

# Intro to eXist-db and Applet XQuery

Angelo Mario Del Grosso (ILC-CNR, Pisa)  
[angelomario.delgrossos@cnr.it](mailto:angelomario.delgrossos@cnr.it)

FeDHLab

# Seminar Outline

- ✓ Intro to eXist-db
- ✓ Intro to XML-based languages
  - XPath
  - XQuery
  - XSL-T
- ✓ Intro to Web Applet with HTML templating package
  - General Architecture
  - File Scaffolding with YEOMAN
  - Applet Structure and Main Resources
  - Step-by-Step Demo Web Applet Implementation
  - Final Review of the Web Applet
- ✓ eXist-db Web Applet Examples



# eXist-db (<http://exist-db.org/>)

The screenshot shows the homepage of the eXist-db website. At the top, there is a navigation bar with the logo "existdb", "About", "Support", "Community", and social media links for GitHub, LinkedIn, Twitter, Google+, and YouTube. The main content area features a large image of a red apple with a green leaf, partially sliced and wrapped in a spiral of various fruit slices (orange, yellow, green). To the left of the image, the text "Vitamins for your Applications" is displayed in a large, serif font, followed by a subtitle "Try the all-in-one solution for application building." Below this, there is an "open source initiative" logo consisting of a stylized 'O' with a 'R' inside, and the text "open source initiative". At the bottom left, it says "eXist-db is Open Source Software licensed under the LGPL". A blue button at the bottom center says "Download eXist-db".

Vitamins for  
your Applications

Try the all-in-one solution for application  
building.

open source  
initiative

eXist-db is Open Source Software licensed under the LGPL

Download eXist-db

# eXist-db: What is

**eXist is an open source piece of software written in Java that is freely available in both source code and binary form**

- A **NoSQL** document database for **XML** and binary (including text)
- A **web server** for consuming and serving documents
- A document **search engine**
- A web **application platform**
- An **Integration** platform
- An embeddable set of **libraries** for use in your own applications

# Important Feature

eXist makes use of a standardized query language developed by the  
W3C: XML Query Language (**XQuery**)

**XQuery** is also a functional programming, designed primarily as a query  
language but in eXist you can build **entire applications** in just XQuery

# Installing eXist-db

Releases / eXist-6.2.0

## eXist-db 6.2.0 Latest

 adamretter released this Feb 4, 2023 · 1138 commits to develop since this release

- Release Notes: <https://exist-db.org/exist/apps/wiki/blogs/eXist/eXistdb620>
- Maven Central: <https://search.maven.org/search?q=g:org.exist-db>

### ▼ Assets 6

 eXist-db-6.2.0.dmg

 exist-distribution-6.2.0-unix.tar.bz2

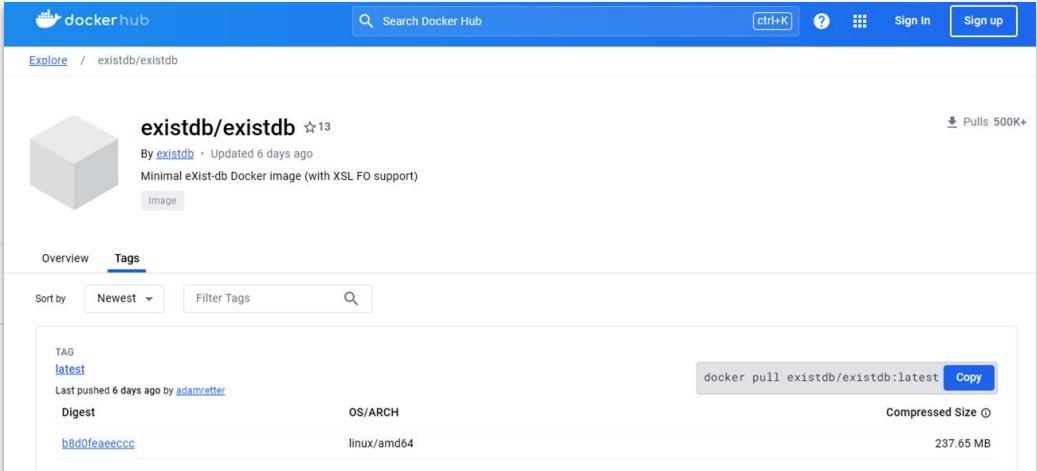
 exist-distribution-6.2.0-win.zip

 exist-installer-6.2.0.jar

 Source code (zip)

 Source code (tar.gz)

  10 people reacted



existdb/existdb ☆ 13  
By [existdb](#) · Updated 6 days ago  
Minimal eXist-db Docker image (with XSL FO support)  
[Image](#)

Overview Tags

Sort by Newest Filter Tags

TAG	Digest	OS/ARCH	Compressed Size
latest	b8d0feaeeccc	linux/amd64	237.65 MB

`docker pull existdb/existdb:latest` [Copy](#)

services:  
exist:  
image:  
**existdb/existdb:releas  
e**  
container\_name:  
**exist-studenti  
ports:**  
- 48002:8080  
- 48004:8443

# Running eXist - db

```
drwxr-xr-x 4 angelodel80 angelodel80 4096 Feb  6 2023 .
drwxr-xr-x 8 angelodel80 angelodel80 4096 Jan 21 2022 ..
drwxr-xr-x 3 angelodel80 angelodel80 4096 Feb  6 2023 backup
-rw xr-xr-x 1 angelodel80 angelodel80 4116 Dec 14 2021 backup.sh
-rw xr-xr-x 1 angelodel80 angelodel80 4116 Dec 14 2021 client.sh
drwxr-xr-x 2 angelodel80 angelodel80 4096 Feb  6 2023 export
-rw xr-xr-x 1 angelodel80 angelodel80 4120 Dec 14 2021 export-gui.sh
-rw xr-xr-x 1 angelodel80 angelodel80 4116 Dec 14 2021 export.sh
-rw xr-xr-x 1 angelodel80 angelodel80 4119 Dec 14 2021 jmxclient.sh
-rw xr-xr-x 1 angelodel80 angelodel80 67316 May 16 2021 jsvc-linux-x86_64
-rw xr-xr-x 1 angelodel80 angelodel80 88848 May 16 2021 jsvc-mac-x86_64
-rw xr-xr-x 1 angelodel80 angelodel80 4118 Dec 14 2021 launcher.sh
-rw xr-xr-x 1 angelodel80 angelodel80 4118 Dec 14 2021 shutdown.sh
-rw xr-xr-x 1 angelodel80 angelodel80 4117 Dec 14 2021 startup.sh
```

```
angelodel80@LAPTOP-V8V3MLG0:~/eXist-db/bin$ ./startup.sh
```

```
[main] INFO (JettyStart.java [run]:287) - -----
[main] INFO (JettyStart.java [lifeCycleStarted]:643) - Jetty server started.
[main] INFO (JettyStart.java [run]:287) - -----
```

```
[main] INFO (JettyStart.java [run]:288) - Server has started, listening on:
[main] INFO (JettyStart.java [run]:290) - http://127.0.1.1:8080/
[main] INFO (JettyStart.java [run]:290) - https://127.0.1.1:8443/
[main] INFO (JettyStart.java [run]:293) - Configured contexts:
[main] INFO (JettyStart.java [run]:299) - / (eXist-db portal)
[main] INFO (JettyStart.java [run]:315) - /iprange (IPrange filter)
[main] INFO (JettyStart.java [run]:299) - /exist (eXist XML Database)
[main] INFO (JettyStart.java [run]:315) - /exist/iprange (IPrange filter)
[main] INFO (JettyStart.java [run]:323) - -----
```

# Dashboard

The screenshot shows the eXist-db dashboard interface. At the top, there's a blue header bar with the word "Launcher" and a user profile icon. Below the header, the main content area displays several application tiles:

- eXistdb Version 5.3.1**: A central tile featuring the eXistdb logo and version information.
- Annotations**: Represented by a yellow 'greater than' symbol icon.
- Bellini Digital Correspondence**: Represented by a blue and grey dot cluster icon.
- eXide**: Represented by the eXide logo.
- eXist-db Demo Apps**: Represented by a circular icon containing three user icons and a dot cluster.
- eXist-db Documentation**: Represented by an open book icon.
- Monex**: Represented by a circular icon containing user and gear icons.
- My amazing nedofiano application**: Represented by a blue and grey dot cluster icon.
- Postille - Bianconi Vanessa**: Represented by a blue and grey dot cluster icon.
- Progetto Tesi Erika Deboni 578063**: Represented by a blue and grey dot cluster icon.
- Stanford Natural Language Processing**: Represented by the Stanford NLP logo.
- TEI Publisher**: Represented by a yellow 'greater than' symbol icon.
- XQuery Function Documentation**: Represented by an open book icon.

