

28msec

XQuery in the Cloud

Cutting Edge Web Applications with
XQuery and Sausalito

William Candillon {william.candillon@28msec.com}

The screenshot shows a web browser window with the URL `http://www.ibm.com/developerworks/xml/library/x-zorba/index.html`. The page is from IBM developerWorks and features the following content:

- Page Header:** Includes the IBM logo, a search bar, and navigation links for Home, Solutions, Services, Products, Support & downloads, and My IBM.
- Breadcrumbs:** `developerWorks > XML > Technical library >`
- Article Title:** **Building XQuery-powered applications with PHP and Zorba**
- Subtitle:** **Add XQuery support to PHP with Zorba**
- Author:** [Vikram Vaswani](#), Founder, Melonfire
- Date:** 03 Nov 2009
- Level:** Intermediate
- PDF:** [A4 and Letter](#) (207KB | 27 pages) [Get Adobe® Reader®](#)
- Activity:** 3099 views
- Summary:** Zorba is an open-source, robust, and standards-compliant XQuery processor. The Zorba extension in PHP provides an API to Zorba functions from within PHP, and thereby allows developers to add sophisticated XQuery processing to their PHP/XML applications. Examine the Zorba PHP API in detail, and how to use it for a variety of purposes.
- Tags for this article:** `php, vikram_vaswani, xml, xquery, zorba`
- Introduction:** Even the most hardened critic can agree that when it comes to working with XML-based technologies, you might do a lot worse than PHP. Very few programming languages offer the depth and breadth of XML extensions of PHP:
 - SimpleXML and XMLReader for basic XML processing
 - XSLT and XML-DAS for XML transformation
 - SAX for event-based parsing
 - DOM for tree-based parsing
 - WDDX for data serialization
 - XMLWriter for dynamic XML document creationAll of these extensions are based on the libxml2 library, and provide a robust, unified foundation for XML processing in PHP.

That said, one important XML-based technology doesn't appear in the list above: XQuery. While SimpleXML does include some basic XQuery support, it's not (nor is it supposed to be) a full-fledged


Frequently used acronyms

- API: Application program interface
- DOM: Document Object Model
- HTML: Hypertext Markup Language
- HTTP: Hypertext Transfer Protocol
- JSON: JavaScript Object Notation
- PDF: Portable Document Format
- REST: REpresentational State Transfer
- SAX: Simple API for XML
- USP: Unique Selling Proposition
- W3CL World Wide Web Consortium
- WDDX: Web Distributed Data Exchange
- XML: Extensible Markup Language
- XML-DAS: XML Data Archiving Service

Table of contents

- Introduction
- Installing Zorba
- Understanding basic usage
- Processing sequences
- Filtering and sorting data
- Manipulating node collections
- Using REST functions
- Using JSON functions
- Conclusion
- Resources
- About the author
- Comments

Next steps from IBM



The screenshot shows a web browser window with the title "[xquery-talk] XQuery Implementation for PHP". The address bar contains the URL "http://x-query.com/pipermail/talk/2009-November/003621.html" and a search bar with "Google". The main content area displays the email subject "[xquery-talk] XQuery Implementation for PHP" in a large, bold font. Below the subject, the sender is identified as "Vyacheslav Sedov" with the email address "vyacheslav.sedov at gmail.com" and the date "Wed Nov 4 10:03:02 PST 2009". A list of navigation links follows: "Previous message: [xquery-talk] XQuery Implementation for PHP", "Next message: [xquery-talk] XQuery count()", and "Messages sorted by: [date] [thread] [subject] [author]". A horizontal line separates the header from the body text. The body text consists of two paragraphs: "look like 'soup from stone' (or nails or axe - depend from country) - php should leave our lives at all" and "Stephen Leacock - 'I detest life-insurance agents: they always argue that I shall some day die, which is not so.' - http://www.brainyquote.com/quotes/authors/s/stephen_leacock.html".

[xquery-talk] XQuery Implementation for PHP

http://x-query.com/pipermail/talk/2009-November/003621.html

[xquery-talk] XQuery Implementation for PHP

Vyacheslav Sedov [vyacheslav.sedov at gmail.com](mailto:vyacheslav.sedov@gmail.com)
Wed Nov 4 10:03:02 PST 2009

