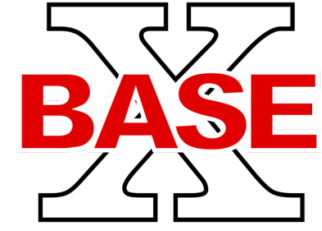
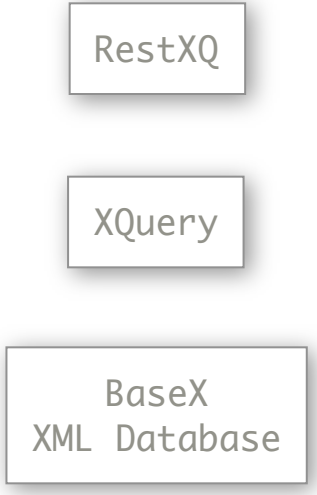


Web Application Development

How to develop RESTQX-based web application with BaseX

Arve Gengelbach & Sabine Teubner

Building XML Web Applications



Prerequisites

- Java
- Maven
  http://maven.apache.org/

- blog example source code stub
  https://github.com/siserle/blog-example
- or start from scratch:
  http://files.basex.org/etc/webapp-stub.zip

Building XML Web Applications

RestXQ

XQuery

BaseX
XML Database

Directory Layout

```
blog-webapp
├── pom.xml  ..................  application configuration
├── WEB-INF
│   └── web.xml  ..............  service configuration
├── data  .....................  XML databases
├── repo  .....................  XQuery library modules
├── restxq  ...................  XQuery RESTXQ handlers
└── static
```

Get BaseX WebApps up and running

Possibilities

- Maven
- BaseX Standalone
- Webapplication Archive (*.war)

① Run Webapp with Maven

```
blog-example $ mvn -U jetty:run
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building Blog Example 0.1-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] >>> jetty-maven-plugin:8.1.4.v20120524:run (default-cli) @ webapp-stub >>>
...
Downloading: http://repo2.maven.org/maven2/org/basex/basex-api/7.5.1-SNAPSHOT/maven-metadata.xml
...
[INFO] Configuring Jetty for project: Blog Example
[INFO] Webapp source directory = /Users/mag/github/blog-example
[INFO] Reload Mechanic: automatic
[INFO] Classes directory /Users/mag/github/blog-example/target/classes does not exist
[INFO] Context path = /
[INFO] Tmp directory = /Users/mag/github/blog-example/target/tmp
[INFO] Web defaults = org/eclipse/jetty/webapp/webdefault.xml
[INFO] Web overrides =  none
[INFO] web.xml file = file:/Users/mag/github/blog-example/WEB-INF/web.xml
[INFO] Webapp directory = /Users/mag/github/blog-example
...
2013-02-06 10:18:56.192:INFO:oejs.AbstractConnector:Started SelectChannelConnector@0.0.0.0:9876
[INFO] Started Jetty Server
```

① Maven starts the application as configured in pom.xml

```
$ cat WEB-INF/web.xml
 ...
  <!--  RESTXQ Service (can b
  <servlet>
    <servlet-name>RESTXQ</serv
    <servlet-class>org.basex.
  </servlet>
  <servlet-mapping>
    <servlet-name>RESTXQ</serv
    <url-pattern>/restxq/*</u
  </servlet-mapping>
```

② Jetty server runs on localhost port 9876
serving what is configured in WEB-INF/web.xml

② Run with BaseX Standalone (BaseX.zip *)

- place .basex in project root
- start server with script http script (basex/bin/basexhttp)

* http://files.basex.org/releases/latest/

```
# BaseX Property File.

# General Options
DBPATH = data
REPOPATH = repo
DEBUG = false
LANG = English
LANGKEYS = false

# Client/Server Architecture
HOST = localhost
PORT = 1984
SERVERPORT = 1984
EVENTPORT = 1985
USER =
PASSWORD =
SERVERHOST =
PROXYHOST =
PROXYPORT = 80
NONPROXYHOSTS =
TIMEOUT = 30
KEEPALIVE = 600
PARALLEL = 8
LOG = true
LOGMSGMAXLEN = 1000

# HTTP Services
WEBPATH = .
RESTXQPATH = restxq
HTTPLOCAL = false
STOPPORT = 8985

# Experimental Options
```

