



Xml e Ajax

Matteo Scarpa
845087

Indice

Capitolo 1

XML

1.1 Cos'è un file XML

XML è un *linguaggio di markup* e XML sta per **eXtensible Markup Language**¹ ovvero un linguaggio che consente di definire gli elementi di un documento attraverso meccanismi sintattici. È derivato dallo standard SGML ed è pensato per formattare informazioni ed è nato per uso legato alla rete anche se poi è stato esteso anche a formato per il salvataggio di file.

1.1.1 Storia del XML

XML si prefigge di essere una versione semplificata di **Standard Generalized Markup Language** (conosciuto anche come SGML o ISO 8879) da cui deriva. Il SGML è un metalinguaggio definito come standard ISO² avente lo scopo di definire linguaggi da utilizzare per la stesura di testi destinati ad essere trasmessi ed archiviati con strumenti informatici, ossia per la stesura di documenti in forma leggibile da computer (machine readable form). Da questo linguaggio XML prende il DTD (o Document Type Definition) che è un documento che definisce le componenti ammesse e come devono essere strutturati.

Lo sviluppo dell' XML parte nel 1996 quando si costituì l'XML Working Group all'interno del W3C per poter definire un linguaggio di markup che offrisse maggior libertà nella definizione dei tag rispetto a SGML. Questo viene fatto perchè l'assenza di uno standard abbastanza elastico stava portando i vari sviluppatori di browser e di formati per il web a fare "estensioni" non standard creando caos nella condivisione dei file nel web in quanto solo il loro

¹Vedi [site:w3cxml]

²Vedi [site:isosgml]

prodotto poteva visualizzare correttamente i file. L' XML Working Group ha quindi definito la struttura di base dell' XML e ha portato alla sua prima definizione di uno standard riconosciuto dell XML.

Questo standar, inizialmente pensato per la trasmissione dei dati attraverso il web, ha avuto risvolti inaspettati. Dopo poco tempo ha infatti permesso lo sviluppo di tutta una serie di formati di file indipendenti dalla trasmissione via web, pensati per il semplice salvataggio. Per questo motivo spesso il formato XML viene considerato appartenente al livello di presentazione del modello ISO.³

Tra i vari formati derivati ricordiamo:

- **ODF** formato di documenti OpenSource. Si struttura come una cartella compressa contenente file XML per la definizione del contenuto del documento⁴
- **SVG** formato per le immagini vettoriali. Definisce gli oggetti geometrici attraverso funzioni matematiche. Questo permette di avere immagini "leggibili" da codice e facilmente modificabili con processi automatici, che vanno a leggere il contenuto dell'XML e a modificare i tag di interesse.⁵
- **X3D** formato per gli oggetti tridimensionali e delle scene in cui tali oggetti esistono. Utilizzato da alcuni programmi di grafica 3D per la visualizzazione web⁶

In oltre molti database strutturano i dati in formato XML perchè è più facile elaborarli in tempi costanti e risulta più facilmente adattabile al paradigma del database funzionale.

1.1.2 Struttura

La struttura di un file XML è sempre la stessa:

- Un **header**, solitamente sotto la forma di un tag XML, che indica la versione XML (o derivate) utilizzata per la stesura del documento e la versione della codifica utilizzata nel documento XML

³come è scritto anche nel **tanenbaum:reti** il livello di presentazione come quello di codifica della sintassi dei file per questo motivo viene spesso associato a questo livello

⁴Vedi [**site:isoodf**]

⁵Vedi [**site:w3csvg**]

⁶Vedi [**site:w3dx3**]

- Un **tag di corpo** contenente tutti gli altri tag strutturati ad albero. In oltre, nel tag di corpo c'è la possibilità che sia anche presente la definizione del namespace utilizzato per la definizione dei tag.
- Tutti gli altri tag, che seguono rigide regole di definizione.

In più, per essere correttamente visualizzato o utilizzato deve essere **ben formattato** ovvero

- Ogni etichetta o tag deve essere compreso tra due parentesi angolate (<>) e deve avere il corrispettivo tag di chiusura. Questo può essere realizzato attraverso una coppia di tag (<etichetta></ etichetta>) o un tag unico che si chiude in se stesso (<etichetta/>)
- I tag non possono iniziare con numeri o caratteri speciali
- I tag non possono contenere spazi (solitamente vengono sostituiti con _ o semplicemente non vengono messi)
- I commenti devono essere definiti nel tag di commento (<!-- corpo del commento -->) e possono essere posizionati in ogni punto del documento
- I tag devono essere correttamente annidati (vanno chiusi nell'ordine inverso di come sono stati aperti)
- I tag sono *case sensitive* quindi bisogna fare attenzione quando definisco i tag (esempio: il tag *PianoForte* è diverso dal tag *Pianoforte* e possono essere utilizzati per indicare tag dalla funzione diversa)

1.1.3 Esempio di XML

Un esempio di XML ben formattato è il seguente:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="esempio.css"?>
<lettera xsi:noNamespaceSchemaLocation="esempio.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <mittente>Romeo Rossi</mittente>
  <data>2012-09-28</data>
  <destinatario>Giulietta Verdi</destinatario>
  <oggetto>Riunione operativa</oggetto>
  <forma_cortesia>Egregia Dott.ssa Verdi </forma_cortesia>
  <corpo>
    <paragrafo>come da accordi telefonici ,
```

