



DTD

Document Type Definition



Per què cal un esquema en XML? L'origen i la necessitat dels DTD

Quan es treballa amb documents **XML**, una de les característiques fonamentals d'aquest llenguatge és que no imposa cap estructura fixa als documents. Això vol dir que **qualsevol persona pot escriure un XML com vulgui**, sempre que compleixi la sintaxi bàsica del llenguatge. Per exemple, podríem tenir dos documents XML que representen una biblioteca, però amb estructures diferents:

Exemple 1: Un XML amb una estructura determinada

```
xml

<llibreria>
  <llibre>
    <títol>1984</títol>
    <autor>George Orwell</autor>
    <any>1949</any>
  </llibre>
</llibreria>
```

Exemple 2: Un XML amb una altra estructura

```
xml

<llibreria>
  <volum>
    <nom>Brave New World</nom>
    <escriptor>Aldous Huxley</escriptor>
    <publicació>1932</publicació>
  </volum>
</llibreria>
```

Com veiem, Ambdós XML descriuen una biblioteca, però fan servir noms de tags diferents (**llibre** vs. **volum**, **títol** vs. **nom**, **autor** vs. **escriptor**, etc.). Això pot generar **problemes d'inconsistència** quan sistemes diferents han d'intercanviar dades. Per tant, **com ens assegurem que totes les persones o sistemes segueixen el mateix patró quan creen documents XML?**

Aquí és on entra la necessitat d'un **esquema**, i en particular, un **DTD (Document Type Definition)**.

I com es defineix un DTD?

Aquí, el DTD especifica que:

- Un element **<llibreria>** ha de contenir un o més **<llibre>**.
- Un **<llibre>** ha de contenir exactament un **<títol>**, **<autor>** i **<any>**.
- Els elements **<títol>**, **<autor>** i **<any>** han de ser **text pur (#PCDATA, Parsed Character Data)**.

Un DTD pot estar **incrustat dins del mateix document XML o en un fitxer extern**.

```
xml

<!DOCTYPE llibreria [
  <!ELEMENT llibreria (llibre+)>
  <!ELEMENT llibre (títol, autor, any)>
  <!ELEMENT títol (#PCDATA)>
  <!ELEMENT autor (#PCDATA)>
  <!ELEMENT any (#PCDATA)>
]>

<llibreria>
  <llibre>
    <títol>1984</títol>
    <autor>George Orwell</autor>
    <any>1949</any>
  </llibre>
```

En VSCode, cerqueu i
instal·leu l'extensió:
XML de Red Hat

Revisa el XML i el compara amb el
DTD automàticament.

Si detecta qualsevol incidència, et
marcarà el problema al terminal

The screenshot shows the VS Code marketplace page for the 'XML' extension by Red Hat. The extension is marked as a 'Preview' and has a rating of 5 stars. It has been downloaded 7,672,516 times. The page includes tabs for 'DETAILS', 'FEATURES', and 'CHANGELOG'. The 'Installation' section shows the identifier 'redhat.vscode-xml', version '0.28.0', last updated on '2025-03-05' at '23:35:09', and a size of '69.5 MB'. The 'Marketplace' section shows it was published on '2018-09-25'. The 'PROBLEMS' tab is active, showing a single error in the file 'l1libreria.xml' at line 3, column 6: 'The content of element type "llibres" must match "(Empty)". xml(MSG_CONTENT_INVALID)'. The error message is displayed in the terminal area at the bottom.

XML Preview

Red Hat redhat.com | 7,672,516 | ★★★★★

XML Language Support by Red Hat

[Disable](#) [Uninstall](#) [Switch to Pre-Release Version](#) ✓ A

[DETAILS](#) [FEATURES](#) [CHANGELOG](#)

XML Language Support by Red Hat

[CHAT](#) [ON GITTER](#)

[VS MARKETPLACE](#) [RATE LIMITED BY UPSTREAM SERVICE](#)

[VS MARKETPLACE](#) [RATE LIMITED BY UPSTREAM SERVICE](#)

[ESLINT](#) [PASSING](#)

Installation

Identifier	redhat.vscode-xml
Version	0.28.0
Last Updated	2025-03-05, 23:35:09
Size	69.5 MB

