



UNIVERSITA' DEGLI STUDI DI BARI ALDO MORO

Wordpress Mendeley Plugin documentazione

Autori:

Parisi Davide
Mat.: 613478
(9 settembre 2014)

Cianciaruso Gabriele
Matr.: 623589
(15 Settembre 2015)

Docenti:

Prof. Filippo Lanubile
Dott. Fabio Calefato

Indice

1. Introduzione	3
2. Problemi riscontrati	3
2.1. Organizzazione del codice	3
2.2. Protocollo OAuth2	3
2.3. Caching	4
2.4. Gestione delle chiavi	6
2.4.1. Salvataggio e uso delle chiavi	6
2.5. Passaggio alla versione <i>multisite</i>	7
2.6. Formattazione <i>citeproc</i>	7
2.6.1. Libreria citeproc	7
2.6.2. Formattazione dell'elenco pubblicazioni	7
3. Mendeley REST services	8
3.1. JSON	8
3.2. HTTPS	8
3.3. OAuth 2.0	8
3.4. CORS & AJAX	8
3.5. Servizi REST utilizzati	8
4. Sviluppi futuri	11
4.1. Miglioramenti nell'output	11
4.2. Internazionalizzazione	11
5. Manuale Utente	12
5.1. Installazione del plugin	12
5.2. Attivazione	12
5.2.1. Attivazione Single Site	12
5.2.2. Attivazione Multisite	12
5.3. Gestione delle chiavi e dell'access token	13
5.4. Import dell'elenco pubblicazioni	14
5.5. Inserimento dell'elenco nelle pagine	15

1. Introduzione

Lo scopo del presente lavoro è stato quello di riscrivere e riorganizzare il codice del plugin wordpress che consente di inserire nelle pagine l'elenco completo delle pubblicazioni di cui si è autore.

2. Problemi riscontrati

2.1. Organizzazione del codice

Si è voluto riorganizzare il codice del plugin in chiave Object Oriented, i principali vantaggi di questa soluzione sono:

❖ **Organizzazione Object Oriented del plugin**

L'organizzazione in classi del plugin fornisce una maniera per individuare più rapidamente quali parti di codice ne implementano le funzionalità e per tenere distinte le varie responsabilità. Si distinguono, quindi, due aree:

❖ **Amministrazione del plugin:**

- comprende le classi e gli assets che si occupano di gestire la corretta visualizzazione e funzionamento delle opzioni del plugin

❖ **Area pubblica:**

- comprende le classi e gli assets che si occupano di gestire il funzionamento del plugin

La classe *CollabMendeleyPlugin* si occupa invece del *bootstrapping* del plugin e del coordinamento fra le due aree (admin e public)

2.2. Protocollo OAuth2

Dal 14 aprile 2014 Mendeley ha cambiato il protocollo di accesso alle proprie API passando a OAuth2.

Il nuovo protocollo (Figura 1) introduce alcune novità rispetto alla precedente versione:

- non obbliga il client ad implementare una meccanismo di cifratura (tutte le comunicazioni richiedono HTTPS);
- non si basa su protocolli proprietari (come *FlickrAuth* o *AuthSub* di Google);
- migliore *user experience* per applicazioni native (non *browser-based*) e poten-

- zialmente per tutti i futuri device;
- migliore scalabilità.

