



**OGGETTO://**      eXtraWay & LUA: Xw Batteries.

## :// Perché si Usa

---

- :// Triggers, amministrativi ed utente.  
In questo caso i comandi hanno necessariamente un set definito di parametri che non può essere cambiato
- :// Stored Procedures.  
Possono essere definite con la massima libertà espressiva e si possono prestare a molteplici scopi

Nonostante questo, i primi e le seconde sono sostanzialmente identici



## **:// Dove si trova cosa**

---

**La distribuzione delle componenti necessarie per l'esecuzione degli script LUA è la seguente:**

**:// Cartella 'nomearchivio.script'** In questa cartella si trovano gli script di un archivio, vale a dire i file in cui si trovano triggers e stored procedures ad uso esclusivamente di questo archivio.

**:// Cartella 'script' (parallela a bin)** Script LUA generici, questi non si riferiscono ad un archivio specifico ovvero sono applicabili a qualsiasi archivio.

**:// Cartella 'lib' nella precedente** Una raccolta di Lua Batteries non dipendenti da eXtraWay.

## **:// Com'è fatto un file LUA**

---

**Ogni script LUA si compone sostanzialmente di 3 parti:**

**:// Testata dello script**      In questa parte si “includono” tutte le altre *batteries*, che si tratti di script proprietari che batterie di terzi.

**:// Funzioni**                      Tutte le funzioni, che siano a solo utilizzo interno ovvero esportate.

**:// Elenco dei metodi**              Una tabella di metodi messi a disposizione. Essi si distinguono tra metodi accessibili ad altri script Lua e metodi esportati ad uso di eXtraWay. I primi hanno solo un “nome”, gli altri, che sono Trigger e Stored Proc, sono invece corredati da “parametri”.

## :// Testata dello Script

```
local dbg = require 'xw.dbg' ({active = false, key = 'owDev'})

local xw = require 'extraway'

xw.addPackage(".owcfg")
xw.addPackage(".usage")
xw.addPackage(".geocoding")
xw.addPackage(".rdfconverter")
xw.addPackage(".ntconverter")
xw.addPackage(".ttlconverter")
local exp = require 'xw.export'
```

## **:// Testata dello Script**

---

**:// Zona Debug** Richiede il pacchetto 'xw.dbg' ed impostazioni che si riferiscono al contesto Eclipse.

**:// 'xw.addPackage'** Include uno script Lua dell'ambiente eXtraWay. Tutti i metodi in esso presenti sono a disposizione come `<nomescript>.<nomemetodo>`.

La sintassi prevede:

`.nomescript` (script locale)

`$.nomescript` (script globale)

`*.nomescript` (script locale con fallback)

**:// 'require'**

Include una 'lua battery' generica. Le modalità di accesso ai metodi che esso contiene sono analoghe.

## :// Funzioni

```

local preWorker = function(host, paramUrl, fullUrl, database, action, dataSet, mainKey, filterKey, filterValue)
    xw.log.msg("Host:"..host)
    xw.log.msg("paramURL:"..paramUrl)
    xw.log.msg("fullURL:"..fullUrl)
    xw.log.msg("db:"..database)
    local prefix = fullUrl:sub(1, fullUrl:len()-paramUrl:len())
    local ruleRecord = xw.loadFirstRecordByQuery(['[..owcfg.Dataset.subjectIdPath..']='[..dataSet..']')
    if ( not ruleRecord.isValid() ) then
        .....
        return showUsage(usage.openData, "Can't find valid dataSet for "..dataSet)
    end
    local uniqueColumn = ruleRecord.getFirstNodeValue(owcfg.Dataset.uniqueFieldIdPath)
    -- Compose URL
    -- local recordUrl = "http://"..host..prefix.."/"..database.."/"..action.."/"..dataSet.."/"
    local recordUrl = "http://"..host..prefix.."/"..database.."/"..action.."/"
    xw.log.msg("recordUrl: "..recordUrl)
    return worker(action, dataSet, mainKey, filterKey, filterValue, ruleRecord, recordUrl, uniqueColumn)
end

```

## :// Elenco dei Metodi

```
triggers = {  
  iter = {  
    onLoad = iterDecorator,  
    onLoad_signature = {  
      inParams = {"doc"},  
      outParams = {"doc"}  
    },  
    onTitle = iterTitleImprover,  
    onTitle_signature = {  
      inParams = {"doc"},  
      outParams = {"doc"}  
    },  
    onSort = iterTitleImprover,  
    onSort_signature = {  
      inParams = {"doc"},  
      outParams = {"doc"}  
    },  
    onExtend = iterExtender,  
    onExtend_signature = {  
      inParams = {"doc"},  
      outParams = {"doc"}  
    }  
  }  
}
```

## :// Elenco dei Metodi

---

Si compone di una tabella che elenca dei metodi, eventualmente racchiusi in “sub-tables” per ciascuna unità informativa.