③ Run Webapp compiled as WAR

Compile WAR file with Maven.

```
blog-example mag$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building Blog Example 0.1-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
...
[INFO] Building war: /Users/mag/github/blog-example/target/webapp-stub-0.1-SNAPSHOT.war
[INFO] WEB-INF/web.xml already added, skipping
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 4.933s
[INFO] Finished at: Wed Feb 06 10:22:05 CET 2013
[INFO] Final Memory: 5M/81M
[INFO] ------------------------------------------------------------------------
```

③ Run Webapp compiled as WAR

Test WAR file locally*. Make sure paths fit.

```
$ java -jar jetty-runner-8.1.8.v20121106.jar ~/github/blog-example/target/webapp-stub-0.1-SNAPSHOT.war
2013-02-06 10:25:35.362:INFO:omjr.Runner:Runner
2013-02-06 10:25:35.363:WARN:omjr.Runner:No tx manager found
2013-02-06 10:25:35.426:INFO:omjr.Runner:Deploying file:/Users/mag/github/blog-example/target/webapp-stub-0.1-
SNAPSHOT.war @ /
...
2013-02-06 10:25:39.034:INFO:oejs.AbstractConnector:Started SelectChannelConnector@0.0.0.0:8080
```

* e.g. using http://repo2.maven.org/maven2/org/mortbay/jetty/jetty-runner/

The **Blog Example** shows how to perform some of the basic database tasks using **BaseX** from within a **RESTXQ** module.

Version 1 of the Blog Example

- uses BaseX-specific `db:create()` function to create a new blog database (and `db:drop()` to offer the possibility to delete the blog database).

```
declare %restxq:path("blog-v1/create-database")
        %restxq:GET
        updating function page:create-blog-database() {

 let $blogdb := <blog> ... </blog>

  (: Create the database. :)
  return
    db:create("my-blog", $blogdb, "blog.xml")
};
```

Version 2 of the Blog Example

- uses expressions of the **XQuery Update Facility** (XQUF):
  - to insert postings into the blog: `insert node $insert-node as first into $db/blog/postings`
  - to change postings: `replace node $db/blog/postings/posting [@postingdate=$id] with $posting`
  - to delete postings:

```
declare %restxq:path("blog-v2/delete-posting/{$id}")
        %restxq:GET
        updating function page:delete-posting($id as xs:string) {
  (: Open blog database. :)
  let $db := db:open("my-blog")
  return
    delete node $db/blog/postings/posting[@postingdate=$id]
};
```

Complete Version of the Blog Example

- **Problem:** "Dead ends"/empty pages after executing updating expressions.
- **Reason:** updating expressions do not return any value and it is not allowed to mix **updating and non-updating expressions**.
- **Solution:** A combination of the BaseX `db:output()` function and the `restxq:redirect` xml element.

Complete Version of the Blog Example

- The BaseX-specific db:output() function is the only way to perform updates and return results in a single query.

```
declare %restxq:path("blog-complete/delete-posting/{$id}")
        %restxq:GET
        %output:method("xhtml")
        updating function page:delete-posting($id as xs:string) {

  (: Open blog database. :)
  let $db := db:open("my-blog")

  return
    (db:output("The posting has been successfully deleted."),
     delete node $db/blog/postings/posting[@postingdate=$id])

};
```

Complete Version of the Blog Example

- The `restxq:redirect` element can be used to redirect a client to the specified URL.

```
declare function page:redirect($redirect as xs:string) as element(restxq:redirect)
{
  <restxq:redirect>{ $redirect }</restxq:redirect>
};

declare %restxq:path("blog-complete/delete-posting/{$id}")
        %restxq:GET
        updating function page:delete-posting($id as xs:string) {

  (: Open blog database. :)
  let $db := db:open("my-blog")

  return
    (db:output(page:redirect("/restxq/blog-complete")),
     delete node $db/blog/postings/posting[@postingdate=$id])
};
```