```

        indico la riunione operativa per il 3
        ottobre 2012 alle ore 10:00.
    Prego di portare tutti i documenti relativi al progetto
</paragrafo>
</corpo>
<chiusura>Cordiali Saluti</chiusura>
<firma>Romeo Rossi
</firma>
</lettera>

```

1.2 XSLT

Assieme all'XML è stato sviluppato un meccanismo per visualizzare direttamente i dati XML dal browser attraverso una trasformazione in un altro documento: XSLT appunto. **EX**tensible **S**tylesheet **L**anguage **T**ransformations permette infatti di rielaborare il documento XML in un nuovo documento leggibile dal browser⁷ o dall'applicazione in particolare. Questo va a produrre un documento XML o in un altro formato derivato dall'XML più facilmente utilizzabile o visualizzabile dall'applicativo.

Un esempio di un applicativo di questo tipo può essere il sito di una banca, che richiede gli aggiornamenti in tempo reale dei dati della borsa. Va a richiedere i dati attraverso un applicativo AJAX al server della borsa e lo rielabora attraverso il suo XSLT in un formato più facilmente visualizzabile dalla pagina web o semplicemente elimina informazioni in eccesso nell'XML per semplificare la successiva elaborazione e visualizzazione.

Questo permette, in oltre, di avere un filtro nei dati visualizzati alleggerendo il javascript dall'oneroso lavoro di parsare il contenuto del file XML.

1.2.1 Esempio di output di XSLT

Questo è un esempio di output ottenuto eseguendo un XSLT sul codice XML precedentemente mostrato

```

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/css" href="esempio.css"?>
<html xmlns:h="http://www.w3.org/1999/xhtml" lang="it"
  xml:lang="it">

```

⁷Vedi [site:w3cxslt]

```

<h:h1>Riunione</h:h1>

<lettera xsi:noNamespaceSchemaLocation="lettera.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.business.org">
<mittente>Romeo Rossi</mittente><h:br />
<data>2012-09-28</data>
<br />
<destinatario>Giulietta Verdi</destinatario>
<p>
<oggetto>Riunione operativa</oggetto>
<forma_cortesia>Egregia Dott.ssa Verdi </forma_cortesia>
<corpo>
<paragrafo>
come da accordi telefonici , indico la riunione operativa
per il 3 ottobre 2012 alle ore 10:00.
Prego di portare tutti i documenti relativi al progetto
<h:br />Nota: per informazioni ulteriori si veda a questo
<h:a href="http://www.unive.it">link</h:a>
</paragrafo>
</corpo>
<chiusura>Cordiali Saluti
</chiusura>
</p>
<firma>Romeo Rossi
</firma>
</lettera>
</html>

```

1.3 Grammatica e definizione degli elementi sintattici

Il formato di file XML per certi punti di vista è molto aperto e permette una grande libertà nei tag pur mantenendo una sintassi stretta e bene definita. Questo comporta, però, una definizione di sintassi unica per la rappresentazione dei documenti che bisogna concordare tra gli utilizzatori di quel determinato tipo di documento XML. Questo ha portato alla necessità di definire delle **grammatiche** ovvero dei documenti che definisco quali sono

i tag XML da utilizzare e come all'interno del documento che si ha intenzione di utilizzare.

Questa tipologia di documenti viene definita in un file (solitamente a parte rispetto all' XML e facilmente reperibile come può essere con un indirizzo internet) che viene inserito sottoforma di link all'interno dell XML in cui vengono inserite le informazioni. Questo documento solitamente esprime dei vincoli su come debba essere strutturato XML:

- Qual'è il tag radice
- Quali e quanti tag sono sotto a un determinato tag
- Quali attributi possono avere i vari tag
- Quali tag non possono mancare

Per questo genere di lavoro sono stati definiti due tipi di file: DTD e il XML Schema

1.4 Document Type Definition

Il **Document Type Definition** (o DTD) permette di descrivere i tag html da utilizzare all'interno di un documento e il loro rapporto reciproco. Questo permette un ampio controllo sul contenuto del documento e semplifica il lavoro di elaborazione di entrambi le parti.

1.5 XML Schema

Successivamente per le definizioni di grammatiche per le applicazioni XML è stato introdotto il linguaggio XML Schema, pensato apposta per questa funzione. Questo nuovo tipo di documenti permette un maggior controllo nella definizione dei documenti Xml attraverso i namespace ed è l'unico linguaggio di definizione del contenuto per XML approvato dalla W3C ([[site:w3cxmlschame](http://www.w3.org/2001/XMLSchema)])

1.5.1 Esempio di XMLS

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="lettera">
<xs:complexType>
```



```

<xs:sequence>
  <xs:element name="mittente" type="xs:string"/>
  <xs:element name="data" type="xs:date"/>
  <xs:element name="destinatario" type="xs:string"/>
  <xs:element name="oggetto" type="xs:string"/>
  <xs:element name="forma_cortesia" type="xs:string"/>
  <xs:element name="corpo" minOccurs="1">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="paragrafo"
          type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="chiusura" type="xs:string"/>
  <xs:element name="firma" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>

</xs:schema>

```

Capitolo 2

AJAX

AJAX è una tecnica di sviluppo software basato sullo scambio dati in background tra il client e il server¹. Il nome è un acronimo che sta per **A**synchronous **J**avascript **A**nd **X**ML spiega anche come è nato.

