

Botlist Ghost Network - Building a Web Application with a Multi-Language Architecture

Berlin Brown
Founder of Botnode.com Software
berlin.brown at gmail.com
Atlanta, Ga

January 31, 2011

Abstract

This article presents Botlist-Network (Botlist). Botlist is a web application and information retrieval system that aims to collect and analyze content from the vast world wide web. Botlist is an specification for describing the platform, the Botlist web front end and the Botlist backend infrastructure for crawling and analyzing web documents and converting those documents into a machine readable format such as RDF semantic web formats.

This article also looks at the use of several powerful open source programming languages. Each language is used for it strenghs.

Contents

0.1	Overview of Botlist	2
0.2	Java Programming Language and Tools	3
0.2.1	Spring MVC Framework	3
0.2.2	Hibernate Object Relational Mapping and Model Beans	3
0.2.3	Full Text Search with Lucene	4
0.2.4	Developer tools, Eclipse and Emacs	4
0.3	JRuby JVM Language	5
0.3.1	JVM Languages	5
0.3.2	Jython, Scala and Clojure	5
0.4	Scripting and other tasks with Python	6
0.4.1	Code Generation Library	6
0.4.2	Crawler Architecture	6
0.5	Functional Programming and Scala and the Liftweb Web Framework	7
0.6	Network programming with Erlang	8
0.7	Network programming with Erlang	9
0.8	Botlist and Test Driven Development	10
0.8.1	RSpec	10
0.8.2	JUnit	10
0.9	Factor, a Stack Based Language	11
0.10	Named Entity Recognition, Semantic Web, and Botlist Future Directions	12
0.10.1	Amazon EC2 and S3	12

0.1 Overview of Botlist

Empty

0.2 Java Programming Language and Tools

Empty

0.2.1 Spring MVC Framework

Spring is one of the latest Java Enterprise Edition (J2EE) replacement frameworks that is supposed to provide minimal presentation layer support, business logic support as well as the backend, persistence support. It is basically a framework to bind all frameworks. And, on top of everything I just mentioned, it is lightweight. From working with it, it is a lot easier to work with than with the early EJBs.

It was created by Rod Johnson, originally in the year 2000, but recently released over the last few years. It is a Java framework that is growing rapidly. Recently, BEA have been in talks with Interface21 (Spring's Host company) for a possible partnership. The key is light but flexible. What does that mean exactly? Spring is modular, all the Spring libraries are broken up so that you can work with separate parts of your application individually as opposed to tackling everything at once.

With a MVC(Model-View-Controller) application, you are concerned with modular design, separating the View from the Controller and Action logic, but also you need to worry about "Flow". "If the user submits a 'print' command, separate that logic from the 'add' logic, and what pages are presented to the user". Doing this within the limitations of JSP, Servlets, and Taglibs becomes difficult. So, This created a need for frameworks like the Pico container and Spring, Struts.

0.2.2 Hibernate Object Relational Mapping and Model Beans

Our web-application would not be complete without a clear approach for persisting the link data. So we have used the Hibernate ORM (object relational mapping) library to do the backend persistence work for us. It is not really necessary to use Hibernate for such a simple application, but as your enterprise application grows, the need for a more robust persistence mechanism will greatly become evident. MySQL 5.0.2 is used for our database and most of the recent MySQL connector APIs will work with this example.

Almost like Struts, a lot of the hibernate settings are defined in a hibernate configuration file, 'hibernate.cfg.xml' and your hibernate mapping file, 'Botlist.hbm.xml'. Normally the most important settings for your application include what database dialect you are using; we are using MySQL and the definition of your hibernate POJO beans. The simple bean contains an almost one-to-one mapping between your database fields and the Java members, accompanied by the appropriate getters and setters.

0.2.3 Full Text Search with Lucene

0.2.4 Developer tools, Eclipse and Emacs

0.3 JRuby JVM Language

JRuby is a JVM language. Allows for the syntactic sugar of Ruby.

We discussed earlier how Jython is basically used for the backend coding, that includes communicating with Hibernate. Here are the code snippets associated with each of those operations. Most of the code is fairly intuitive; at the heart of the create operation, you must get the Hibernate SessionFactory and initiate a transaction. Once that is done, create an instance of the Hibernate POJO bean and populate the bean with the data from the Struts ActionForm. Once that is taken care of, use the session and transaction object to save the data. The Edit operation probably contains the most code and is separated into two Jython classes.

```
class BotverseController

  def initialize(controller)
    @controller = controller
    @daohelper = @controller.entityLinksDao
  end

  # Generate the view
  def getModel(request)
    # Audit the request
    @controller.auditLogPage(request, "botverse.html")
    query = "from org.spirit.bean.impl.BotListEntityLinks"
    postListings = @daohelper.pageEntityLinks(query,
      nextPage, BotListConsts::MAX_RESULTS_PAGE)
    map = BotListMapEntityLink.new
    map['listings'] = postListings
    return map
  end

  def onSubmit(request, response, form, errors)
    link = @daohelper.readLinkListing(ratingId)
    link.rating = link.rating + 1
    @daohelper.createLink(link)
    return form
  end
end
```

0.3.1 JVM Languages

0.3.2 Jython, Scala and Clojure

0.4 Scripting and other tasks with Python

Empty

0.4.1 Code Generation Library

0.4.2 Crawler Architecture

0.5 Functional Programming and Scala and the Liftweb Web Framework

Scala is a JVM language.

0.6 Network programming with Erlang

Empty

0.7 Network programming with Erlang

Empty

0.8 Botlist and Test Driven Development

0.8.1 RSpec

0.8.2 JUnit

0.9 Factor, a Stack Based Language

Empty

0.10 Named Entity Recognition, Semantic Web, and Botlist Future Directions

Empty

0.10.1 Amazon EC2 and S3

Bibliography

- [1] Joe Armstrong. *Programming Erlang*. The Pragmatic Bookshelf, 2007.
- [2] Mehmet Bilgi and Haluk G. An architecture design process using a supportable meta-architecture and roundtrip engineering. In *Advances in Information Systems*, volume 4243 of *LNCIS*, pages 324–333. Springer, 2006.
- [3] James Gosling, Bill Joy, Guy Steele, and Gilad Bracha. *The Java Language Specification*. Addison-Wesley, second edition, 2000.
- [4] Paul Hudak. *The Haskell School of Expression, Learning Functional Programming Through Multimedia*. Cambridge University Press, 2000.
- [5] Rod Johnson, Juergen Hoeller, Alef Arendsen, Thomas Risberg, and Colin Sampaleanu. *Professional Java Development with the Spring Framework*. Wiley Publishing, Inc, 2005.
- [6] Lasse Koskela. *Test Driven, Practical TDD and Acceptance TDD for Java Developers*. Manning Publications Co., 2008.
- [7] Toby Segaran. *Programming Collective Intelligence*. O’Reilly Media, Inc, 2007.
- [8] Peter Seibel. *Practical Common Lisp*. Apress, 2005.
- [9] Dave Thomas. *Programming Ruby The Pragmatic Programmer’s Guide*. The Pragmatic Bookshelf, second edition, 2005.
- [10] Simon Thompson. *The Craft of Functional Programming*. Addison Wesley Longman Limited, second edition, 1999.