Marketplace

Published 2018-09-25,

[PROBLEMS](#) 1 [OUTPUT](#) [DEBUG CONSOLE](#) [TERMINAL](#) ... [Filter \(e.g. text, **/*.ts, !...\)](#) [Copy](#) [List](#) [Up](#) [Down](#)

l1libreria.xml htdocs • BEME3\PT-4 DTD\1 1

⊗ The content of element type "llibres" must match "(Empty)". xml(MSG_CONTENT_INVALID) [Ln 3, Col 6]

Primer, definim l'entitat autor en el

```
<!ENTITY autor "Jordi Pérez">
```

Dins del document XML ens podem referir a aquesta entitat fent servir el nom de l'entitat precedit per un **ampersand** (&) i acabar per un punt i coma (;). Així, en qualsevol lloc on vulguis usar el nom "Jordi Pérez", simplement inclou `&autor;`.

Per això, si vols inserir literalment un ampersand dins del text en XML, has de fer servir l'entitat reservada, que és: `&`:
`<titol>XML & DTD</titol>`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE llibres [
  <!ELEMENT llibres (llibre+)>
  <!ELEMENT llibre (titol, autor)>
  <!ELEMENT titol (#PCDATA)>
  <!ELEMENT autor (#PCDATA)>
  <!ENTITY autor "Jordi Pérez">
]>
<llibres>
  <llibre>
    <titol>Aprenentatge XML</titol>
    <autor>&autor;</autor>
  </llibre>
  <llibre>
    <titol>Exemples Pràctics</titol>
    <autor>&autor;</autor>
  </llibre>
</llibres>
```

ID

l'XML, els atributs de tipus **ID** s'utilitzen per identificar únicament elements dins d'un document XML. Un atribut **ID** ha de tenir un valor únic dins de tot el document XML, de manera que pot ser utilitzat com a referència única per altres elements.

```
<llibres>
  <llibre id="llibre1">
    <titol>Don Quijote de la Mancha</titol>
    <autor>Miguel de Cervantes</autor>
  </llibre>
  <llibre id="llibre2">
    <titol>La vida es somni</titol>
    <autor>Calderón de la Barca</autor>
  </llibre>
</llibres>
```

```
<!ELEMENT llibre (titol, autor)>
<!ATTLIST llibre
  id ID #REQUIRED>
<!ELEMENT titol (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
```

ID

Imagina un sistema de biblioteques on cada llibre té un ID únic i cada prestatge també té un ID únic. Pots fer un document XML que mostri les relacions entre prestatges i llibres així:

Amb aquesta estructura de XML que tens a la dreta, pots emprar XPath per accedir a llibres específics o prestatges de manera eficient.

Per exemple, per trobar tots els llibres del prestatge amb `id="prestatge1"`, podries utilitzar una expressió XPath així:

```
//prestatge[@id='prestatge1']/llibre
```

Aquesta expressió XPath retornaria tots els elements `<llibre>` dins del prestatge amb `id="prestatge1"`.

L'ús d'atributs ID en XML facilita aquestes operacions i fa que l'accés a les dades siguin més eficient i estructurat. És a dir, quan dissenyeu un XML heu de tenir aquestes coses en ment

```
<!DOCTYPE biblioteca [  
<!ELEMENT biblioteca (prestatge*)>  
<!ELEMENT prestatge (llibre*)>  
<!ATTLIST prestatge id ID #REQUIRED>  
<!ELEMENT llibre (titol, autor)>  
<!ATTLIST llibre id ID #REQUIRED prestatgeID IDREF #REQUIRED>  
<!ELEMENT titol (#PCDATA)>  
<!ELEMENT autor (#PCDATA)>  
>  
<biblioteca>  
  <prestatge id="prestatge1">  
    <llibre id="llibre1" prestatgeID="prestatge1">  
      <titol>Don Quijote de la Mancha</titol>  
      <autor>Miguel de Cervantes</autor>  
    </llibre>  
    <llibre id="llibre2" prestatgeID="prestatge1">  
      <titol>La vida és somni</titol>  
      <autor>Calderón de la Barca</autor>  
    </llibre>  
  </prestatge>  
  <prestatge id="prestatge2">  
    <llibre id="llibre3" prestatgeID="prestatge2">  
      <titol>El Quadern Gris</titol>  
      <autor>Josep Pla</autor>  
    </llibre>  
  </prestatge>  
</biblioteca>
```


IDREF

L'atribut **IDREF** es fa servir per referenciar un sol **ID**. Indica que l'atribut només pot contenir el valor d'un **ID** existent en el document.

En aquest exemple, l'atribut **prestatgeID** dins de cada element **<llibre>** fa referència a l'**ID** del prestatge on es troba el llibre. El **prestatgeID="prestatge1"** indica que els llibres **llibre1** i **llibre2** es troben dins del prestatge amb **id="prestatge1"**.

