni** nel sito Plone (*Topic* è stato il primo nome del tipo di contenuto, poi diventato *Cercatore* ed infine ha preso il nome odierno).

Vale quanto detto per gli altri *permessi di aggiungibilità dei contenuti* Plone, ma i ruoli che lo posseggono sono solo i seguenti *Manager* e *Amministratore del sito*.

[Configurazione del sito](#) ›

Impostazioni sicurezza

Impostazioni sulla sicurezza in questo sito.

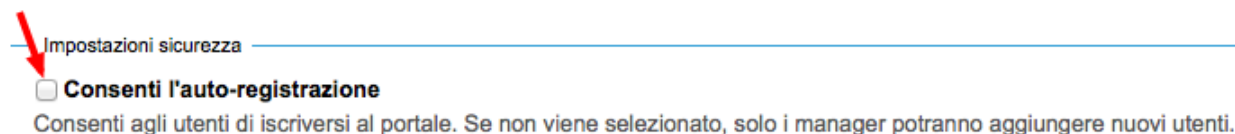


Fig. 7.1: Il controllo nella gestione della “Sicurezza” del sito, che permette di abilitare l’auto-registrazione degli utenti

Ignorate questo permesso se non dovete gestire le vecchie collezioni, poiché il permesso usato ora è “*plone.app.collection: Add Collection*”.

7.4 Allow sendto

È il permesso che permette di utilizzare una vista che permette di inviare un collegamento al documento corrente per e-mail.

Il link a questa pagina è stato disabilitato di default nelle recenti versioni di Plone (in realtà non è una funzionalità così utile e probabilmente il link così esposto era facile preda di crawler malevoli).

È ancora utilizzabile conoscendone l’URL (inserendo `/sendto_form` dopo l’URL di un documento) o riabilitando il link dal `portal_actions` in ZMI.

Il permesso è dato al ruolo *Anonimo*, quindi chiunque può utilizzare questo form.

7.5 Change portal topics

Questo permesso è storicamente associato al permesso di modifica delle *Collezioni*.

Se le *Collezioni* che state gestendo sono quelle introdotte con Plone 4.2, questo stesso permesso è diventato inutile, poiché ora il permesso di riferimento è *Modify portal content*, come per tutti gli altri tipi.

Questo permesso vale ancora la pena essere gestito se avete a che fare con le vecchie collezioni. Vedere quanto detto per i *vecchi permessi di gestione dei criteri*.

- *Manager*
- *Amministratore del sito*
- *Possessore*

7.6 Content rules: Manage rules

Questo permesso è legato alla possibilità di poter gestire le **regole** di Plone sulla cartella corrente.

Per impostazione predefinita: solo *Manager* e *Amministratore del sito* posseggono questi permessi.

7.7 Copy or Move

Questo permesso è legato alle operazioni di **copia** e **taglia**.

Non è nei fatti un permesso molto importante; per impostazione predefinita è infatti dato agli *Anonimi* quindi a chiunque. Il motivo è perché il vero “lavoro” viene fatto con l’operazione di *incolla*, che non è gestito da questo permesso.

Vale la pena gestire questo permesso (magari in un workflow specifico) se per qualche motivo volete rendere impossibile la copia o lo spostamento di un documento. In questi casi il fatto che il permesso sia unificato per copia e taglia a volte crea problemi.

7.8 List portal members

È il permesso che controlla la possibilità di accedere alla lista degli utenti del sito.

Per impostazione predefinita questo permesso è dato ai *Manager*, all’*Amministratore del sito* e al *Collaboratore* (quindi in pratica tutti gli utenti del sito possono vedere gli altri).

Vale la pena modificarlo in presenza di stringenti motivi di privacy.

7.9 List undoable changes

È il permesso che permette di accedere alla pagine per annullare transazioni effettuato dello ZODB: “**Annulla azioni**”. In pratica permette di annullare operazioni svolte nel sito Plone e tornare ad uno stato precedente del sistema.

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Collaboratore*

7.10 Mail forgotten password

Anche se letteralmente la traduzione del permesso è *invio della password per e-mail* (in ricordo dei tempi in cui Plone memorizzava le password in chiaro e le inviava agli utenti), oggi questo permesso controlla il potere di ricevere il link per eseguire il reset della password in caso si sia dimenticata.

Se volete disabilitare la funzionalità (magari perché le password non sono gestite in Plone ma in un LDAP esterno) vale la pena togliere questo permesso a chiunque.

È ovviamente dato agli utenti *Anonimi*.

7.11 Manage Groups

Era il permesso generale per poter gestire i gruppi di Plone.

