

eXtraWay Platform Server - eXtraWay Linked OpenData library



####Not yet implemented/released####

Con l'**introduzione** del **server** di classe **24** si è fatto un più ampio uso delle **librerie dinamiche**, specialmente in forma di **PlugIn**. Per accedere ad essi si fa uso di un **comando XML** generico col quale indicare **parametri** di varia natura ottenendo in risposta un corrispondente **XML**.

La libxwlod è una di questi **PlugIn** ed è finalizzata ad accedere alle **unità informative** di un **archivio** in forma diretta, vale a dire esprimendo un **Linked Open Data**.

Per semplificare la **navigazione** sono stati supportati anche **comandi** per scandire un **albero di nodi**. Si veda la [eXtraWay Tree library](#) per i **dettagli**.

N.B: Per quanto concerne la **numerazione** dei **moduli** si rimanda alla [descrizione della numerazione adottata per il Server](#)

Versioni '1'

Versione 1.0.0

- Prima versione. Sfrutta sia i **link** di tipo `tree`, propri della [eXtraWay Tree library](#), sia **accesso diretto** ad una **resource**;
- Introdotta la gestione degli **identificativi univoci** di **record compositi**;
- Introdotta la gestione delle **descrizioni**¹⁾ dei **nodi** degli **alberi** così che possano essere sfruttati durante la rappresentazione degli **esiti** dei **Linked Open Data**;
- Garantito il corretto **ordine** (come da vocabolario) dei **valori** presenti in **alberi** di tipo **template**;
- Introdotta variante per acquisire la **configurazione** da qualsiasi **plugin** caricato;
- Modificato il processo di **caricamento** degli **allegati** in modo da riuscire nell'intento anche da **server remoto**.

Composizione del comando

I **comandi** per questo **Plug-In** si rifanno alle [regole standard per questo tipo di comandi XML](#)

Unifichiamo quindi nel concetto di **parametro** un **valore** che può essere espresso sia come **attributo** che come **elemento**.

Vediamo comunque che **comandi** sono **disponibili** e quali sono i loro **modificatori**:

Comando	Dettagli
XwLod.doLink	Richiede l'esecuzione di una URI che restituirà un risultato sulla base della sua composizione. Non prevede un parametro <code>arc</code> in quanto l'archivio interessato è parte della URI stessa con l'indicazione dell'azione richiesta. Prevede ovviamente un parametro <code>uri</code> che indichi cosa fare <pre><?xml version="1.0?> <cmd c="0" func="XwLod.doLink" uri="/rubricaIBC/tree/2011/1"/></pre>

Vediamo ora la **composizione** del **parametro** `uri`.

La **prima parte** della **URI** rappresenta l'**id logico** dell'archivio al quale si intende far riferimento.

La **seconda parte** della **URI** rappresenta l'**azione richiesta** su tale archivio.

Dalla **terza parte** in poi si valutano le diverse **parti** che concorrono alla **formazione** del **comando** ed alla sua **esecuzione**.

Alcuni **esempi**:

URI	Azione
/rubricaIBC/tree/ChronoTree/2011/1	Richiede il caricamento del nodo dell'albero ChronoTree, appartenente all'archivio rubricaIBC caratterizzato dall'ID ChronoTree:2011:1.
/rubricaIBC/resource/00001234567	Richiede il caricamento del record dell'archivio rubricaIBC identificabile per mezzo del suo valore univoco 00001234567. Qualora il valore univoco sia rappresentato da una combinazione di componenti esse vanno elencate nella URI nel giusto ordine.

1)

label