# Documentation

The screenshot shows the eXist-db Documentation homepage. At the top, there's a header bar with the URL <http://localhost:8080/exist/apps/doc/>. The header also includes a star icon, a refresh icon, a folder icon, and a user profile icon, along with a button labeled "Termina aggiornamento". Below the header is the eXist-db logo and the text "Open Source Native XML Database". A navigation bar contains links for Home, Demo, Documentation, Development, and Administration. To the right of the navigation bar is a search bar with the placeholder "Search Documentation" and a "Search" button. The main content area has a breadcrumb trail "Home / Documentation". The "Documentation" section title is followed by "(1Q20)". A welcome message says: "Welcome to eXist-db. This article serves as an index to the eXist-db documentation articles, which will help you get to know, install, and use eXist-db." Below this are two sections: "Getting Started" and "Basic Installation". On the right side, there's a "Contents" sidebar with a list of documentation topics.

Home / Documentation (1Q20)

Welcome to eXist-db. This article serves as an index to the eXist-db documentation articles, which will help you get to know, install, and use eXist-db.

**Getting Started**

The following articles and resources will help you get started using eXist-db.

**Basic Installation**

**Contents**

- [Getting Started](#)
- [XQuery](#)
- [Application development](#)
- [Interfaces](#)
- [Operations](#)
- [Java development](#)
- [Developing eXist-db](#)
- [Alphabetical index](#)
- [Subject index](#)

# Standard and Third Party Functions

## XQuery Function Documentation

Search

Search

Search in

 Browse

 Regenerate



<http://exist-db.org/xquery/transform>

*java:org.exist.xquery.functions.transform.TransformModule*

A module for dealing with XSL transformations.

*transform:transform*

```
transform:transform($node-tree as node()* , $stylesheet as item() , $parameters as node()?) as node()?
```

Applies an XSL stylesheet to the node tree passed as first argument. The stylesheet is specified in the second argument. This should either be an URI or a node. If it is an URI, it can either point to an external location or to an XSL stored in the db by using the 'xmldb:' scheme. Stylesheets are

# Monitoring (Monex)

existdb

≡

User

Monitoring (local)

Profiling

Indexes

Remote Console

Remote Monitoring

## Monitoring localhost

Home > Monitoring

ACTIVE DB PROCESSES

0 of 6

UPTIME

00h 31m

RUNNING QUERIES

0

WAITING THREADS

0

### Java Memory

Used memory (mb)

Committed memory (mb)

Memory Used

489 / 3140 M

Memory Committed

567 / 3140 M

Garbage Collect

# Admin Features



Version 5.3.1

Dashboard

Launcher

Package Manager

User Manager

Backup

Settings

Logout admin

PackageManager

type here to filter

UPLOAD DROPZONE - put your package(s) here.

Installed (20) Available (63)

APPLICATION <b>(TEI) Graphing for eXist-db</b> Version 0.2	
APPLICATION <b>Airlock</b> Version 2.1.0	
LIBRARY <b>airtable.xq</b> Version 1.0.3	

# eXide - XQuery Integrated development environment

The screenshot shows the eXide IDE interface. At the top, there is a menu bar with File, Edit, Navigate, Buffers, Application, XQuery, and Help. To the right of the menu, it says "Logged in as admin." Below the menu is a toolbar with buttons for New, New XQuery, Open, Save, Close, and Eval. The "Eval" button is highlighted. Next to it, it says "Current app: unknown" and "File Type: XQuery".

The main area has tabs for update-papyri..., test-gs.xql, lsj.xql, query-gs.xql, helmadik\_LSJL..., test.xls, and test-xsl.xml. The "update-papyri..." tab is active and contains the following XQuery code:

```
xquery version "3.1";
let $input := doc("/db/apps/cdt/test-xsl.xml")
let $xsl := doc("/db/apps/cdt/test.xsl")
return
    transform:transform($input,$xsl,())
```

Below the code editor, there is a navigation bar with a back arrow and the path /db/apps/cdt/transform.xq. Underneath the path are several configuration options: Adaptive Output (dropdown), Number of results: 10 (input field), Indent (checkbox checked), Live Preview (checkbox uncheckable), and Highlight Index Matches (checkbox checked).

The preview pane at the bottom shows the XML output of the XQuery. It includes the first few lines of the XML document:

```
<lb xmlns="http://www.tei-c.org/ns/1.0" n="1"/>
2
3 <gap xmlns="http://www.w3.org/1999/xhtml" xml:id="c932897" reason="illegible" quantity="1"
unit="character" ana="#vestige"/>
```

