caBIG®
cancer Biomedical Informatics Grid®

Grid Without Java: JRuby On Rails

Rapid Application Development using JRuby On Rails & caGrid Services

Author: Mark Vance

July 2011

Goals

Rapid Application Development & caGrid

Investigate new technologies that give users the ability to quickly develop a UI for a Grid application.

- Try to auto-generate commonly used functionality.
- Use languages that are popular and/or easy to learn.
- Compatibility with JVM.
- Meet/Exceed performance standards.

Provide installable modules to users that provide common caGrid functionallity.

- SyncGTS
- Dorian Authentication
- Role Management
- CQL Query Building

caBIG cancer Biomedical Informatics Grid

Ruby

Why Ruby?

Dynamic language that is growing in popularity. Designed to be a “dense” programming language that maintains the ability to be easily readable. Strives to follow the “Principal of Least Astonishment”.

Features:

- Object-Oriented (ex. inheritance, mixins and metaclasses)
- Dynamic Typing
- Package Management using RubyGems
- Interactive Ruby Shell (irb)
- Available on all major platforms
- Large Standard Library (RDoc)
- Unique and easy-to-use Iterators and block syntax (.each)
- Exception Handling
- Automated Testing

Drawbacks:

- Not compatible with JVM (caGrid)

caBIG cancer Biomedical Informatics Grid

JRuby

Why JRuby?

JRuby is a 100% Java implementation of the Ruby programming language. It is Ruby for the JVM.

Features:

- Runs on the JVM
- Completely compatible with previously written Java programs, such as caGrid (require “*.jar”, require “java”)
- It offers access to **both** Java and Ruby libraries
- All of the features of Ruby (gems, dynamic typing, etc.)
- Compatible with Rails
- Compatible with Interactive Ruby Shell (irb)
- JRuby supports interpreted mode, AOT mode, and JIT mode
- Use of Foreign Function Interface (FFI) to allow the use of C-libraries bundled as gems.

Safe to say anything available in Ruby is available to JRuby

caBIG cancer Biomedical Informatics Grid

SyncGTS Example

```
require "java"

include_class "gov.nih.nci.cagrid.common.Utils"
include_class "gov.nih.nci.cagrid.syncgts.bean.SyncDescription"
include_class "gov.nih.nci.cagrid.syncgts.core.SyncGTS"

class SyncGridTrust

    def self.synchronizeOnce(syncDescriptionFile)
    success = false
    begin
      pathToSyncDescription = syncDescriptionFile
      description = Utils.deserializeDocument(pathToSyncDescription, SyncDescription)
      SyncGTS.getInstance().syncOnce(description)
      success = true
    rescue
      puts "SyncGridTrust Error: " + $!
    end
    return success
  end

end
```


caBIG cancer Biomedical Informatics Grid

JRuby Example

Example Service Class (in Ruby)

In this class, there are three “attributes” (fields) of an unspecified type. Using the function **attr_accessor** provides each attribute with “getters” and “setters” functions for each attribute.

```
class Service
  attr_accessor :address, :name, :description
end
```

Example Use of Module

Modules can be used to make entire java packages available.

This also increases readability when using common class names, like “Object”.

```
module CqlQuery
  include_package "gov.nih.nci.cagrid.cqlquery"
end
```

```
client = DataServiceClient.new(serviceEndpoint)
query = CqlQuery::CQLQuery.new
target = CqlQuery::Object.new()
target.setName(objectName)
query.setTarget(target)
```


caBIG cancer Biomedical Informatics Grid

JRuby Example

```
def processResults(results)
  serviceArray = Array.new
  unless results.nil?
    results.each { |serv|
      @service = Service.new
      @service.address = serv.getAddress().toString()
      begin
        metadata = MetadataUtils.getServiceMetadata(serv)
        @service.description = metadata.getServiceDescription().getService().getDescri
        @service.name = metadata.getHostingResearchCenter().getResearchCenter().getDis
        serviceArray << @service
      rescue Exception
        puts "MetadataUtils Error: ", $!
      end
    }
  end
  return serviceArray
end
```

- results.nil? is Ruby convention for Boolean methods
- .each is the universal iterator method of Ruby
- Service “setter” being called for each attribute
- MetadataUtils is a caGrid class being instantiated in Ruby
- “<<“ used to push object into the array

caBIG cancer Biomedical Informatics Grid

Rails

Why Rails?

Rails is the web framework build on Ruby that provides most of the desired functionality for rapid application development.

Referred to as Ruby on Rails, Rails is credited for making Ruby a popular language.