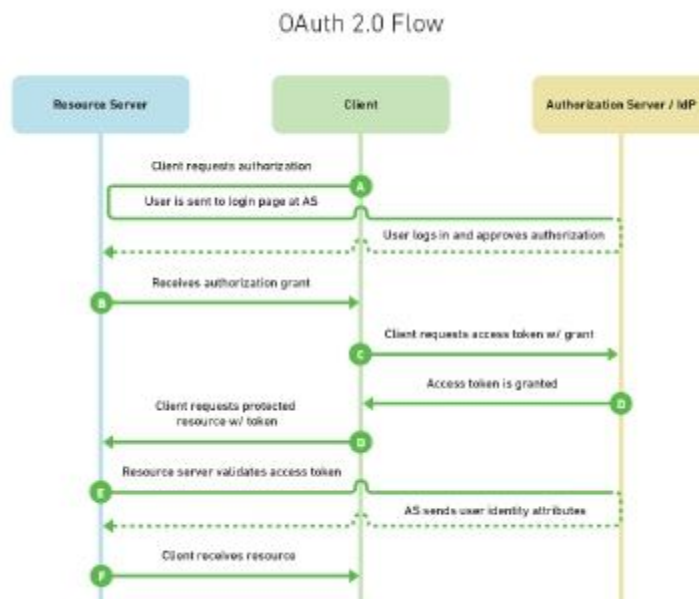


Figura 1: Oauth2 authorization low

Per poter adattare il plugin alla nuova versione del protocollo è stato integrato un client oauth2 php (PHP-OAuth2) che si occupa della corretta comunicazione col server di autenticazione e della corretta gestione dei dati richiesti ed inviati (conversione da json in oggetti/array etc.). La gestione del client oauth2 integrato è avvenuta tramite il gestore di dipendenze per PHP composer.

2.3. Caching

Nella prima versione del plugin realizzato si sono riscontrati notevoli rallentamenti nel caricamento delle pagine in cui compare l'elenco pubblicazioni, i tempi di caricamento risultavano direttamente proporzionali alla quantità di pubblicazioni.

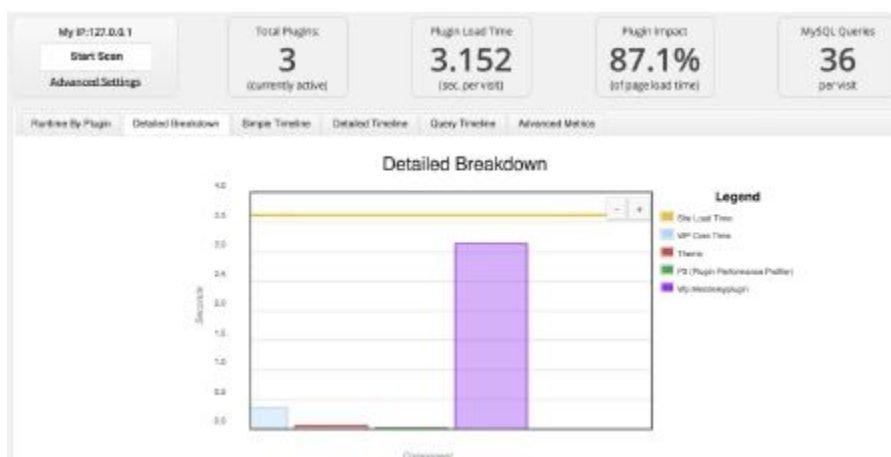


Figura 2: Tempi di caricamento delle pagine e impatto del plugin prima dell'introduzione del caching

Come si può vedere dalla Figura 2 il tempo di caricamento della pagina, misurato con un apposito plugin¹ si aggirava intorno ai tre secondi, ed il plugin è risultato essere il maggior responsabile di tale ritardo.

La causa del rallentamento sta nella comunicazione diretta tra la pagina che ospita l'elenco e le API di mendeley.

Si è resa quindi necessaria l'introduzione di una forma di caching, realizzata mediante l'implementazione di una funzionalità di *import* delle pubblicazioni, il cui elenco viene salvato in un apposito record della tabella wp_options (Figura 3).

Con l'introduzione del caching, l'elenco delle pubblicazioni viene recuperato principalmente dal database, ed in caso di disallineamento tra la cache del database e le proprie pubblicazioni su Mendeley è possibile aggiornare la cache rifacendo l'import (viene riportata la data e l'ora dell'ultima importazione Figura 3). Questo accorgimento ha fatto sì che il tempo di caricamento delle pagine scendesse drasticamente a circa 1/10 del tempo precedente (Figura 4).

Non si è ritenuto necessario creare una tabella di database *ad hoc* per contenere l'elenco pubblicazioni, vista la scarsa complessità della struttura da memorizzare (un semplice elenco) e vista la prassi comune, fra gli sviluppatori wordpress, di evitare la creazione di tali tabelle se non strettamente necessario.