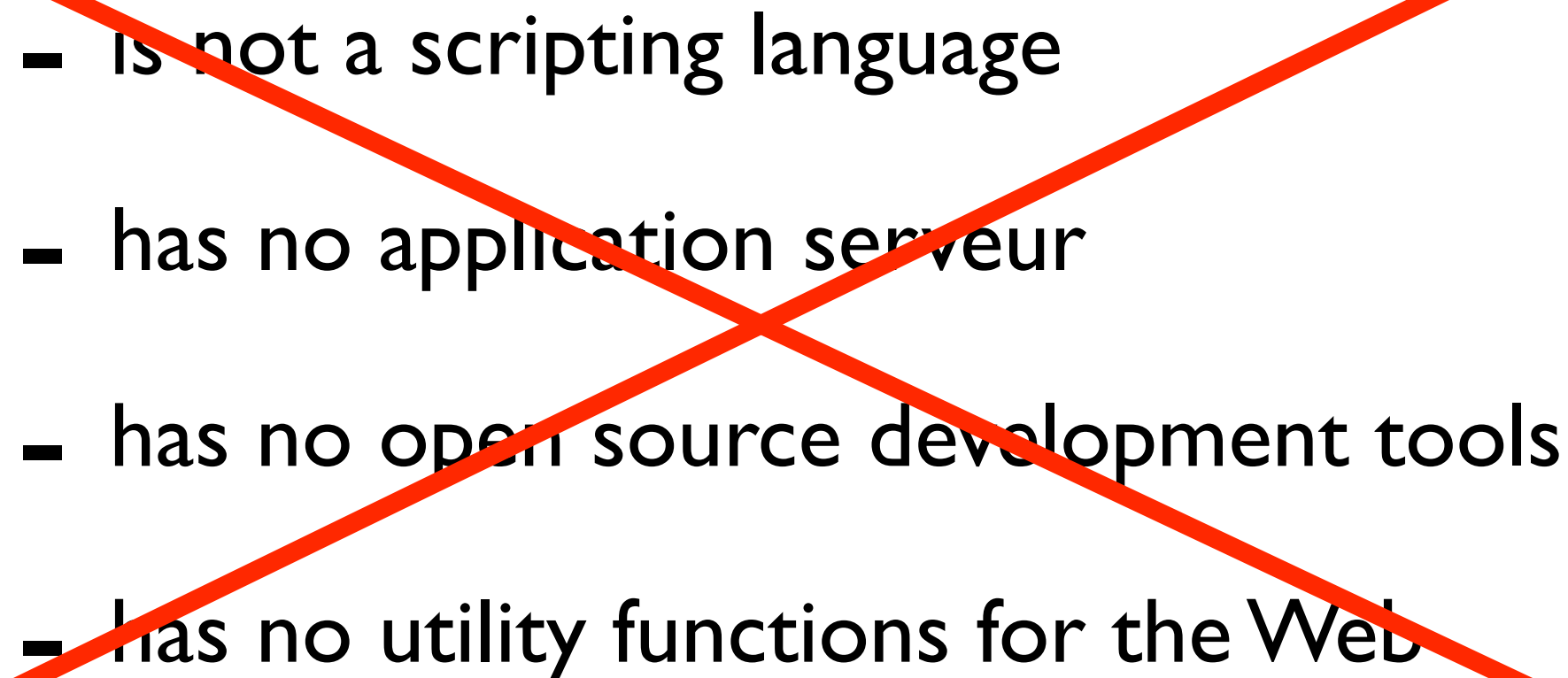
- Previous message: [\[xquery-talk\] XQuery Implementation for PHP](#)
- Next message: [\[xquery-talk\] XQuery count\(\)](#)
- Messages sorted by: [\[date\]](#) [\[thread\]](#) [\[subject\]](#) [\[author\]](#)

look like "soup from stone" (or nails or axe - depend from country) -
php should leave our lives at all

Stephen Leacock - "I detest life-insurance agents: they always argue
that I shall some day die, which is not so." -
http://www.brainyquote.com/quotes/authors/s/stephen_leacock.html

- XQuery:
 - is not a scripting language
 - has no application server
 - has no open source development tools
 - has no utility functions for the Web
-

- XQuery:

- is not a scripting language
 - has no application server
 - has no open source development tools
 - has no utility functions for the Web
- 

~~XQuery is not a
scripting language!~~

- XQuery Scripting Extension 1.0 and XQuery Update Facility 1.0

- Imperative programming

```
declare function sequential local:hello-world() {  
  declare $hello := "Hello";  
  declare $world := "World";  
  exit with concat($hello, " ", $world);  
}
```

