

RDFa plugin

Giulio Bazzanti

Niccoló Biondi

24/07/2018

1 Descrizione del plugin

RDFa plugin é un plugin per WordPress. Questa applicazione si occupa di aggiungere del markup ad una pagina web, in particolare di introdurre della semantica aggiuntiva per poter avere una migliore reindexazione da parte di un motore di ricerca.

L'RDFa é utile da utilizzare poiché un motore di ricerca può rispondere in maniera più accurata a delle query che gli vengono poste sulla base delle proprietà che vengono specificate all'interno della pagina. Questo é proprio il concetto del Web Semantico, ovvero un web in cui i documenti pubblicati sono associati ad informazioni e dati che ne specificano il contesto semantico.

Il plugin si basa sul vocabolario schema.org e fa sí che l'utente possa impostare due proprietà: *typeof* e *property*. La prima serve per specificare il contesto dell'argomento del post WordPress in questione, mentre la seconda specifica il contenuto del testo che l'utente ha selezionato e a cui si vuol aggiungere della semantica.

Il plugin aggiunge due componenti alla barra degli strumenti di WordPress:

- Una list box che serve per la scelta del *typeof* del vocabolario.
- Un bottone che assegna la proprietà che si vuole assegnare a del testo selezionato, aggiungendo, cioè, la semantica.

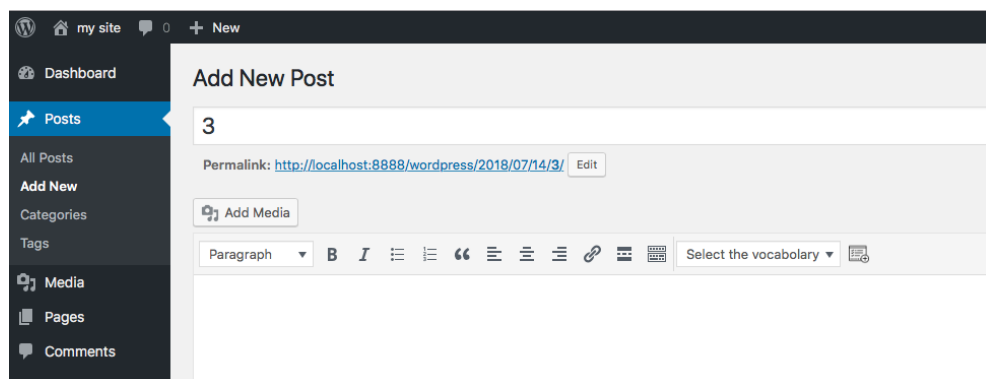


Figure 1: RDFa plugin

2 Funzionamento

L'utente che vuole utilizzare il plugin deve:

1. Selezionare del testo dal post che sta scrivendo per poter poi introdurre l'RDFa.
2. Impostare il `typeof` del vocabolario scegliendolo all'interno della list box.
3. Cliccare sul bottone "Add Semantic" per impostare la proprietà desiderata utilizzando l'apposito pop up che appare.

Una volta che questi passi sono stati completati e il post pubblicato, si sarà aggiunto il mark up dell'RDFa al post.

3 Struttura del plugin

3.1 RDFa-plugin.php

Questo é il file che viene interpretato da WordPress e segue la sua sintassi. Al suo interno, infatti, si trova il costruttore del plugin, che verrà chiamato da WordPress all'inizializzazione della pagina. In particolare le funzionalità di WordPress vengono estese implementando un'azione che permette di aggiungere nella barra degli strumenti i due componenti del plugin descritti in precedenza.

3.2 rdfa-custom-class.js

Questo file rappresenta il comportamento del plugin a seguito dell'interazione dell'utente. Quest'ultimo, infatti, seleziona il *typeof* del vocabolario e ciò comporta che venga caricato il corrispondente file .htm del pop up con le proprietà che caratterizzano quel contesto, che apparirà soltanto quando il bottone verrà cliccato.