# Main XML-based Languages

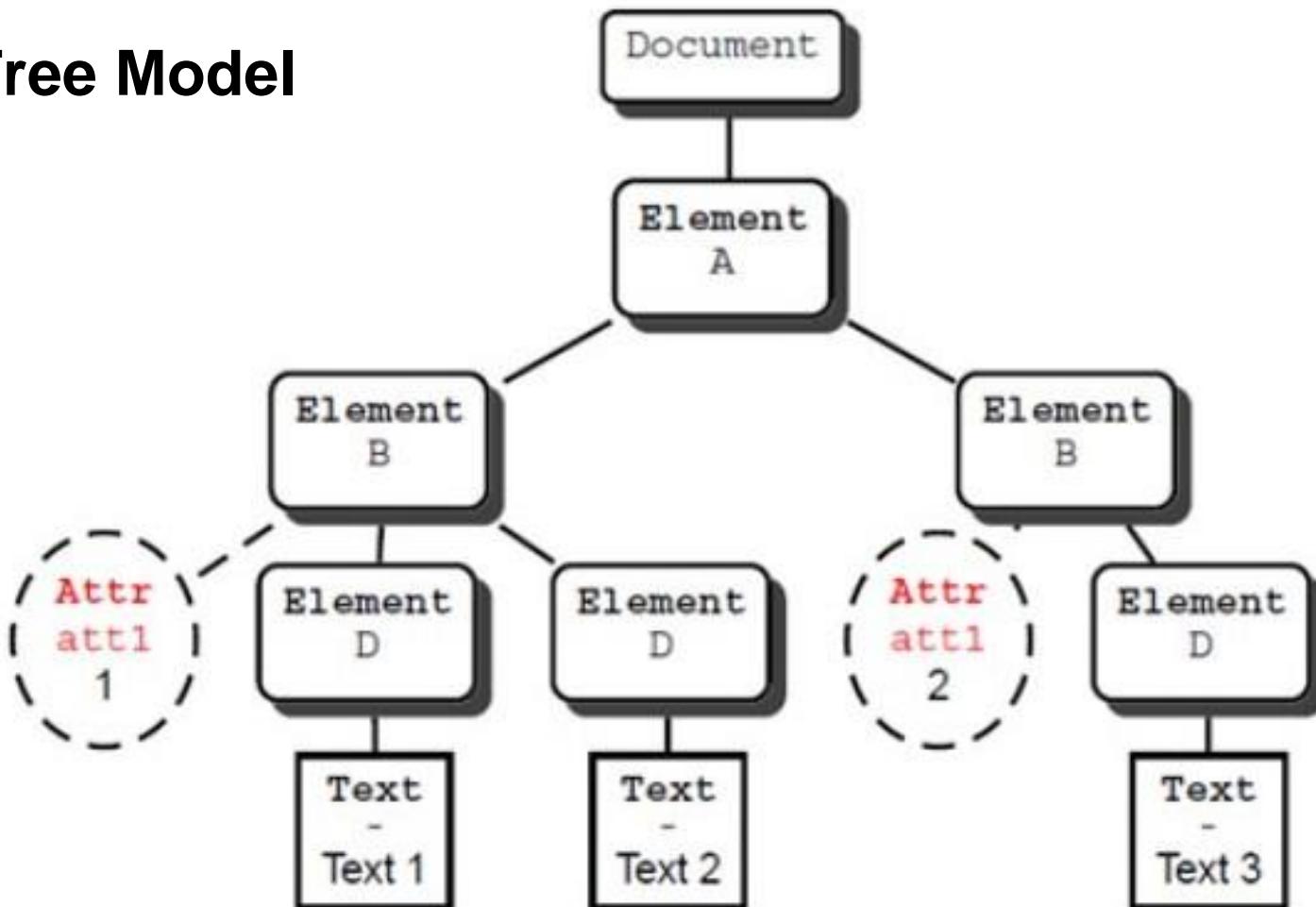
- **XPath**: expression language to select nodes from an XML document
- **XQuery**: powerful language designed to query and manipulate XML data
- **XSLT**: flexible language used to manipulate and transform XML documents



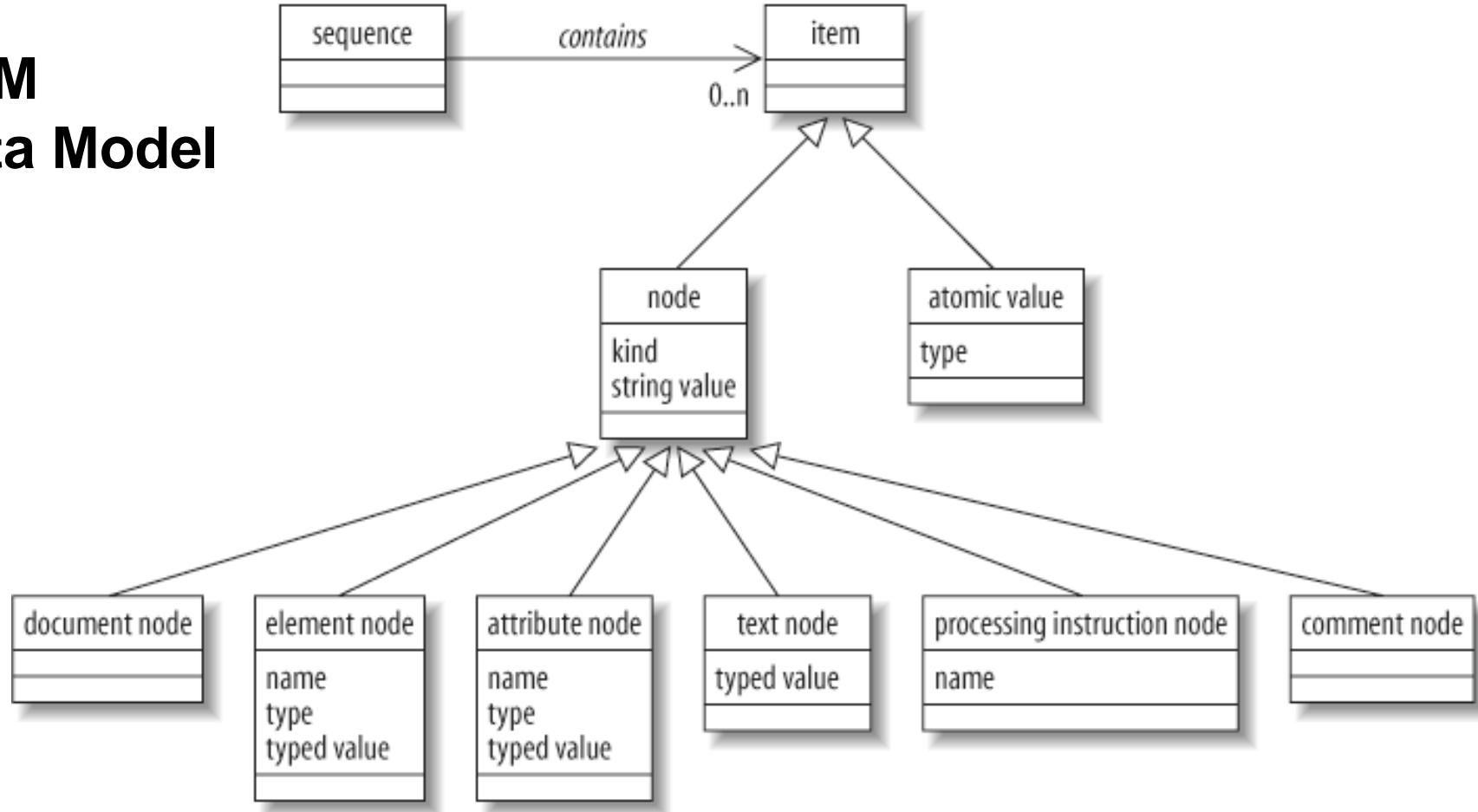


- The W3C specifications for XSLT, XQuery, and XPath **model an XML document as a tree**. This data model is known as **XDM**, and the nodes of an XDM tree are known as XDM nodes.
- XDM defines the information contained in the input to an XSLT processor as well as it defines all **permissible values of expressions** in the XSLT
- The **node-sets of XPath 1.0** are replaced in XPath 2.0 by **sequences of nodes**.

# XDM - Tree Model



# XDM Data Model



# XPath

- XPath is an **expression language** to select nodes from an *XML document*
- It is used to **traverse elements and attributes** in an XML tree
- It is **embeddable** in *XSLT, XQuery* and many other programming or scripting languages
- It is **based on axes** to establish the relations among nodes (e.g. *child, sibling, parent, ancestor*, etc.)

**Step patterns** are location paths that allow the processor to move freely around the node tree in any direction.

The **axis** part of the step pattern specifies the direction the XSLT processor should move.

The **node-test** part of the step pattern specifies the node to be matched.

```
<h1>New Listings</h1>

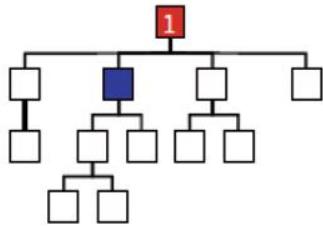
<section id="city_list">
|
<xsl:apply-templates
  select="listings/property[not(city=preceding::property/city)]"
  mode="cityList">
  |
    <xsl:sort select="city" />
  </xsl:apply-templates>
</section>
```

A green arrow points from the left margin to the `select` attribute of the `xsl:apply-templates` element. Three green arrows point down from the callout boxes to the corresponding parts of the `select` attribute: one to the axis (`preceding::`), one to the node-test (`property`), and one to the predicate (`not(city=preceding::property/city)`).