AJAX nasce come tecnica di scripting che permette lo scambio di dati asincrono tra il client e il server, evitando così la richiesta al server da parte del browser e permettendo all'utente di non aggiornare la pagina visualizzata pur avendo una interazione col server. Nonostante il nome suggerisca l'uso di due tecnologie (Javascript e XML per l'appunto) questo non vuole dire che AJAX si limita solo a queste due ma utilizza anche altre tecnologie quali JSON, JScript e VBScript.

Questa tecnica di sviluppo è largamente utilizzata nelle applicazioni web con poca o nessuna richiesta di refresh della pagina ma comunque la necessità di inviare e/o ricevere dati dal server. Spesso infatti si hanno applicazioni che recuperano in tempo reale le informazioni e "aggirano" la pagina con i nuovi dati senza dover riscrivere tutto oppure nel caso di pagine particolarmente pesanti di caricare la struttura HTML della pagina e appena sono pronti visualizza i dati richiesti dalla pagina.²

Succesivamente è stato definito nel **flanaganjavascript** come

Ora (AJAX) è semplicemente un termine utile a identificare qualsiasi architettura di una web application basata su script di richieste HTTP.

Questo dimostra come la tecnica risulta essere efficace ed è stato necessario estenderla ad altri linguaggi di scripting in quanto solo il javascript non è sufficiente.

¹Richiede infatti l'applicazione di un paradigma cliente servernte

²Vedi [**flanaganjavascript**]

2.1 XMLHttp

Per realizzare un applicativo AJAX è necessario utilizzare un API o una libreria come XMLHttp che ti permetta di aggiornare la pagina dinamicamente richiedendo i dati al server sotto forma di file XML.³ La maggior parte dei browser supportano l'API XMLHttp⁴ anche se alcune versioni di Internet Explorer non lo supportano e usano altre API. Questa libreria gestisce tutte una serie di funzionalità utili per l'elaborazione delle richieste da inviare al server e per elaborare le risposte ottenute.

2.2 Problemi e vantaggi derivanti da AJAX

Come tutte le cose AJAX dispone di vantaggi e di svantaggi.

Per quanto riguarda i vantaggi questi sono molteplici:

- Permette di caricare la struttura della pagina senza i dati. Questo porta ad accelerare i tempi di caricamento della pagina (componente grafica) mentre i dati vengono elaborati in background permettendo all'utente di continuare l'esperienza di navigazione della pagina
- Permette di avere i dati aggiornati in tempo reale. Questo permette a dei siti, come ad esempio quelli delle banche, di avere indici o altri dati aggiornati in tempo reale
- Permette di riciclare le pagine e di non dover caricare ulteriori pagine se cambiano solo i dati. Infatti se io ho una pagina (intesa come struttura html/php) a cui devo solo refreshare i dati con una versione più aggiornata uso uno script che lo esegue l'aggiornamento dei dati piuttosto che aggiornare l'intera pagina.⁵

Per quanto riguarda gli svantaggi sono, per fortuna, pochi e sistemabili seppur parzialmente:

³Ovviamente dal nome è facile capire che questa API permette di lavorare senza problemi soprattutto se il server risponde con file XML anche se supporta il JSON e i testi semplici

⁴vedi **site:xmlhttpcompatibilita**

⁵Un esempio di questo tipo di pagina si può trovare nel progetto del gruppo Veneziani DOC (Matteo Scarpa, Andrea Frison, Margherita Rossi) per l'esame di Interazione di Uomo Macchina di CaFoscari <http://venezianidoc.github.io/Interazione-Uomo-Macchina/> in cui un javascript aggiorna i dati che comportano modifiche all'ambiente circostanze in base all'anno selezionato

- Mancanza di gestione delle pagine da parte della cronologia. La richiesta AJAX non viene registrata da browser direttamente e quindi rende l'utilizzo della cronologia inutile.
- Necessitano di script per funzionare. Questo implica che un user⁶ supporti il linguaggio di scripting e che lo abbia abilitato, il che non è sempre vero o possibile.

Questo ha portato a sviluppare nel relativamente nuovo standard HTML5 la possibilità di modificare la cronologia del browser⁷ per poter supportare la pagina generata attraverso AJAX. Questo viene infatti fatto perchè i comandi e le richieste degli script possano aggiungere i propri "eventi" all'interno della cronologia del browser ottenendo così coerenza tra la cronologia e le pagine navigate anche se soltanto aggiornate in parte.

⁶User è inteso in questo caso come browser/applicativo client richiedente le informazioni

⁷vedi [site:w3html5] per approfondimento

Riferimenti bibliografici