Many popular sites are built in Ruby on Rails (Twitter, Hulu, Github, & many more).

Designed to integrate web applications and databases.

Features:

- Scaffolding of DB Models, Views, Controllers (MVC Layout)
- Using Gems to add functionality (Popular Gems)
- Active Community (Tutorials, Forums, Conferences)
- Ability to write Ruby code directly into Views.
- Use of HTML/CSS
- Interactive Ruby Shell (irb)
- Structured around forms with built-in functionality.
- Rails Server
- Rails “Routes” (RESTful)

caBIG cancer Biomedical Informatics Grid

Rails Commands

%> **rails new caGrid_WebApplication**

- Creates a new rails application called “caGrid_WebApplication”
- Includes all necessary files to start the web application

%> **rails generate scaffold role name:string description:string**

- Creates domain class with attributes “name” & “description”
- Creates controller with CRUD functions
- Creates views for each function provided by the controller

%> **rake db:migrate**

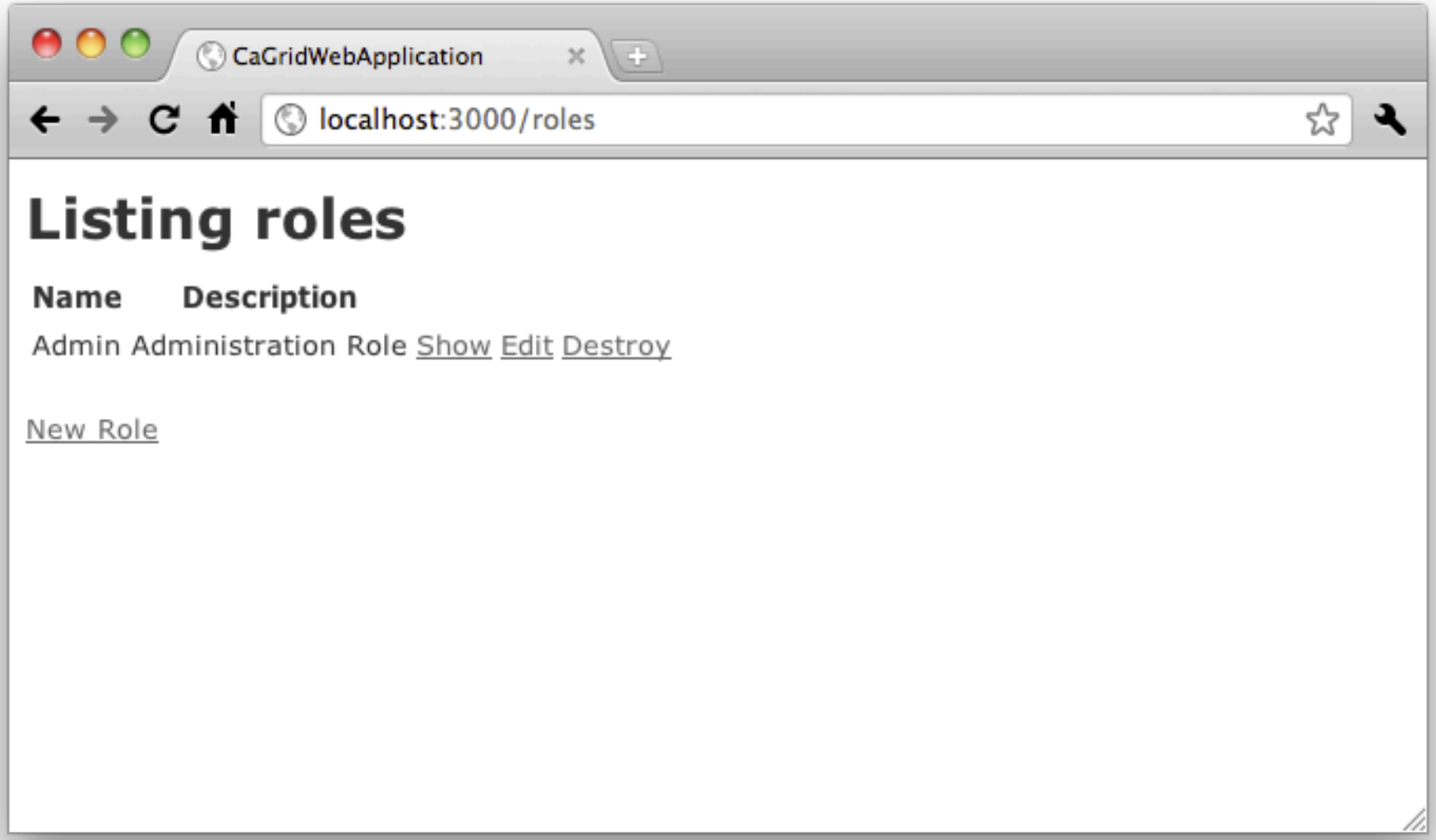
- Creates database table “ROLE” with corresponding attributes