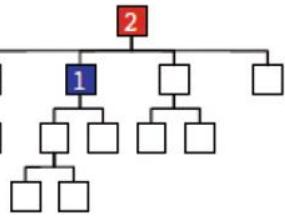
The **predicate** provides an expression that the node value should match.

 context node

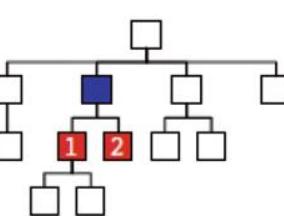
ancestor



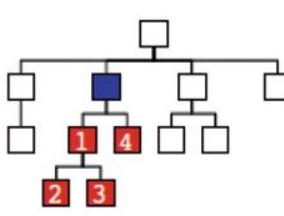
ancestor-or-self



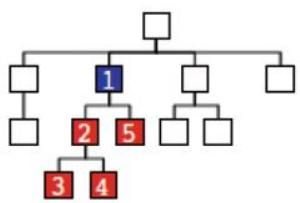
child



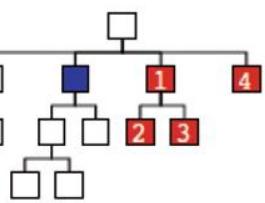
descendant



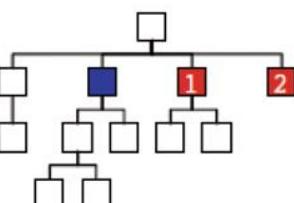
descendant-or-self



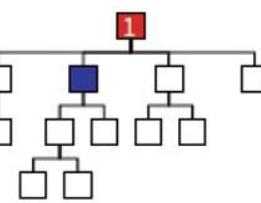
following



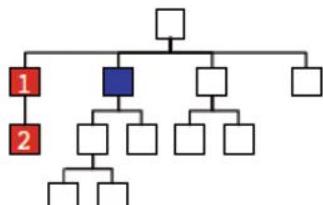
following-sibling



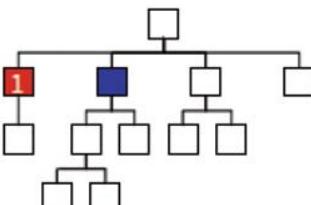
parent



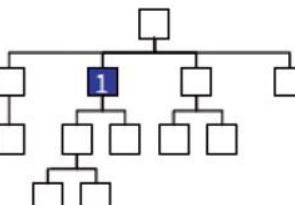
preceding



preceding-sibling



self



<a href="#">abs</a>	<a href="#">fold-left</a>	<a href="#">parse-xml</a>
<a href="#">accumulator-after</a>	<a href="#">fold-right</a>	<a href="#">parse-xml-fragment</a>
<a href="#">accumulator-before</a>	<a href="#">for-each</a>	<a href="#">path</a>
<a href="#">adjust-date-to-timezone</a>	<a href="#">for-each-pair</a>	<a href="#">position</a>
<a href="#">adjust-dateTime-to-timezone</a>	<a href="#">format-date</a>	<a href="#">prefix-from-QName</a>
<a href="#">adjust-time-to-timezone</a>	<a href="#">format-dateTime</a>	<a href="#">put</a>
<a href="#">analyze-string</a>	<a href="#">format-integer</a>	<a href="#">QName</a>
<a href="#">apply</a>	<a href="#">format-number</a>	<a href="#">random-number-generator</a>
<a href="#">available-environment-variables</a>	<a href="#">format-time</a>	<a href="#">regex-group</a>
<a href="#">available-system-properties</a>	<a href="#">function-arity</a>	<a href="#">remove</a>
<a href="#">avg</a>	<a href="#">function-available</a>	<a href="#">replace</a>
<a href="#">base-uri</a>	<a href="#">function-lookup</a>	<a href="#">resolve-QName</a>
<a href="#">boolean</a>	<a href="#">function-name</a>	<a href="#">resolve-uri</a>
<a href="#">ceiling</a>	<a href="#">generate-id</a>	<a href="#">reverse</a>
<a href="#">codepoint-equal</a>	<a href="#">has-children</a>	<a href="#">root</a>
<a href="#">codepoints-to-string</a>	<a href="#">head</a>	<a href="#">round</a>
<a href="#">collation-key</a>	<a href="#">hours-from-dateTime</a>	<a href="#">round-half-to-even</a>
<a href="#">collection</a>	<a href="#">hours-from-duration</a>	<a href="#">seconds-from-dateTime</a>
<a href="#">compare</a>	<a href="#">hours-from-time</a>	<a href="#">seconds-from-duration</a>
<a href="#">concat</a>	<a href="#">id</a>	<a href="#">seconds-from-time</a>
<a href="#">contains</a>	<a href="#">idref</a>	<a href="#">serialize</a>
<a href="#">contains-token</a>	<a href="#">implicit-timezone</a>	<a href="#">snapshot</a>
<a href="#">copy-of</a>	<a href="#">in-scope-prefixes</a>	<a href="#">sort</a>
<a href="#">count</a>	<a href="#">index-of</a>	<a href="#">starts-with</a>
<a href="#">current</a>	<a href="#">innermost</a>	<a href="#">static-base-uri</a>
<a href="#">current-date</a>	<a href="#">insert-before</a>	<a href="#">stream-available</a>
<a href="#">current-dateTime</a>	<a href="#">iri-to-uri</a>	<a href="#">string</a>
<a href="#">current-group</a>	<a href="#">json-doc</a>	<a href="#">string-join</a>
<a href="#">current-grouping-key</a>	<a href="#">json-to-xml</a>	<a href="#">string-length</a>

# XPath example in eXide

```
10 doc('../xml/switchboard.clarin.eu-step9.xml')//titleStmt//title[@type="short"]/text()
```

```
→ /db/apps/first//resources/xq>xpath-exemple.xq
```

◀ Adaptive Output ▾  Indent  Live Preview  Highlight Index Matches  +

```
1 eclo
```

# XQuery

XQuery is a powerful **query language** designed to query and manipulate XML data

Compared to XSLT, XQuery is extremely **compact** and this aspect enhances the **readability** of the scripts

## FLOWR syntax

XQuery is based on the **FLWOR** (*For, Let, Where, Order-by, Return*) construct. It is very compact and also students in traditional humanities can manage it

# FLOWR syntax (cont.)

A **FLWOR** query is an XQuery structure built around for, let, where, order by, and return clauses.

The **for clause** iterates through a sequence of nodes or atomic values.

The **let clause** is the local variable used in the query.

```
for $o in doc('gjc_orders.xml')//order
let $total := sum($o/product/(@qty*@salesPrice))
where $total >= 600
order by $total descending

return
<order id="{{$o/@orderID}" date="{{$o/@orderDate}}">
  <totalCharge>{
    concat("$", round-half-to-even($total, 2))
  }</totalCharge>

  {$o/product}
</order>
```

The **where clause** filters the result of the query.

The **order by clause** sorts the query result.

The **return clause** specifies the format and structure of the query result.

# For clause

- **Purpose:** Iterates over a sequence of items
- **Functionality:** The *For clause* binds a variable to each item in a sequence, allowing for iteration over large datasets, much like a loop in other programming languages. It can **iterate** over *nodes*, *values*, or *any sequence* generated by an XQuery expression

```
for $word in doc("aen.xml")//tei:w  
return $word
```

# Let clause

**Purpose:** Binds values to variables

**Functionality:** The *Let clause* is used to assign a value to a variable. Unlike the For clause, Let does not iterate but simply assigns a value to be used later in the FLWOR statement. It is useful for storing intermediate results

```
let $count = count(doc("aen.xml")//tei:w)
```

# Where clause

**Purpose:** Filters data based on conditions

**Functionality:** The *Where clause* restricts the results returned by the FLWOR expression, acting as a filter that allows only items meeting certain conditions to be processed in the subsequent clauses

```
where $word/@xml:lang = "grc"
```

# Order by clause

- **Purpose:** Sorts the results of the query
- **Functionality:** The *Order by clause* sorts the items returned by the query, based on *specified criteria*. This sorting can be done in ascending or descending order

order by \$word/text() ascending

# Return clause

**Purpose:** Specifies what is to be returned from the query

**Functionality:** The

*Return clause*

determines what will be included in the final output of the FLWOR statement.