- Side-effects

```
replace value of $client/@id with "foo";
```

- Loops

```
while(true()) { server:handle() }
```

- Try/Catch

```
try { 3 + "2" } catch * { "Caught an error" }
```

- Eval

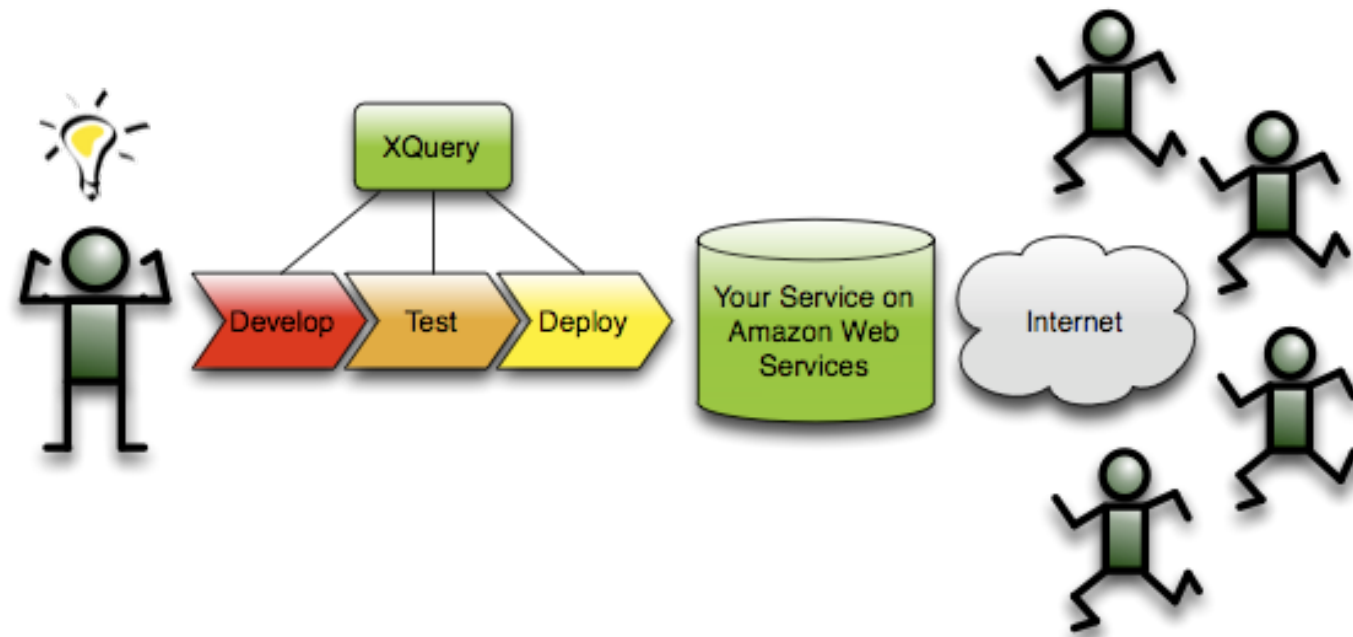
```
let $x := "21" return using $x  
eval { concat($x, "+ xs:double ($x)") }
```