Il permesso è in gran parte inutilizzato (alcune verifiche di questo sono ancora esistenti in vecchi template di gestione gruppi e utenti, ora deprecati e che verranno rimossi con Plone 4.3).

7.12 Manage users

Vedere quanto detto per “*Manage Groups*”.

7.13 Modify view template

Questo permesso controlla la comparsa del menù “*Vista*” e le funzionalità di poter scegliere una vista per una cartella e un documento come vista predefinita.

C’è un solo permesso per entrambe le funzionalità, non è possibile quindi differenziare i comportamenti.

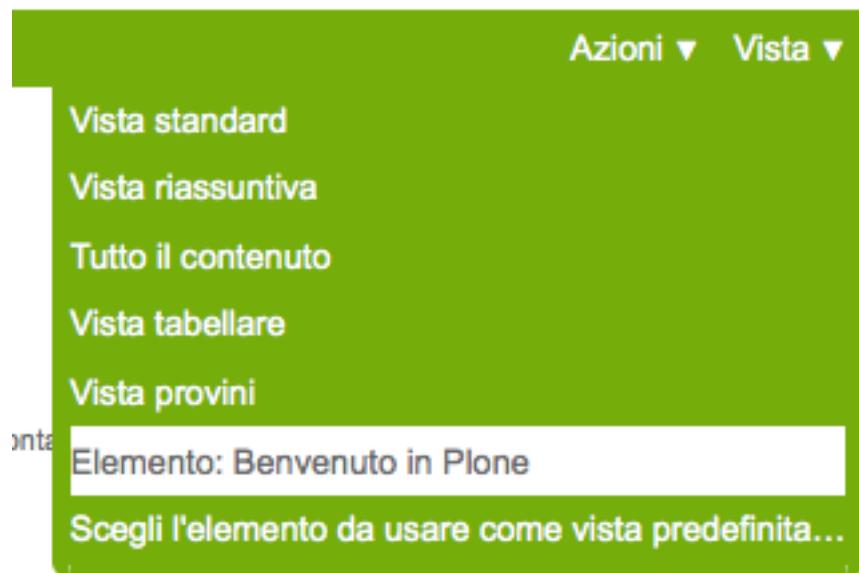


Fig. 7.2: Come si presenta il menù vista

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Possessore*
- *Editor*

7.14 Plone Site Setup: ...

Senza bisogno di scendere in ulteriori dettagli, Plone offre una serie di permessi che servono a gestire in modo puntuale le voci nella *configurazione del sito*.

Per ogni pannello di configurazione c’è un permesso con prefisso “*Plone Site Setup:*”.

Mettiamo solo in una minima evidenza due permessi in particolare:

Plone Site Setup: Overview È il permesso principale, per accedere al pannello di controllo generale.

Plone Site Setup: Users and Groups Questo permesso serve ad accedere alla sezione di gestione gruppi e utenti e pare quindi aver sostituito i vecchi permessi “*Manage groups*” e “*Manage users*”.

Questo permesso permette davvero di gestire utenti e gruppi se assegnato ad altri ruoli (purtroppo, ancora una volta, non è possibile limitarsi ad uno dei due poteri).

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*

È possibile quindi facilmente escludere uno dei pannelli di configurazione di Plone a qualunque modifica, togliendo il permesso associato.

7.15 Portlets: Manage own portlets

È il permesso per gestire le proprie portlet (nella dashboard) e controlla quella voce di menù.

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Collaboratore*

7.16 Portlets: View dashboard

Permesso per poter vedere la propria **dashboard**.

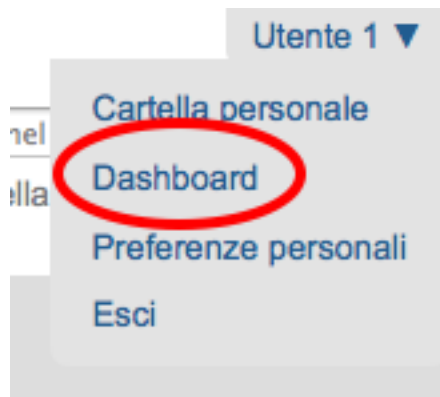


Fig. 7.3: Link alla Dashboard dal menù personale

Rimuovendo questo permesso però il link dal menù personale alla *dashboard* non viene rimosso, ma si ottiene un errore per permessi insufficienti una cliccato (è come se ci fosse la possibilità di vedere la propria dashboard senza poterla modificare, ma al momento la cosa non funziona a dovere).

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*

- *Collaboratore*

In pratica: a tutti gli utenti del sito.

7.17 Set own password

È il permesso associato alla funzionalità di poter cambiare la propria password dalla vista “*Azzera la password*”, accessibile tramite le proprie preferenze personali.