This can be a node, a constructed element, a value, or any combination of these

```
for $line in doc("aen.xml")//l  
let $line_id := $line/@xml:id  
let $text := string-join($line/w, ' ')  
return  
{$line_id} {$text}
```

# XSLT transformations from XQuery

XQuery can *perform XSLT* transformations, which are more familiar to the **XML-TEI community** than the xquery language

```
<!--transform the TEI lg in an XHTML div-->
{transform:transform($lg,doc("trans/fromTeiToHtml.xsl"),())}
```

```
<!-- process tei:lg -->
<xsl:template match="tei:lg">
  <xsl:element name="div">
    <!--<xsl:apply-templates select="tei:head"/><!--&gt;
    &lt;xsl:for-each select="tei:l"&gt;
      &lt;xsl:element name="span"&gt;
        &lt;xsl:attribute name="class"&gt;latcit&lt;/xsl:attribute&gt;
        &lt;xsl:attribute name="id"&gt;
          &lt;xsl:value-of select="@n"/&gt;
        &lt;/xsl:attribute&gt;
        &lt;xsl:value-of select="@n"/&gt;
        &lt;xsl:text&gt; &lt;/xsl:text&gt;
      &lt;/xsl:element&gt;
      &lt;xsl:element name="span"&gt;
        &lt;xsl:attribute name="class"&gt;verseText&lt;/xsl:attribute&gt;
        &lt;xsl:for-each select="tei:w"&gt;
          &lt;xsl:value-of select="text()"/&gt;
          &lt;xsl:text&gt; &lt;/xsl:text&gt;
        &lt;/xsl:for-each&gt;
      &lt;/xsl:element&gt;
      &lt;br/&gt;
    &lt;/xsl:for-each&gt;
  &lt;/xsl:element&gt;
&lt;/xsl:template&gt;</pre>
```

# eXide - XSLT Example

```
1 ▼ <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.tei-c.org/ns/1.0" version="2.0">
2   <xsl:output method="xml" indent="yes" encoding="UTF-8"/>
3   <xsl:param name="user">USER</xsl:param>
4 ▶ <xsl:variable name="license" exclude-result-prefixes="tei" xpath-default-namespace="http://www.tei-c.org/ns/1.0"><!--></xsl:variable>
11
12 ▶ <xsl:template match="node() | @* "><!--></xsl:template>
17
18 ▼ <xsl:template match="/">
19 ▼ <xsl:comment>
20   <xsl:value-of select="/tei:TEI/tei:teiHeader/tei:fileDesc/tei:titleStmt/tei:title"/>
21 </xsl:comment>
22 <xsl:apply-templates/>
23 </xsl:stylesheet>
```

## Adaptive Output

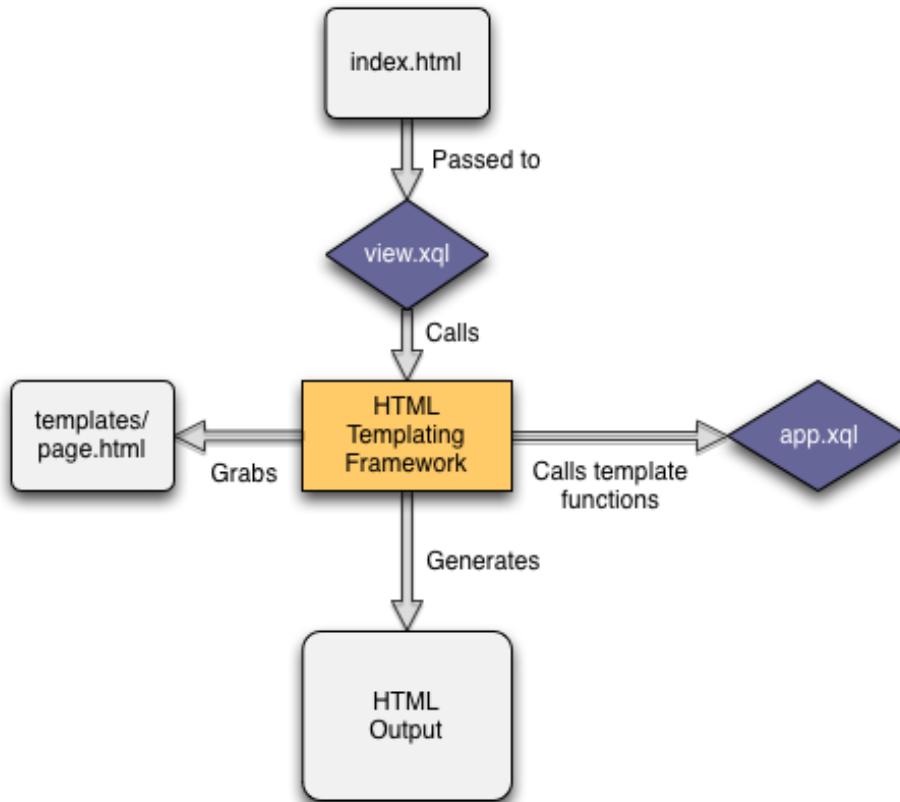
## Indent

## Live Preview

## Highlight Index Matches

1

# eXist-db HTML Templating Framework



- ✓ Clean **separation of concerns**
- ✓ Connect **HTML views** with the **application logic**
- ✓ Call the **HTML Templating** for every **URL ending with .html**
- ✓ **XQuery** function calls
- ✓ **Templating functions** provided by the module

# Demo Web Applet

My amazing first-app application [Home](#) [corpus](#) [credits](#) [about](#)

## My amazing first-app application

Lemma Query:

domus

search

[index data](#)

non umquam grauis aere **domum** mihi dextra redibat. ( 1 , 35 )  
est mihi namque **domi** pater, est iniusta nouerca; ( 3 , 33 )  
Ipsae lacte **domum** referent distenta capellae ( 4 , 21 )  
depulsos a lacte **domi** quae clauderet agnos, ( 7 , 15 )  
Ite **domum** pasti, si quis pudor, ite iuuenci. ( 7 , 44 )  
Ite **domum** saturae, uenit Hesperus, ite capellae. ( 10 , 77 )

## Query Result

- 1 domum (*domus* in 1 , 35)
- 2 domi (*domus* in 3 , 33)
- 3 domum (*domus* in 4 , 21)
- 4 domi (*domus* in 7 , 15)
- 5 domum (*domus* in 7 , 44)
- 6 domum (*domus* in 10 , 77)

# Folder and File Scaffolds using Yeoman

## generator-exist



exist app scaffolding

Let [Yeoman](#) take care of creating directory and file scaffolds for different types of [exist-db](#) EXpath packages. Just answer the prompts about what you want to build, and a sensible directory structure with fully configured readmes, buildfiles, etc will be set up for you.

No more manual closing of html5 tags, inserting app names into config files, or wondering if you actually increase the version number in all the places.