```
<!ELEMENT prestatge (llibre*)>
<!ATTLIST prestatge id ID #REQUIRED>
<!ELEMENT llibre (titol, autor)>
<!ATTLIST llibre id ID #REQUIRED prestatgeID IDREF #REQUIRED>
<!ELEMENT titol (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
```

```
<biblioteca>
  <prestatge id="prestatge1">
    <llibre id="llibre1" prestatgeID="prestatge1">
      <titol>Don Quijote de la Mancha</titol>
      <autor>Miguel de Cervantes</autor>
    </llibre>
    <llibre id="llibre2" prestatgeID="prestatge1">
      <titol>La vida és somni</titol>
      <autor>Calderón de la Barca</autor>
    </llibre>
  </prestatge>
  <prestatge id="prestatge2">
    <llibre id="llibre3" prestatgeID="prestatge2">
      <titol>El Quadern Gris</titol>
      <autor>Josep Pla</autor>
    </llibre>
  </prestatge>
</biblioteca>
```


IDREFs

En aquest exemple, l'atribut `prestatgeIDs` dins de l'element `<llibre>` `llibre1` refereix a múltiples `ID` (`prestatge1` i `prestatge2`), el que indica que el llibre `llibre1` pot estar associat amb més d'un prestatge.

Unicitat garantida: Quan defineixes un atribut com a `ID`, el parser XML s'assegura que cada valor d'aquest atribut és únic dins de tot el document. Això vol dir que no pots tenir dos elements amb el mateix valor d'`ID`.

Referències creuades: Els atributs `ID` poden ser referenciats per altres atributs com `IDREF` o `IDREFS`, que permeten establir relacions clares entre elements. Això és útil per crear estructures complexes i interconnectades.

```
<!ELEMENT biblioteca (prestatge*, llibre*)>
<!ELEMENT prestatge (llibre*)>
<!ATTLIST prestatge id ID #REQUIRED>
<!ELEMENT llibre (titol, autor)>
<!ATTLIST llibre id ID #REQUIRED prestatgeIDs IDREFS #IMPLIED>
<!ELEMENT titol (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
```

```
<biblioteca>
  <prestatge id="prestatge1">
    <llibre id="llibre1" prestatgeIDs="prestatge1 prestatge2">
      <titol>Don Quijote de la Mancha</titol>
      <autor>Miguel de Cervantes</autor>
    </llibre>
  </prestatge>
  <prestatge id="prestatge2">
    <llibre id="llibre2" prestatgeIDs="prestatge2">
      <titol>El Quadern Gris</titol>
      <autor>Josep Pla</autor>
    </llibre>
  </prestatge>
</biblioteca>
```

IDREFs

En aquest exemple, l'atribut `ref` de l'element `<ingredient>` fa referència als `ID` dels elements `<recepta>`. Això permet establir una relació clara entre cada ingredient i la recepta corresponent.

IDREFS permet especificar una llista de valors `ID` en el mateix atribut. Això és útil quan un atribut necessita referenciar múltiples elements amb `ID` diferents.

Exemple de definició d'atribut **IDREFS**

```
<!ATTLIST recepta id ID #REQUIRED>  
<!ATTLIST ingredient ref IDREF #REQUIRED>
```

```
<llibre-cuina>  
  <receptes>  
    <recepta id="recepta1">Patates fregides</recepta>  
    <recepta id="recepta2">Patates bullides</recepta>  
  </receptes>  
  <ingredients>  
    <ingredient ref="recepta1">Oli</ingredient>  
    <ingredient ref="recepta2">Aigua</ingredient>  
  </ingredients>  
</llibre-cuina>
```

```
<!ATTLIST ingredient ref IDREFS #REQUIRED>
```

```
<ingredient ref="recepta1 recepta2">Patates</ingredient>
```