Figura 3: Interfaccia funzioni di amministrazione del plugin

¹P3 - Plugin Performance Pro iler <http://wordpress.org/plugins/p3-profiler/>

La creazione di una tabella di database personalizzata, scarica sullo sviluppatore l'onere di doverla mantenere e gestire, mentre, sfruttare le tabelle preesistenti (in particolar modo la tabella wp_options) con le apposite funzioni di lettura, scrittura e modi ica messe a disposizione da wordpress stesso, rende la gestione delle opzioni del plugin molto più semplice e manutenibile.

2.4. Gestione delle chiavi

Nel protocollo OAuth2, fondamentale è ottenere un access token. L'access token è *short lived*, ossia: ha una scadenza (1 ora). Associato all'access token vi è sempre un refresh token che può essere utilizzato per poter effettuare il refresh del token scaduto senza passare ancora attraverso la procedura di autenticazione mediante login con username e password. Va da sè che in questo quadro operativo, diviene di fondamentale importanza salvare access token e refresh token per poterli utilizzare quando servono.

2.4.1. Salvataggio e uso delle chiavi

Nel plugin il salvataggio delle chiavi viene effettuato in un record della tabella delle opzioni, sotto la *collab-mendeley-plugin* (oggetto JSON).

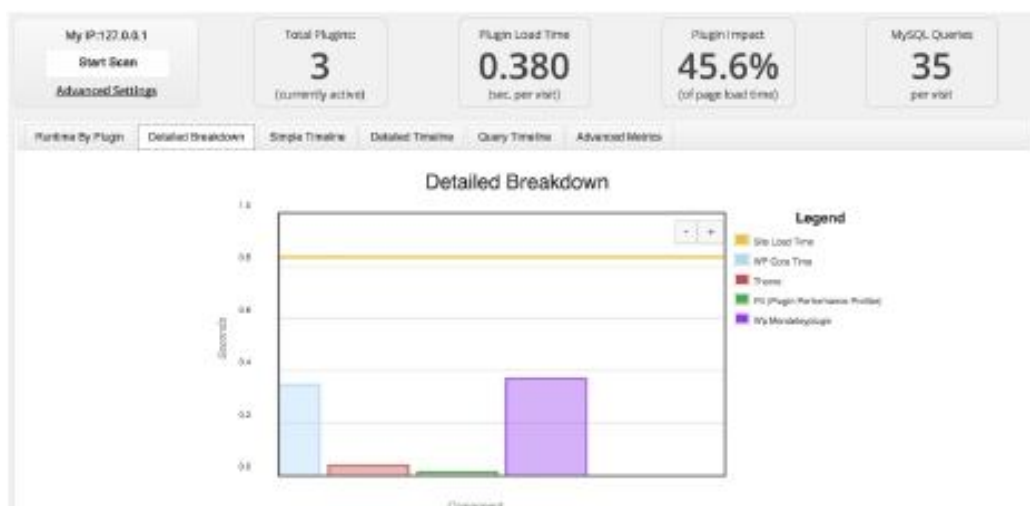


Figura 4: Tempi di caricamento delle pagine e impatto del plugin dopo l'introduzione del caching

2.5. Passaggio alla versione *multisite*

In fase di attivazione il plugin è in grado di distinguere se l'istanza di wordpress in cui è stato attivato è di tipo *single site* o *multisite*, in quest'ultimo caso viene creato un record nella tabella delle opzioni specifica per ogni sito presente nel network.

Tale record conterrà le opzioni di base, inizializzate, per il corretto funzionamento del plugin (*client id/client secret*). Tutte le successive opzioni ed impostazioni saranno salvate nella medesima tabella specifica per il sito in questione.

2.6. Formattazione *citeproc*

La formattazione dell'elenco pubblicazioni è demandata alla libreria *citeproc*, il plugin dispone di una versione riadattata per PHP della libreria.

Tuttavia sono state introdotte delle classi CSS specifiche (contenenti il *pre fix* *collab-mendeley-*) che possono essere sfruttate per personalizzare la formattazione dell'elenco pubblicazioni mediante fogli di stile.