Ciascuna dichiarazione si compone di:

*Nome\_di\_esportazione = Nome\_interno\_della\_funzione*

...e per Trigger & Stored Procedures segue la “segnatura”...

*Nome\_di\_esportazione\_signature = {*  
    *inParms = {“p1”, “p2”, “p3”},*  
    *default = {“dv1”, “dv2”, “dv3”},*  
    *outParms = {“op1”, “op2”}*  
*}*

I valori di default sono facoltativi e solitamente mai usati. I parametri in ingresso non sono obbligatori.

## :// Triggers

---

```
<profile type="trigger.user" value="triggers" />  
<profile type="trigger.auth" value="rights" />  
<profile type="trigger.auth.mode" value="weak" />
```

Nel file di configurazione dell'archivio si indicano i trigger utente (lo script), i trigger autoritativi (lo script) e le modalità autoritative.

Weak: In assenza di un trigger per l'evento previsto il server lascia correre, senza opporsi.

Strong: In assenza di un trigger per l'evento previsto il server da errore.

Dettagli in:

[http://wiki.3di.it/doku.php?id=documentazione\\_3di:extraway\\_os:ver\\_stored](http://wiki.3di.it/doku.php?id=documentazione_3di:extraway_os:ver_stored)

## :// Stored Procedure

Possono essere richiamate in molte modalità  
Direttamente da un'applicazione con apposito comando, ovvero tramite  
Web Service.

Il richiamo può avvenire anche in modalità batch, tramite i 'Jobs' e  
'WatchDoc'.

Il comando XML (da console ad esempio) per l'esecuzione di una  
Stored Procedure dichiara lo script che la contiene, il nome del metodo  
e tutti i suoi parametri (per lo meno quelli che si vogliono indicare).

```
...  
xw.addPackage(".geocoding")
```

```
openway = {  
  geocode = geocoding.getGeoCoding,  
  geocode_signature = {  
    inParams = {"address"},  
    outParams = {"result"}  
  }  
}
```

## **:// Stored Procedure**

---

```
<cmd stored=".openway.geocode">  
  <address>Via Speranza 35, San Lazzaro di Savena</address>  
</cmd>
```

...ovvero...

```
<cmd stored=".openway.geocode" address="Via Speranza 35, San  
Lazzaro di Savena" />
```

## :// Stored Procedure: Jobs

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!-- DOCTYPE xwwd_cfg SYSTEM "http://www.3di.it/extraway/jobs_20131015.dtd" -->
<jobs_cfg>
  <!-- Jobs that are done once, on the server Master start
  -->
  <job arc="xdocwaydoc" when="once" val="" label="RV - Finger Print Starter">
    <cmd stored="$.fingerPrint.fingerPrintMaster">
    </cmd>
  </job>
  <job arc="xdocwaydoc" when="every" val="10m" label="RV - Finger Print Agent">
    <cmd stored="$.fingerPrint.fingerPrintAgent">
      <!-- pause>5</pause -->
    </cmd>
  </job>
</jobs_cfg>

```

## :// Stored Procedure: WatchDoc

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!-- DOCTYPE xwwd_cfg SYSTEM "http://www.3di.it/extraway/xwwd_20070308.dtd" -->
<xwwd_cfg>
  <global ms_timeout="500"/> <!-- arcname_separs="-" -->
  <watch dir="../wd" />
  <!-- watch dir="../wdrw"/ -->
  <arc name="pindexer" update="on">
    <!--
      <monitor dir="c:/xw/db/pindexer/pindexer/tutorial"
              when="m15"
              strict="yes"
              subst="yes"/>
    -->
  </arc>
  <!-- arc name="niterback" remove="on" / -->
  <arc name="tm" test="off" /> <!-- Obbrobrio -->

  <arc name="sestra">
    <job ext="pdf" stored=".converter.ingestPdf"/>
  </arc>
  <arc name="niterback">
    <job ext="b2f" stored=".hstored.callB2F"/>
  </arc>
  <arc name="bnbback">
    <job ext="csv" stored=".csvimport.ingestCsv"/>
  </arc>

```