%> **rails server**

- Starts the rails server on localhost:3000
- Has the built-in UI to Add, Edit, Delete and Search the provided database

caBIG cancer Biomedical Informatics Grid

caGrid WebApplication



Rails Server

Output in Rails Server from Adding Administrator Role

```
Started POST "/roles" for 127.0.0.1 at Tue Jul 19 12:53:23 -0400 2011
  Processing by RolesController#create as HTML
  Parameters: {"utf8"=>"✓", "authenticity_token"=>"v/
MzXrPGVtC87+tyG5nJk93gdksQPZripN3H2tpog60=", "role"=>{"name"=>"Admin",
"description"=>"Administration Role"}, "commit"=>"Create Role"}
  SQL (1.0ms)   SELECT name
 FROM sqlite_master
 WHERE type = 'table' AND NOT name = 'sqlite_sequence'
  AREL (0.0ms)  INSERT INTO "roles" ("name", "description", "created_at", "updated_at")
VALUES ('Admin', 'Administration Role', '2011-07-19 16:53:24.002000', '2011-07-19
16:53:24.002000')
Redirected to http://localhost:3000/roles/1
Completed 302 Found in 111ms
```

Routes – Browser locations

Method Call – Controller method used to process request

Form Variables – Variables pulled in from the submitted form

Database Entry – Table/Attribute descriptions and values submitted

caBIG cancer Biomedical Informatics Grid

Ruby Code in Rails Views

```
<div id="user_nav">
  <% if current_user %>
    Logged in as <b><%= current_user.username.capitalize %></b>.
  <% else %>
    <%= link_to "Sign Up", sign_up_path %> or <%= link_to "Log In", log_in_path %>
  <% end %>
</div>
<div id="xinner">

  <ul id="globalnav">
    <% require "lib/MenuTabBuilder.rb" %>

    <% tabs_tag(:builder => MenuTabBuilder) do |tab| %>
      <%= tab.home 'Sign Up', root_path %>
      <%= tab.login 'Log In', log_in_path %>
      <%= tab.roles 'Roles', roles_path %>
    <% end %>
  </ul>
<br/>

<% flash.each do |name, msg| %>
  <div id=error_explanation h2>
    <%= content_tag :div, msg, :id => "flash_#{name}" %>
  </div>
<% end %>

<%= yield %>
```


caBIG cancer Biomedical Informatics Grid

Dependencies (Gemfile)

Using Gems to Resolve Dependencies

- Gemfile lists the gem dependencies of the project
- “Bundle install” command will install all gem dependencies
- Gems can have dependencies of their own that will automatically be resolved.

Example:

- Create a gem called “AuthenticateDorian” that requires caGrid JARS
- “AuthenticateDorian” requires a gem “SyncGTS”
- SyncGTS requires different caGrid JARS and a cron job gem called “Whenever”

Add the line:
gem ‘AuthenticateDorian’ to a user’s Gemfile

- Download and install “AuthenticateDorian”, “SyncGTS” and “Whenever” gems
- Give the user the ability to authenticate with Dorian and periodically run SyncGTS
- Possibly provide commands to auto-generate associated models, views and controller

caBIG cancer Biomedical Informatics Grid

JRuby On Rails & caGrid

Tutorial posted on caGrid.org providing some basic caGrid functionality:

- SyncGTS
- Dorian Authentication
- Role Management
- Searching available Data & Analytical Services
- Build CQL queries and running them against available Data Services

Possibilities?

- Create gems for specific caGrid functionality (SyncGTS, Dorian Authentication, CQL Query building)
- Host git repository so users can easily install gems into their own web applications
- Allow users to create and post their own gems

Resources

Links

- http://www.jruby.org/ - JRuby Official Site
- http://railscasts.com/ - Quality Screencasts explaining how to implement popular functionality in Rails
- http://stackoverflow.com/ - Rails Support with ranked answers
- http://railsforum.com/ - Rails Support Forum

Questions? Comments? Suggestions?

email: Mark.Vance@osumc.edu

caBIG cancer Biomedical Informatics Grid