L'altro compito principale del file é quello di gestire l'inserimento nell'HTML del post il markup dell'RDFa. Una volta che si é scelto il `typeof` questo inserirà nel post i due attributi `vocab="http://schema.org"` e `typeof="typeof-selezionato"`. In seguito poi il plugin inserirà uno span in corrispondenza del testo selezionato con la specifica della proprietà corretta che gli verrà comunicata dal server, ovvero quella che l'utente ha selezionato dal pop up. Infatti una volta chiusa la finestra, viene inoltrata una richiesta Ajax verso il server per sapere la proprietà selezionata. Il risultato della HTTP request viene inglobato in dei cookies che permettono l'acquisizione dell'informazione. Nello script sono presenti anche le politiche di gestione di altre azioni da parte dell'utente, quali la chiusura del pop up senza la selezione della proprietà oppure l'errato utilizzo del plugin in generale. In tutti questi casi l'utente viene informato dei suoi errori tramite degli alert da parte del browser.

3.3 attribute

In questa cartella é contenuto sia il pop up che si apre dopo il click che avviene sul bottone "Add Semantic", sia il server che gestisce la comunicazione tra la finestra principale, quella di WordPress, e il pop up stesso.

All'interno della cartella sono presenti otto file HTML:

event.htm, **imageObject.htm**, **movie.htm**, **person.htm**, **recipe.htm**, **restaurant.htm**, **sportsEvent.htm** e **videoGame.htm**. Ciascuno di questi rappresenta una delle alternative di tipologia di vocabolario e al suo interno ha tutte le proprietà relative al contesto specificato. Una volta che il bottone Add Semantic viene cliccato si aprirà la pagina HTML corrispondente al tipo di vocabolario che l'utente ha selezionato in precedenza.

Tutti questi file hanno in comune il fatto di avere una stessa struttura, cioè di avere una form HTML con vari input radio, uno per ogni proprietà. Ogni volta che l'utente clicca su un input viene eseguita una funzione *OnClick()*, contenuta nello script JavaScript **funzioni.js** situato nella sottocartella **js**. Questa funzione scatuisce una HTTP request, sempre utilizzando Ajax, al server che fissa il valore della proprietà che l'utente ha selezionato settando il valore dei cookies che, alla chiusura della finestra, saranno utilizzati dal file **rdfa-custom-class.js** come già specificato.

L'altra sottocartella **server** ha al suo interno il file php che corrisponde al server che viene utilizzato nel plugin per gestire la comunicazione tra il pop up e la pagina di WordPress. Il file **actions.php** ha il compito di raccogliere le HTTP request che gli provengono sia dalla finestra di WordPress sia dal pop up e di tener memoria di ciò che gli viene comunicato. All'interno del file, infatti, è presente uno *switch* che invoca una funzione a seconda dell'azione che si vuol fare in un determinato momento. Le azioni sono le seguenti:

modify Questa è l'azione che viene fatta quando la richiesta proviene dal pop up e ha il compito di settare il valore del cookie all'attuale proprietà selezionata attraverso la funzione *modifyProperty()*.

take In questa azione la richiesta proviene dalla finestra principale, ovvero a seguito del click sul bottone "OK" del pop up. Ciò viene interpretato come una avvenuta selezione di una proprietà. In particolare la funzione *takeProperty()* invia dei cookies alla pagina principale con all'interno il valore della proprietà che è stata selezionata.

close In questa azione viene eseguita la funzione *deleteCookie()* che, come si evince dal nome, elimina il cookie resettando, perciò, il valore della proprietà all'interno del server.

Nella cartella sono presenti anche dei file .css e altri file .js che hanno le funzionalità di occuparsi del layout del pop up, i primi, e di includere framework come [Bootstrap](#) e [jQuery](#) che sono stati utilizzati dagli autori nel corso della programmazione del plugin.