- Command-line Input/Output

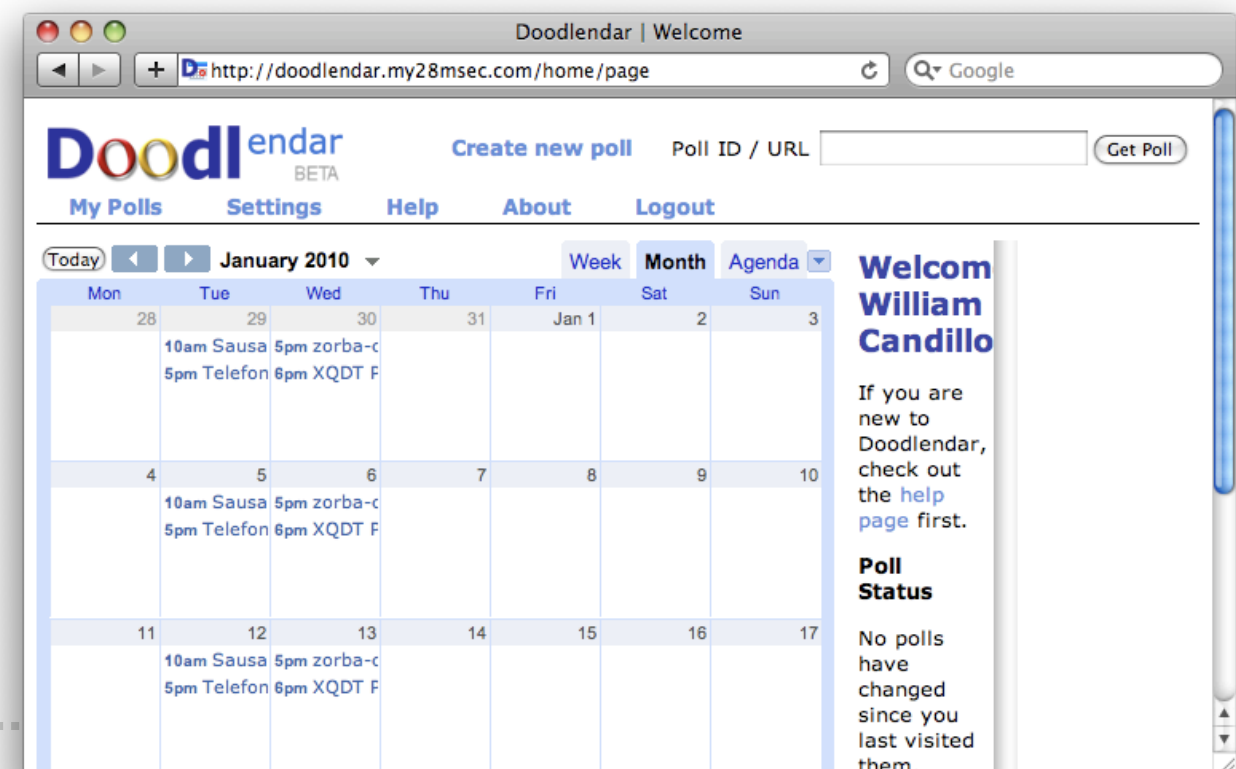
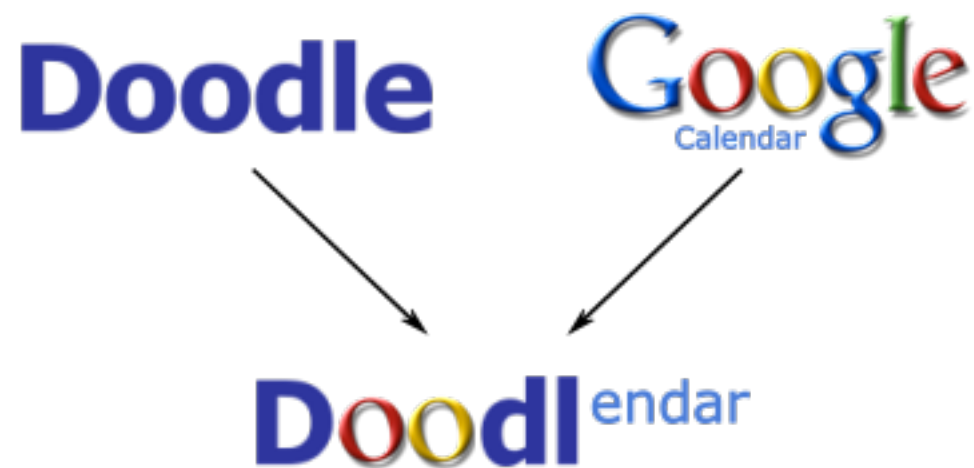
```
let $name := util:read-line()  
return util:print("Hello", $name)
```

~~XQuery has no
Application Server~~

- XQuery Application Server based on Zorba XQuery Processor and Amazon Cloud Computing Stack