2.6.1. Libreria *citeproc*

CiteProc è il nome col quale si suole individuare una serie di librerie e programmi in grado di produrre bibliografie correttamente formattate. La libreria utilizzata nel progetto è una versione modificata dell'originale PHP (scritta specificatamente per *Drupal*).

2.6.2. Formattazione dell'elenco pubblicazioni

L'elenco delle pubblicazioni viene dapprima suddiviso per tipo (Book, Conference Proceeding, Journal Article etc.) e successivamente ordinato per anno di pubblicazione, dal più recente al più remoto.

La formattazione di ogni entry dell'elenco pubblicazioni avviene in parte grazie all'elaborazione della libreria *citeproc*, cui vengono aggiunte:

- in testa, l'elenco degli autori
- in coda il link DOI all'articolo

3. Mendeley REST services

Il plugin implementato, utilizza i servizi messi a disposizione dell'applicazione Mendeley.

Questi sono stati aggiornati ed ampliati rispetto alla precedente versione del plugin stesso. Oggi vengono utilizzate gli ultimi servizi disponibili.

3.1. JSON

Il format dei dati di risposta delle API sono nel formato JSON. JSON è un formato di dati facilmente leggibile dall'uomo e dalla macchina. Questo formato è utilizzato da numerosi linguaggi di programmazione, tanto che alcuni di questi lo utilizzano a livello native.

3.2. HTTPS

Tutte le Richieste API devono essere inviate usando connessioni protette TLS/SSL. Altri tentativi di connessione con normale protocollo HTTP sono rifiutati dal server.

3.3. OAuth 2.0

Come spiegato nel paragrafo precedente, questo è il protocollo di autenticazione adottato da Mendeley. Maggiori informazioni sono disponibili qui: OAuth 2.0

3.4. CORS & AJAX

Le API Mendeley supportano CORS (Cross Origin Resource Sharing) che abilita la possibilità di utilizzare chiamate AJAX direttamente da Javascript. Maggiori informazioni qui: CORS.

3.5. Servizi REST utilizzati

Le API Mendeley consentono agli sviluppatori di accedere ai dati sia da applicazioni desktop che mobile. I servizi REST consentono di accedere ad ogni risorsa identificandola con un indirizzo dato nella URL. Ci sono numerosi tipi di dati disponibili, ma le principali risorse sono disponibili contattando l'API endpoint:

<https://api.mendeley.com/document>

Documenti Utente creati dagli utenti Mendeley e assegnati alle loro biblioteche(Libraries) o gruppi(Groups)

<https://api.mendeley.com/catalog>

Catalogazione Documenti redatta dalla comunità di Mendeley

<https://api.mendeley.com/files>

Informazioni sui [File](#) allegati associati ai documenti utente (esempio sotto)

GET

/files

V1 - Get files

Implementation Notes

Gets a set of files. At most one of document ID and group ID can be supplied. If neither is supplied, then documents for the logged-in user are returned.

Response Class

Model | Model Schema

```
[
  {
    "id": "",
    "document_id": "",
    "catalog_id": "",
    "filehash": "",
    "mime_type": "",
    "file_name": "",
    "size": 0
  }
]
```

Response Content Type

application/vnd.mendeley-file.1+json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
document_id	<input type="text"/>	The document ID	query	string
catalog_id	<input type="text"/>	The catalog document ID	query	string
group_id	<input type="text"/>	The group ID	query	string
added_since	<input type="text"/>	Returns only files added since this timestamp. Should be supplied in ISO 8601 format	query	string
deleted_since	<input type="text"/>	Returns files deleted since this timestamp. Should be supplied in ISO 8601 format	query	string
include_trashed	<input type="checkbox"/>	Returns files that have been trashed.	query	boolean
limit	<input type="text"/>	The maximum number of items on the page. If not supplied, the default is 20. The largest allowable value is 500.	query	string

Response Messages

HTTP Status Code	Reason	Response Model
200	Files found	
400	Bad request	
403	Forbidden	
404	Group or document not found	

Try it out!