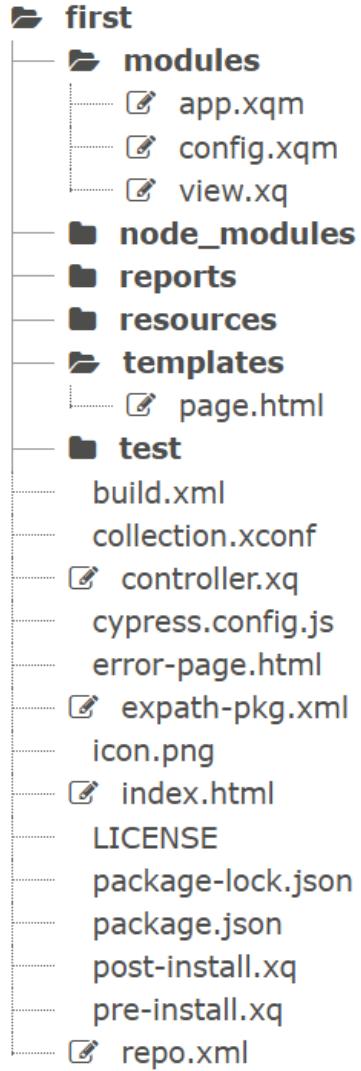
## Installation

First, install [Yeoman](#) and generator-exist using [npm](#) (we assume you have pre-installed [node.js](#)).

## From GitHub

To install a pre-release version:

```
npm i -g yo@4
npm i -g git://github.com/eXist-db/generator-exist.git
```



# Folder and File Scaffolds using Yeoman

```
angelodel80@LAPTOP-V8V3MLG0:~/labs/lab-yo$ ls
first-app
angelodel80@LAPTOP-V8V3MLG0:~/labs/lab-yo$ yo
? 'Allo angelodel80! What would you like to do? (Use arrow keys)
  Run a generator
> @existdb/exist
-----
Update your generators
Install a generator
Find some help
Get me out of here!
(Move up and down to reveal more choices)
-- Y
you like to call your exist-db application? (lab yo) |
```

# Upload the Scaffolding Application to eXist

PackageManager

type here to filter

Installed (13) Available (75)

	<b>APPLICATION</b>	 
<b>My amazing first-app application</b>		
Version	1.0.0	
Url	/exist/apps/first/	
Name	http://exist-db.org/apps/first	
Description	My amazing first-app application	
Author(s)	Angelo	
Abbreviation	first	
Website	www.ilc.cnr.it	
License	MIT	

index.html X

...

page.html X

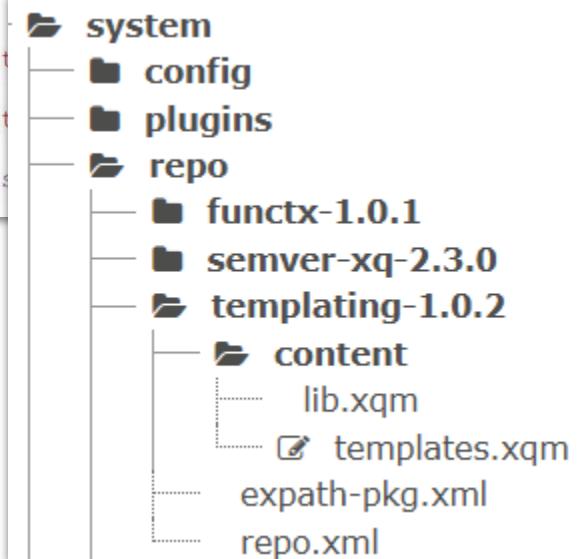
...

```
1 <div xmlns="http://www.w3.org/1999/xhtml"
      data-template="templates:surround"
      data-template-with="templates/page.html"
      data-template-at="content">
2   <div class="row" style="padding-top:
      1.5rem">
3     <div class="col-md-8">
4       <div class="page-header">
5         <h1
            data-template="config:app-title">Gen
            erated page</h1>
6       </div>
7     <div class="alert alert-success"> ...
12   </div>
13   <div class="row">
14     <div class="col-md-6"> ...
16     </div>
17     <div class="col-md-6">
18       <div data-template="app:test" />
19     </div>
20   </div>
21 </div>
22
23   <div class="col-md-4">
24     <h2>Application Info</h2>
25     <div data-template="config:app-info" />
```

```
2   <html xmlns="http://www.w3.org/1999/
      xhtml">
3     <head>
4       <title
            data-template="config:app-title">first
            -app</title>
5       <meta name="viewport"
            content="width=device-width,
            initial-scale=1.0"/>
6       <meta data-template="config:app-meta" />
7
8       <link rel="shortcut icon"
            href="resources/images/
            exist_icon_16x16.ico"/>
9       <link rel="stylesheet" type="text/
            css" href="resources/css/bootstrap.
            min.css"/>
10      <link rel="stylesheet" type="text/
            css" href="resources/css/style.css"/>
11      <script type="text/javascript"
            src="resources/scripts/bootstrap.
            bundle.min.js"/>
12
13     <body id="body">
14       <div class="container">
15         <nav class="navbar navbar-expand-lg
            navbar-light bg-light">
```

# Templates Module

```
9 ▶ (~
0   : @deprecated use lib:include instead
1   :)
2 ▶ declare function templates:include($node as node(), $model as map(*), $path as xs:string) { ↗};
3
4 declare function templates:surround($node as node(), $model as map(*), $with as xs:string, $at as xs:string?,
5   $using as xs:string?, $options as xs:string?) { ↗};
6
7 declare %private function templates:surround-options($model as map(*), $opt
8
9 declare %private function templates:process-surround($node as node(), $cont
6
```



## My amazing first-ap

- [Home](#)
- [TBA](#)
- [TBA](#)
- [TBA](#)

# My amazing first-app application

This is the entry page into your application and was generated by yeoman. It uses HTML templates for a clean separation of HTML views and application logic.

To add your own template functions, start by editing the XQuery module:

`app.xqm`

The page template uses the [Bootstrap](#) CSS library for the page layout.

Dummy template output generated by function `app:test` at 2024-05-06T18:34:08.959Z. The templating function was triggered by the class attribute `class="app:test"`.

# My amazi

This is the entry page into your application and was generated by yeoman. It u

To add your own template functions, start by editing the XQuery module:

`app.xqm`

```
<meta data-template="config:app-meta"/>
<link rel="shortcut icon" href="resources/images/exist_icon_16x16.ico"/>
<link rel="stylesheet" type="text/css" href="node_modules/bootstrap/dist/css/bootstrap.min.css"/>
<script type="text/javascript" src="node_modules/bootstrap/dist/js/bootstrap.bundle.min.js"/>
```

## Application Info

app	/db/apps/first
collection:	
name:	<a href="http://exist-db.org/apps/first">http://exist-db.org/apps/first</a>
abbrev:	first
version:	1.0.0
spec:	1.0
title:	My amazing first-app application
dependency:	<a href="http://exist-db.org/5.3.0">http://exist-db.org 5.3.0</a>
dependency:	<a href="http://exist-db.org/html-templating">http://exist-db.org/html-templating</a>
dependency:	1.0.2
description:	My amazing first-app application
author:	Angelo
website:	<a href="http://www.ilc.cnr.it">www.ilc.cnr.it</a>
status:	SNAPSHOT

# HTML of the Applet

```
<!DOCTYPE html>
<html>
<head>
    <title data-template="config:app-title">My amazing first-app application</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="My amazing first-app application">
        <meta name="creator" content="Angelo">
    <link rel="shortcut icon" href="resources/images/exist icon 16x16.ico">
    <link rel="stylesheet" type="text/css" href="node modules/bootstrap/dist/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="resources/css/style.css">
    <script type="text/javascript" src="node modules/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body id="body">
<div class="container">
    <nav class="navbar navbar-expand-lg navbar-light bg-light" role="navigation">
        <div class="container-fluid">
            <a data-template="config:app-title" class="navbar-brand" href=".index.html">My amazing first-app application</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link active" aria-current="page" href="index.html">Home</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
</div>
```

