

One Pager: Testing Scripting Support in GlassFish v3

Overview

- [1. Introduction](#)
 - [2. JRuby Testing Scenarios](#)
 - [3. Grail Testing Scenarios](#)
 - [4. Admin/Config](#)
 - [5. Document Review](#)
 - [6. Reference documents](#)
-

1. Introduction

This one pager describes test spec for scripting support for GlassFish V3 Application Server <https://glassfish-scripting.dev.java.net/> .

1.1. Names and e-mail of Document Authors:

Ming Zhang: Ming.Zhang@sun.com, Sherry Shen: Sherry.Shen@sun.com

1.2 Date of This Document:

02/11/08

1.3 JRuby Based Framework Coverage :

- JRuby on Rails
- Grails

1.4 Due to resource issue, the following areas will **not** be covered in SQE testing:

- Merb
- Jython/Django

2. JRuby Testing Scenarios

2.1. GlassFish V3 Rails Sniffer and Connector:

Create JRuby helloworld application with controller/view and deploy the application to V3 with directory deployment. V3 should be able to detect the JRuby on Rails application and deploy successfully.

2.2 GlassFish V3 Rails Grizzly adapter:

Deploy the cookbook Jruby/Rails application. Testing the ability of receives requests/response from Grizzly layer directly and delegation of requests to Rails framework.

2.3. JRuby application deployment:

2.3.1. JRuby application directory deployment.

Deploy the cookbook Jruby/Rails application directly to V3 with directory deployment.

2.3.2. JRuby application war deployment.

Use warble to create a war file out of JRuby cookbook application and deploy to V3.

2.4. Scaffolding of a JRuby application:

2.4.1. Create a JRuby bookstore application using Scaffolding of JRuby on Rails.

2.4.2. Populate the database with db:migrate .

2.4.3. Test the validities of Scaffolding JRuby application.

Deploy the JRuby bookstore application and test the functionality of the app created from JRuby Scaffolding.

2.5. JRuby on Rails Packaging and configuration with GF V3:

2.5.1. Pre-install JRuby on Rails

Pre-install JRuby on Rails and required gems (Rails and MySQL JDBC drivers) for JRuby testing.

2.5.2. JRuby IPS package for GlassFish Updatecenter

Download JRuby package from GlassFish Updatecenter. The JRuby release, bundled with required gems (Rails and MySQL JDBC drivers), is delivered in IPS package. This IPS bundle can be installed on existing GlassFish installation using GlassFish updatecenter tool. Testing the JRuby on Rails with the installation from Update Center.

2.6. Deploy an Enterprise JRuby application:

2.4.1. Download the flexible project management application Redmine. It multiple projects & makes it possible to have per-project forums/wikis.

2.4.2. Populate the MySQL database with db:migrate .

2.4.3. Deploy the Redmine application to V3 in directory deployment and test the functionality of the app.

2.4.4. Create a war file using warble gem

2.4.5. Deploy the war file and test the functionality.

3. Grails Testing Scenarios

3.1. Grails Production War

Grails production war is generated with Grails provided scripts ("*grails war*").

The war can be deployed to GlassFish with `asadmin` command.

The tests create Grails application, package production war, deploy war, and use web client to verify application work correctly.

3.2. Grails War without Dependent jars.

Grails production war without Grails dependency ("*grails shared-war*" in v3 prelude with `SharedWar.groovy`, but TBD in v3).

The war can be deployed with `asadmin` command with `--libraries` option in v3 prelude, but `--libraries` will be removed in v3 with the help of `gf-grails-connector.jar`.

The tests create Grails application, package war, deploy war, and verify application usage.

3.2. Grails Native Deployment without War

The GFv3 provides groovy script carries native deployment ("*grails run-app*").

It runs grails application during development using GlassFish embedded API.

This native deployment allows dynamic change and avoids the war package for fast development.

The tests create Grails application, deploy non-packaged application, modify application in its development environment, verify the dynamic change of application without re-deployment.

4. Admin/Config

4.1 The Admin GUI provides the user interface to configure JRuby on Rails applications. Please refer Admin GUI test spec for testing details.

4.2 Using `asadmin deploy --property` option to set properties of JRuby. The properties should be persisted in `domain.xml`. Please see Admin CLI test spec for details.

4.3 JRuby container monitoring is supported through the monitoring infrastructure. The JRuby Container will provide various probes and telemetry objects to emit and make the monitoring data available. Please refer Monitoring test spec for testing details.

5. Document Review

5.1 GlassFish JRuby on Rails guide.

5.2 Quick Starting guide

6. Reference Documents

6.1 One Pager: Scripting support in GlassFish v3:

<http://wiki.glassfish.java.net/attach/V3FunctionalSpecs/Scripting-one-pager.html>