<https://api.mendeley.com/groups>

Mostra i [gruppi](#) di collaborazione per la condivisione di idee e documenti

`https://api.mendeley.com/profiles/me`

Estrale le informazioni relative all'account dell'utente Mendeley autorizzato all'accesso

Servizi di Autenticazione

`AUTHORIZE_ENDPOINT = 'https://api.mendeley.com/oauth/authorize'`

Verifica le autorizzazioni dell'utente Mendeley e rilascio di codice identificativo per OAuth2

`TOKEN_ENDPOINT = 'https://api.mendeley.com/oauth/token';`

Rilascia il TOKEN per l'utente Mendeley autenticato (scambio di user e codice)

Per approfondimenti sugli [oggetti API](#) a disposizione e le [API Reference](#) per scoprire i vantaggi delle risorse disponibili per la programmazione, visita la documentazione ufficiale.

4. Sviluppi futuri

I principali sviluppi futuri e miglioramenti sono:

4.1. Miglioramenti nell'output

Sarebbe interessante migliorare l'output dell'elenco pubblicazioni cercando di dare all'utente una maggiore possibilità di personalizzazione.

Attualmente sono supportati solo gli stili di formattazione *IEEE* ed *ACM*².

4.2. Internazionalizzazione

la base di codice del plugin ha tutte le predisposizioni per l'internazionalizzazione e il supporto per altre lingue, si potrebbe lavorare in tal senso e fornire il supporto ad altre lingue attraverso il file *.pot*

²per una elenco completo degli stili è possibile consultare il repository degli stili di Zotero <https://www.zotero.org/styles>

5. Manuale Utente

5.1. Installazione del plugin

Per installare il plugin nella propria istanza di wordpress (*single site* o *multisite*) è sufficiente decomprimere il pacchetto zip del plugin nella cartella wp-content->plugins.

A questo punto il plugin apparirà nell'elenco dei plugin da attivare e sarà disponibile per l'uso.

5.2. Attivazione

5.2.1. Attivazione Single Site

Per attivare il plugin in una istanza di wordpress *single site*, effettuare il *log-in* come amministratore, entrare nella pagina dei plugin installati, individuare il plugin *Collab Mendeley* e fare click su “activate” (Figura 5)



Figura 5: Attivazione single site

5.2.2. Attivazione Multisite

Per attivare il plugin in una istanza di wordpress *multisite*, effettuare il *log-in* come amministratore del network, entrare nella pagina di amministrazione e visualizzare la pagina dei plugin installati. Individuare il plugin *Collab Mendeley* fare click su “network activate” (Figura 6)

A questo punto il plugin sarà disponibile per tutti i siti del network e la relativa pagina delle impostazioni sarà disponibile ai loro amministratori



Figura 6: Attivazione multisite

5.3. Gestione delle chiavi e dell'access token

Entrare nella pagina impostazioni->Mendeley Settings (Figura 7) ed inserire *client id* e *client secret* precedentemente ottenuti registrandosi e registrando il plugin sul portale degli sviluppatori mendeley. Per facilitare la registrazione del plugin sul portale mendeley si può usufruire dell'apposito URL indicaro come *Redirect URL* in cima alla pagina delle impostazioni (Figura 8).

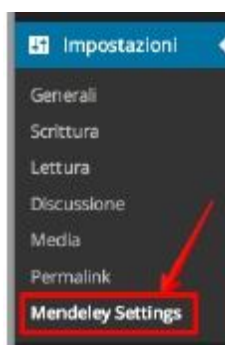


Figura 7: Mendeley Settings

Una volta inserite le chiavi (client id/client secret) bisogna salvarle nel database mediante il pulsante “Save keys to DB” (Figura 9).