# Config Module Template (Config.xqm)

```
declare %templates:wrap function config:app-title($node as node(), $model as map(*)) as text() {
    $config:expat-descr/xpath:title/text()
};

declare function config:app-meta($node as node(), $model as map(*)) as element()* {
    <meta xmlns="http://www.w3.org/1999/xhtml" name="description" content="{$config:repo-descriptor
        /repo:description/text()}"/>,
    for $author in $config:repo-descriptor/repo:author
    return
        <meta xmlns="http://www.w3.org/1999/xhtml" name="creator" content="{$author/text()}"/>
};
```

```
declare variable $config:repo-descriptor := doc(concat($config:app-root, "/repo.xml"))/repo:meta;

declare variable $config:expat-descr := doc(concat($config:app-root, "/xpath-pkg.xml"
))/xpath:package;

declare variable $config:data-root := $config:app-root || "/data";
```

# Input - Query Lemma

```
<div class="alert alert-success">
  <div class="row">
    <div class="col-md-8">
      <form action="" method="POST" class="form form-horizontal">
        <div class="form-group">
          <label for="lemma" class="col-md-3 hidden-xs control-label">Lemma Query:</label>
          <div class="col-md-6 col-xs-14">
            <span class="input-group">
              <input name="lemma" type="search" data-template="templates:form-control"
                     class="form-control" placeholder="Search for lemma"/>
              <span style="margin-left:2px" class="input-group-btn">
                <button id="f-btn-search" type="submit" class="btn btn-primary">
                  <span class="glyphicon glyphicon-search">search</span>
                </button>
              </span>
            </span>
          </div>
        </div>
      </form>
    </div>
  </div>
```

Lemma Query:

domus

search

```
<div class="row">
  <div class="col-md-4">
    <p>
      <input type="button" value="index data"/>
    </p>
  </div>
  <div class="col-md-8">

    <div data-template="app:context" style="padding-bottom:3em; margin-bottom:2ex; max-height
      :450px; overflow:auto">
      <div data-template="templates:each" data-template-from="words" data-template-to="w">
        <span class="loci" data-template="app:locus"/> (<span data-template="app:eclo"/>,
          <span data-template="app:verse"/> )
      </div>
    </div>
  </div>
</div>
```

non umquam grauis aere domum mihi dextra redibat. ( 1 , 35 )  
est mihi namque domi pater, est iniusta nouerca; ( 3 , 33 )  
Ipsae lacte domum referent distenta capellae ( 4 , 21 )  
depulsos a lacte domi quae clauderet agnos, ( 7 , 15 )  
Ite domum pasti, si quis pudor, ite iuuenci. ( 7 , 44 )  
Ite domum saturae, uenit Hesperus, ite capellae. ( 10 , 77 )

```
declare
    %templates:wrap %templates:default("lemma", "ttt")
function app:context($node as node(), $model as map(*), $lemma as xs:string) as map(*) {
    let $l := if ($lemma = "") then "ttt" else $lemma
    let $eclo := doc('/db/apps/first/resources/xml/switchboard.clarin.eu-step9.xml')
    let $words := $eclo//w[ft:query(@lemma, string($l))]
    return map {
        "words" : $words
    }
};

declare
    %templates:wrap
function app:locus($node as node(), $model as map(*)) as xs:string {
    string-join($model("w")/ancestor::l)
};
declare
    %templates:wrap
function app:eclo($node as node(), $model as map(*)) as xs:string {
    string($model("w")/ancestor::div1/descendant::head/descendant::title)
};
declare
    %templates:wrap
function app:verse($node as node(), $model as map(*)) as xs:string {
    string($model("w")/ancestor::l@n)
};
```

```
<div class="col-md-4" style="max-height:450px; overflow:auto">
  <h2>Query Result</h2>
  <ul id="results" data-template="app:query">
    </ul>
</div>
```

My amazing first-app application [Home](#) [corpus](#) [credits](#) [about](#)

# My amazing first-app application

Lemma Query:

miser

search

[index data](#)

Heu heu, quid uolui misero mihi? floribus Austrum ( 2 ,58 )  
stridenti miserum stipula disperdere carmen? ( 3 ,27 )  
Mantua uae miserae nimium uicina Cremonae, ( 9 ,28 )

## Query Result

- 1 miser (*misero*)
- 2 miser (*miserum*)
- 3 miser (*miserae*)

```
declare
    %templates:wrap %templates:default("lemma", "ttt")
function app:query($node as node(), $model as map(*), $lemma as xs:string) {
    let $l := if ($lemma = "") then "ttt" else $lemma
    let $eclo := doc('/db/apps/first/resources/xml/switchboard.clarin.eu-step9.xml')
    let $words := $eclo//w[ft:query(@lemma, string($l))]
    for $w at $idx in $words
    return
        <li style="list-style-type:none">
            <span style="font-weight:bold">{$idx}</span>
            <span class="lemmata" style="padding-left:1em">{string($w)}</span>
            <span style="font-style:italic; padding-left:1em">
                (
                    {string($w/@lemma)} in
                    {string($w/ancestor::div1/descendant::head/descendant::title)},
                    {string($w/ancestor::l/@n)}
                )
            </span>
        </li>
} ;
```

```
<script type="text/javascript">
  window.addEventListener("load", (event) => {
    var lemmata = document.querySelectorAll(".lemmata");
    var loci = document.querySelectorAll(".loci");

    for (let i = 0; i < lemmata.length; i++) {
      var mark = new Mark(loci[i]);
      mark.mark(lemmata[i].innerText);
    }
  });
</script>
```

Lemma Query:

domus

search

index data

non umquam grauis aere **domum** mihi dextra redibat. ( 1 , 35 )  
est mihi namque **domi** pater, est iniusta nouerca; ( 3 , 33 )  
Ipsae lacte **domum** referent distenta capellae ( 4 , 21 )  
depulsos a lacte **domi** quae clauderet agnos, ( 7 , 15 )  
Ite **domum** pasti, si quis pudor, ite iuuenci. ( 7 , 44 )  
Ite **domum** saturae, uenit Hesperus, ite capellae. ( 10 , 77 )



```
1 <collection xmlns="http://exist-db.org/collection-config/1.0">
2   <index>
3     <lucene>
4       <analyzer class="org.apache.lucene.analysis.standard.StandardAnalyzer"/>
5       <text qname="@lemma"/>
6     </lucene>
7   </index>
8 </collection>
```

My amazing first-app application [Home](#) [corpus](#) [credits](#) [about](#)

# My amazing first-app application

Lemma Query:

noverca~0.5

search

est mihi namque domi pater, est iniusta **noverca;** ( 3 , 33 )

index data

## Query Result

1 nouerca; ( nouerca; in 3 , 33 )

# My amazing first-app application

Lemma Query:

mat\*

search

index data

Sic canibus catulos similes, sic matribus haedos ( 1 , 22 )  
Triste lupus stabulis, maturis frugibus imbræ, ( 3 , 80 )  
nec Linus, huic mater quamuis atque huic pater adsit, ( 4 , 56 )  
Incipe, parue puer, risu cognoscere matrem ( 4 , 60 )  
( Matri longa decem tulerunt fastidia menses ) ( 4 , 61 )  
atque deos atque astra uocat crudelia mater. ( 5 , 23 )  
( dux ego uester eram ) uidì cum matre legentem. ( 8 , 38 )  
Saeuus Amor docuit natorum sanguine matrem ( 8 , 47 )  
commaculare manus; crudelis tu quoque, mater. ( 8 , 48 )  
Crudelis mater magis, an puer improbus ille? ( 8 , 49 )  
inprobus ille puer; crudelis tu quoque, mater. ( 8 , 50 )  
aut custos gregis aut maturaे uinitor uuae! ( 10 , 36 )