È differente dal permesso “*Mail forgotten password*” perché in questo caso l’utente è autenticato nel sistema. Anche in questo caso però potreste voler togliere questo permesso in casi di fonti dati utente esterne (quali LDAP).

Il permesso è dato a tutti gli utenti *Autenticati*

7.18 Set own properties

È il permesso legato al potere dell’utente di modificare le proprie informazioni personali.

Togliendo questo permesso (assegnato a tutti gli *Autenticati*) l’utente non è più in grado di accedere alla voce “*Preferenze personali*” nel proprio menù di autenticazione.

Purtroppo non è la voce in se a sparire ma si ottiene un errore di permessi insufficienti nel caso si clicchi sulla voce.

7.19 Use mailhost services

Questo permesso è collegato all’utilizzo del sistema di invio e-mail interno di Plone.

Normalmente l’unico punto di contatto tra gli utenti del sito e le e-mail inviate dal sito si hanno per l’invio del reset della password (“*Set own password*”) e per l’invio di un link alla pagina corrente (“*Allow sendto*”). In entrambi i casi Plone verifica due permessi specifici.

Se però un prodotto aggiuntivo, o una vostra funzionalità specifica, dovessero tentare di inviare un messaggio e-mail, questo permesso verrebbe verificato, quindi in questi casi vale la pena verificarne le impostazioni.

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*

7.20 View Groups

È un permesso collegato a vari metodi di basso livello per accedere ai gruppi

È assegnato ai *Manager*, *Amministratori del sito* e *Collaboratori*, quindi a tutti gli utenti autenticati.

Da test eseguiti, se si rimuove il permesso per il *Collaboratore*, gli utenti sono comunque in grado di accedere alla pagina di *Condivisione* e ricercare gruppi.

7.21 View management screens

Vale la pena dire due parole su questo permesso, assegnato solo al *Manager* (e al *Possessore*, ma il proprietario del “sito” è sempre un *Manager*) ma non all'*Amministratore del sito*.

Questo permesso permette agli utenti di entrare in ZMI ed è stato uno dei motivi scatenanti per la creazione del ruolo separato “*Amministratore del sito*”.

7.22 iterate : Check ...

I due permessi *iterate : Check in content* e *iterate : Check out content* sono forniti dal prodotto che si occupa del supporto alla *copia di lavoro*.

Abbiamo già visto alcuni permessi che si occupano del versionamento e che lavorano con questo prodotto (vedere i *permessi relativi a CMFEditions*).

Questi due permessi sono definiti, ma sembrano non usati da nessun componente Plone.

7.23 Reply to item

Questo permesso identifica il potere di poter **commentare**.

Il Plone i commenti sono ora controllati dal prodotto [plone.app.discussion](http://pypi.python.org/pypi/plone.app.discussion)³⁰ e possono anche essere sottoposti a workflow.

Tenete presente che il permesso controlla i commenti *se i commenti sono abilitati* sul contenuto.

Nella pratica infatti il permesso è dato a tutti gli *Autenticati*, ma di base nessun contenuto Plone è di per se automaticamente commentabile.

7.23.1 Review comments

Quando la revisione dei commenti è attivata, chi possiede questo permesso può effettuare la revisione.

Questo comportamento viene innanzi tutto abilitato dal pannello di controllo Plone, alla voce “*Commenti*”.

☐ Abilita la moderazione dei commenti

Se selezionato, i commenti verranno creati in stato 'In attesa' in cui sono non sono visibili pubblicamente. Un utente con il permesso 'Revisiona i commenti' ('Revisore' o 'Manager') possono approvare i commenti per renderli pubblici. Se si vuole abilitare un workflow personalizzato per i commenti, bisogna andare nel pannello di controllo dei tipi.

Fig. 7.4: L'abilitazione della revisione dei commenti, dal pannello “Impostazioni dei commenti”

Per impostazione predefinita i seguenti ruoli posseggono questo permesso:

- *Manager*
- *Amministratore del sito*
- *Revisore*

³⁰ <http://pypi.python.org/pypi/plone.app.discussion>

Il motivo per cui esista un permesso separato per la revisione dei commenti (e non venga usato invece il permesso “*Review portal content*”) è opinabile. Sarebbe stato possibile usare quello stesso permesso, applicato al workflow dei commenti.

7.23.2 plone.portlet.collection: Add collection portlet

Questo permesso è simile al permesso “*Portlets: Manage portlets*”, ma è specifico per poter creare nuove **portlet collezione**.

7.23.3 plone.portlet.static: Add static portlet

Questo permesso è simile al permesso “*Portlets: Manage portlets*”, ma è specifico per poter creare nuove **portlet statiche**.