Figura 8: Redirect URL da ricopiare

Figura 9: Chiavi salvate nel database

Una volta salvate le chiavi nel database sarà possibile richiedere un access token mediante l'apposito bottone (Figura 10)

Figura 10: Bottone per la richiesta access token

la richiesta di un access token rimanda alla pagina di autenticazione di Mendeley. Una volta autenticati si verrà reindirizzati alla pagina di amministrazione del plugin, l'access token e le informazioni sulla sua scadenza compariranno sulla pagina (Figura 11)

Figura 11: La richiesta access token è andata a buon fine

5.4. Import dell'elenco pubblicazioni

Per importare dal proprio account Mendeley l'elenco delle pubblicazioni di cui si è autore fare click sul bottone Import Publications (Figura 12)

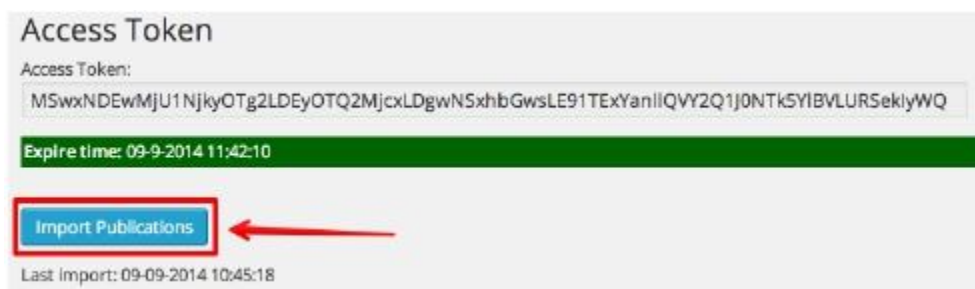


Figura 12: Import delle pubblicazioni nel db

L'elenco delle pubblicazioni verrà salvato in un record della tabella wp-options (utilizzata come cache)

5.5. Inserimento dell'elenco nelle pagine

Per inserire l'elenco delle pubblicazioni in una pagina o post fare click sull'apposito bottone con l'icona di Mendeley sull'editor (Figura 13)

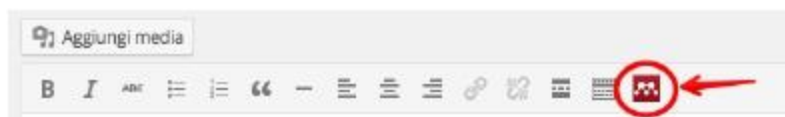


Figura 13: Bottone nell'editor per l'inserimento dello shortcode Mendeley

si aprirà una finestra (Figura 14) in cui sarà possibile impostare:

- Intestazione dell'elenco pubblicazioni
- tag html in cui sarà inclusa l'intestazione
- tag html per le sottosezioni (Book, Conference Proceedings, Journal Article etc.)

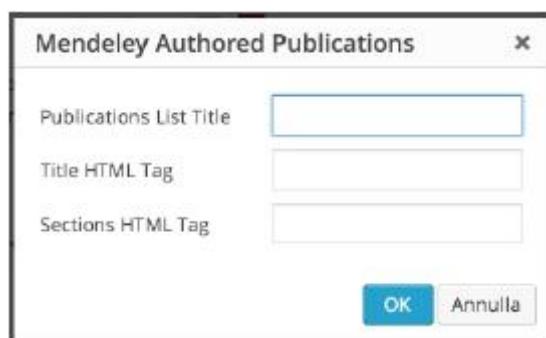


Figura 14: Opzioni per la visualizzazione dell'elenco pubblicazioni

Tutte queste opzioni sono *facoltative*, se non impostate non compromettono la visualizzazione dell'elenco.

Riferimenti bibliografici

[1] Wordpress stack exchange. <http://wordpress.stackexchange.com/>, Settembre 2014.

[2] B, B. *Wordpress 3 Plugin Development Essentials*. Pakt Publishing, 2011.

[3] L, S. *Wordpress for Web Developers*. Apress, 2013.

[4] M. Mendeley developer portal. <http://dev.mendeley.com/>, Settembre 2015.

[5] P, A. Oauth2 simplified.

<https://aaronparecki.com/articles/2012/07/29/1/oauth2-simplified>, Settembre 2015.

[6] S, R. Introduction to oauth (in plain english).

<http://blog.varonis.com/introduction-to-oauth/>, Settembre 2014.

[7] W, B., R, O., T, J. *Professional Wordpress Plugin Development*. Wiley Publishing, Inc., 2011.

[8] W. Wordpress codex. <http://codex.wordpress.org/>, Settembre 2014.