## Query Result

- 1 matribus (*matris in 1 , 22*)
- 2 maturis (*maturus in 3 , 80*)
- 3 mater (*mater in 4 , 56*)
- 4 matrem (*mater in 4 , 60*)
- 5 (Matri (*Mater in 4 , 61*)
- 6 mater. (*materuintus in 5 , 23*)
- 7 matre (*mato in 8 , 38*)
- 8 matrem (*mater in 8 , 47*)
- 9 mater. (*materuintus in 8 , 48*)
- 10 mater (*mater in 8 , 49*)
- 11 mater. (*materuintus in 8 , 50*)
- 12 maturaе (*maturus in 10 , 36*)

# Mode: Bassani's Postille Applet

Postille verbali

[Postilla 1](#) [Postilla 2](#)

Postilla:

*Ma allora è fare storia l'atto di imparziale obiettività, e solo questo; o anche il successivo etico-pedagogico di valutazione etica.*

Testo:

" Di qui il dovere d'imparziale obiettività e quello di suggestione etico-pedagogica, che ad un tempo incombono allo storico. "

[Dettagli](#) [Analisi linguistica](#) [Interventi autoriali](#)

## Dettaglio immagine

Pagina 100

In primo luogo, anche se siamo consapevoli sensibilmente collegabili a quelli che, ascoltando una persona, interpretano gli elementi semantic che

[◀ Elenco ▶](#)

Postille non verbali

Barra laterale ondulata:

" Ma il gusto del commerciante di schiavi o dello speculatore di borsa non è il gusto del martire o del filantropo o del missionario, così come il gusto del cafone non è il gusto del conoscitore. Chi vede nel passato le manifestazioni di tale volgarità o finezza, è normalmente portato a preferir questa a quella: così la storia aiuta a farsi un palato. "

U Testo sottolineato:

" Ma il gusto del commerciante di schiavi "

★ Asterisco:

" Di qui il dovere d'imparziale obiettività e quello di suggestione etico-pedagogica, che ad un tempo incombono allo storico. "

U Testo sottolineato:

" Di qui il dovere d'imparziale obiettività e quello di suggestione etico-pedagogica, che ad un tempo incombono allo storico. "

powered by

# Mode: Voci dall'Inferno Applet

[HOME](#)[IL PROGETTO](#)[ELENCO DEI TESTIMONI](#)[CERCA UN TESTIMONE](#)[INTERROGA IL CORPUS](#)[PERSONE](#)

AS: lo registro quello che diciamo, poi **inspira** e... allora noi abbiamo constatato, facendo queste... venti interviste, (...) perché ci sono **inspira** degli argomenti che vengono sempre toccati, (...) **inspira eh** alcuni (...) meno, altri più, ma tutti quanti, comunque, abbastanza. **inspira Eh** lo a **Goti** anche ho chiesto **inspira ehm** cosa pensa lei dell'umiliazione, **cosa-pens-, cosa-pensava-le-**, cioè, umiliazione, **ehm** solidarietà, se **se** esisteva, se era possibile in che senso secondo te **inspira ehm** e poi se ci sono (...) delle cose, degli episodi, perché poi in realtà il campo (...) **sospira** da quello che ho capito è fatto di piccole... (...) **di, di, di** miliardi di piccole...

LS: Piccole **storie**

LS: Piccole grandi storie

AS: Si, (...) si, quindi questo, (...) **inspira salvo** che appunto tutto questo con te nasce perché (...) **perché** è come se tu avessi mantenuto, no?, **ne-**, nel raccontarti la storia, (...) una dignità familiare, uno spessore... relazionale emotivo (...) che gli altri tralasciano, **tralasciano**, tu non lo tralasci

LS: Beh **XXX**

LS: Che gli altri tralasciano?

AS: Si

LS: Com'è possibile?

AS: Non lo so perché. (...) Credo, forse, perché non ce la fanno

## Fenomeni marcati

Buco nella registrazione: **GAP**

Parola non chiara: **UNCLEAR**

Pausa: **PAUSE**

Esclamazione: **VOCAL**

Rumore accidentale: **INCIDENT**

Movimento: **KINESIC**

Frase o parola riformulata/ripetuta: **DEL**

Parola errata: **SIC**

Parola corretta: **CORR**

Forma dialettale: **ORIG**

Forma regolarizzata: **REG**

Parola enfatizzata: **EMPH**

Parola in lingua straniera: **FOREIGN**

Antroponimo: **PERSNAME**

Luogo: **PLACENAME**

[MOSTRA TUTTI I FENOMENI](#)

# What's next



APPLICATION



**TEI Publisher**

Version 9.0.0



LIBRARY



**TEI Publisher: Processing Model Libraries**

Version 4.0.0

# TEIPublisher

## What is TEI Publisher?



- a standard-based toolbox designed for reusability and sustainability
- providing reusable building blocks for creating a digital edition

# TEIPublisher - Web Component

The screenshot shows a web browser interface for the TEIPublisher Web Component documentation. The title bar includes the logo and the component name: <pb-ajax> /src/pb-ajax.js. The navigation bar has tabs for API and Demo, with Demo selected. A checkbox labeled "only elements with demos" is checked. On the left, a sidebar lists various components: dts-client, pb-ajax, pb-autocomplete, pb-browse-docs, pb-clipboard, pb-code-editor, pb-code-highlight, pb-codepen, pb-collapse, pb-combo-box, pb-custom-form, pb-document, pb-download, pb-drawer, pb-edit-app, and pb-edit-xml. The main content area displays a demo for the pb-ajax component, titled "Accedi". It features four buttons: DOCUMENTATION: PDF, DOCUMENTATION: FO SOURCE, LETTER: PDF, and EDIT ODD. Below the buttons is a code snippet in a monospaced font:

```
1  <pb-page endpoint="https://teipublisher.com/exist/apps/tei-publisher" api-version="1.0.0">
2    <pb-document id="document1" path="doc/documentation.xml" odd="docbook" view="div"></pb-document>
3
4    <pb-progress></pb-progress>
5    <main>
6      <pb-login id="loginElem"></pb-login>
7
8      <pb-download type="pdf" src="document1" dialog="downloadDialog" title="Download PDF">
9        <paper-button raised="">Documentation: PDF</paper-button>
10       </pb-download>
11       <pb-download type="pdf" src="document1" title="Download PDF" source="" download="">
12         <paper-button raised="">Documentation: FO source</paper-button>
13       </pb-download>
14       <!-- old api: -->
15       <!-- pb-download type="pdf" url="test/graves6.xml" dialog="downloadDialog" title="Download PDF"
16           odd="graves.odd" -->
17       <pb-download type="pdf" url="api/document/test%2Fgraves6.xml" dialog="downloadDialog" title="Download PDF" odd="graves.odd">
18         <paper-button raised="">Letter: PDF</paper-button>
19       </pb-download>
20
21       <pb-restricted login="loginElem">
22         <!-- old api: -->
23         <!--pb-ajax url="modules/lib/regenerate.xql?odd=graves.odd"-->
24         <pb-ajax url="api/odd?odd=graves.odd" method="post">
25           <paper-button raised="">Recompile ODD</paper-button><span slot="title">Recompile ODD</span>
26         </pb-ajax>
27       </pb-restricted>
```

# Intro to eXist-db and applet XQuery

# THANKS

Angelo Mario Del Grosso (ILC-CNR, Pisa)

FeDHLab