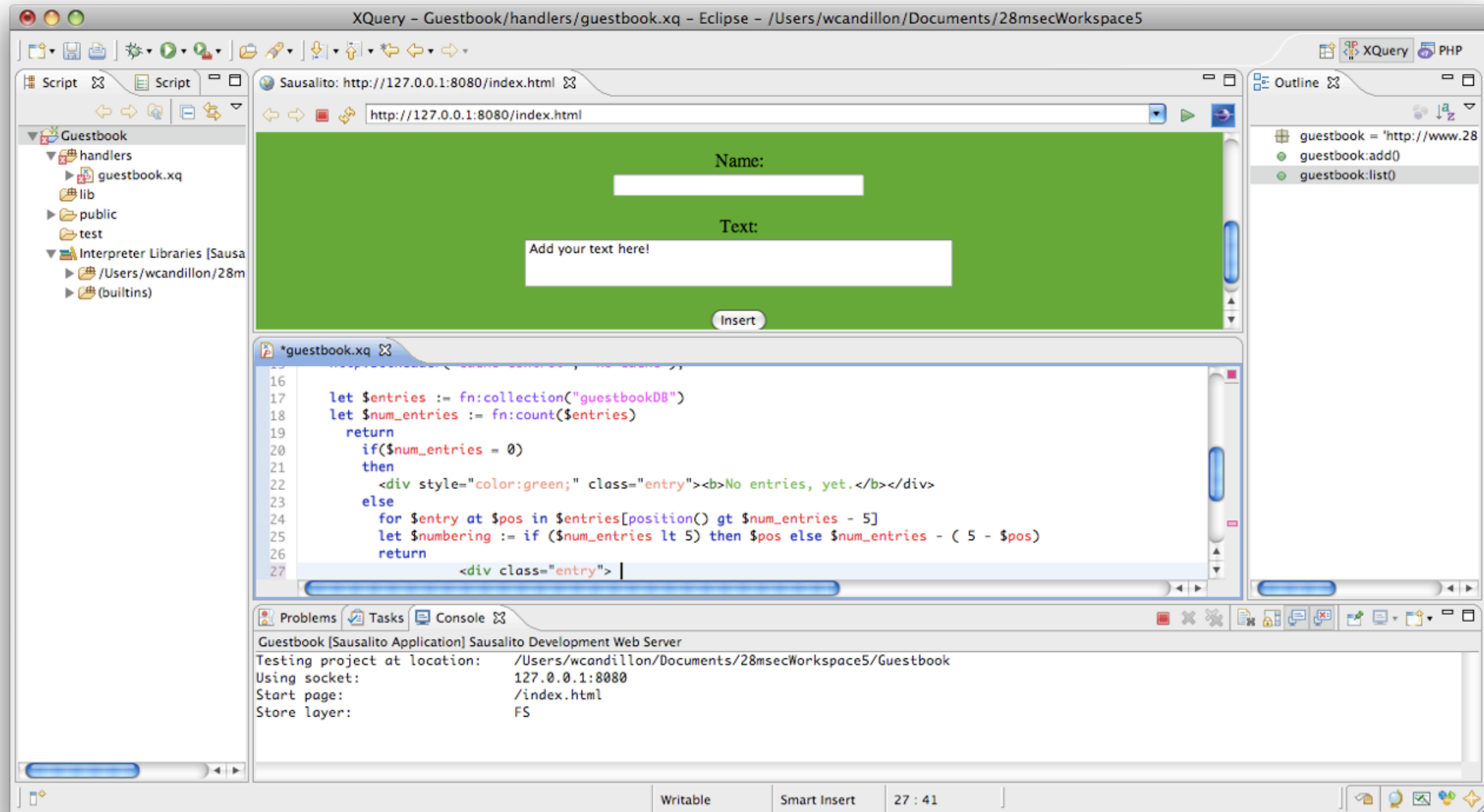


- Success Story: <http://doodlendar.my28msec.com>



~~XQuery has no
Open Source Tools~~

<http://www.eclipse.org/webtools/incubator/>



~~XQuery has no utility
functions for the Web~~

More than **28 modules** in Sausalito library

- HTTP
 - REST
 - Security
 - Sessions
 - OAuth
 - AtomPub
 - Captchas
 - AtomPub
 - Emails
 - Serializer
 - Collections
 - Strings
 - Maths
 - Media
 - Etc...
-

- Compliant with RFC 5023

- AtomPub Server for the Cloud

```
declare function atom:entries(){
  atompub:handle($serviceDocument)
};
```

- Atom Store

```
atomstore:new-feed("/atom/entries",
  <a:feed><a:title>Demo Blog</a:title></a:feed>);
```

- Model helpers

```
declare $e := f:new-entry(util:uuid(), "XQuery Meetup");
f:set-content($e, f:new-content("Go for a drink with XML/
XQuery lovers!"))
```

Thank You!

28msec

Questions ?

contact@28msec.com

More info

<http://sausalito.28msec.com>



Because green is the new blue...

<http://try.zorba-xquery.com>

The screenshot shows a web browser window titled "Try Zorba" with the URL "http://try.zorba-xquery.com/". The page content includes:

- Zorba Live Demo**
> Get started with XQuery (using Zorba 0.9.8)
- Your XQuery**
- Example Queries:**
 - Try-Catch: ▾
 - GroupBy: ▾
 - Eval: ▾
 - node-reference: ▾
 - REST: ▲
 - [retrieve Google website](#)
 - [invoke Google search](#)
 - [use referer header](#)
 - [post with multipart string params](#)
 - [post with xml part](#)
 - [get with tidy and options](#)
 - Window-Clause: ▾
- XQuery (required):**

```
(: Example of zorba-rest:post with string parameters and custom headers :)  
  
import module namespace zorba-rest = "http://www.zorba-xquery.com/zorba/rest-functions";  
  
zorba-rest:post(  
  "http://www.example.com/path",  
  <payload content-type="multipart/form-data">  
    <part name="parameter1">value1</part>  
    <part name="parameter2">value2</part>  
  </payload>,  
  <headers>  
    <header name="custom">custom header value</header>  
  </headers>)
```