XQuerRail Framework Getting Started Guide

Created by: Gary Vidal

Created: 5/7/2012 6:36:00 AM Last Modified: 5/8/2012 8:03:00 AM

Table of Contents:

- I. Installation Instructions
- II. Configuration Setup
- III. Setting up a new Application
- IV. Configuring and changing your routes
 - a. Setting up resource routes
 - b. Setting up controller routes and parameters
- V. Configuring and changing interception configuration
 - a. Configuring ml-security interceptor
 - b. Configuring profiling interceptor
 - c. Configuring logging interceptor
- VI. Managing and editing your application domain file.
 - a. Domain settings
 - b. Configuring namespaces
 - c. Creating Models
 - d. Creating Controllers
- VII. Creating and managing unmanaged controllers
- VIII. Using base models and extending base models
 - IX. Creating Views and Templates
 - a. XQuerrail View Architecture
 - b. Using the XQuerRail templating language
 - c. Creating and Managing Tag Libraries
 - d. Creating and managing application resources
 - i. controller-scripts
 - ii. controller-stylesheets
 - iii. controller-images
 - e. Using helpers
 - i. formbuilder-helper.xqy
 - ii. javascript-helper.xqy
 - iii. jquery-helper.xqy
 - iv. kendo-helper.xqy

Appendix

- A. Request/Response Pipeline
- B. Framework directory structure
- C. Framework files and resources
- D. Application files directory structure

Installation instructions

- Checkout source code from http://www.github.com/garyvidal/xquerrail
- Create application database named for your application and database settings.
- Ensure anonymous user and anonymous-role have been defined. These are used to access login splash screen, so ensure permissions are set accordingly.
- Create an http application server
 - Set the root directory to /path-to-your-framework-src-file. (Ex "/code/xquerrail/src/")
 - o Use FileSystem not modules database.
 - Use application-level security
 - o Set user to anonymous (anonymous user requires certain permissions to be set. Please
 - Set URL Rewriter to /_framework/rewriter.xqy
- Access your application from HTTP url. Should see login splash screen.
- Login using your elevated credentials(not anonymous).

Setting up your new application:

There are a few important things to setup if creating a new application from scratch. First you must configure your application in /_config/config.xml. Take the following steps to ensure proper setup:

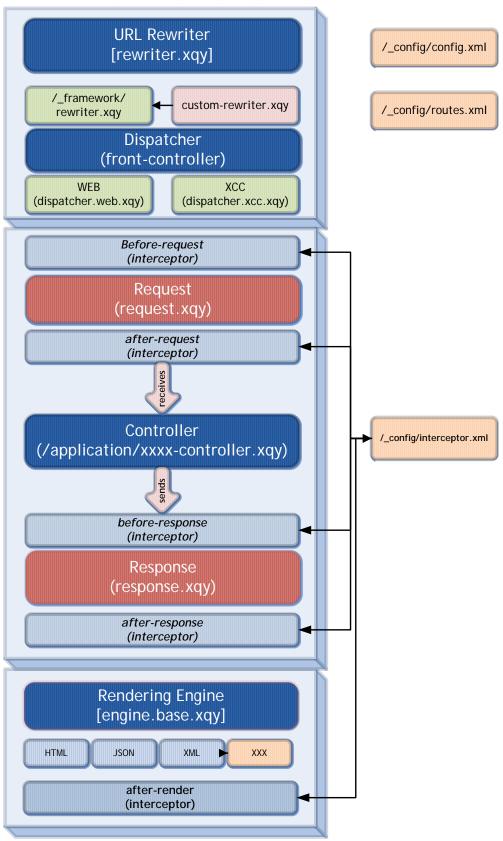
 Make sure you configure your application settings. The following template is setup in your config.xml file. You can modify these settings to change these to reflect your applications settings. Each application must be registered in order for system to route information accordingly.

- 2. Modify your application-domain.xml file in \${application-name}/domain/application-domain.xml to add new models and controllers or drop in controller files into /\${application-name}/controller/ directory.
 - a. The controller filename should be in form \${controller.name}-controller.xqy.

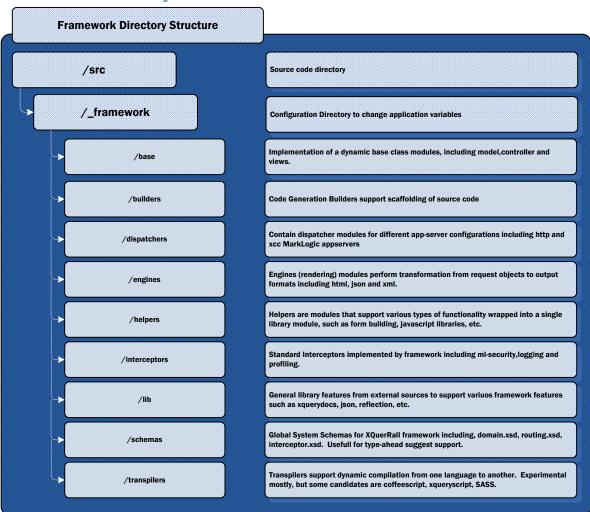
```
ex. country-controller.xqy
  person-controller.xqy
```

- b. Your module library namespace should be in form: \${application-namespace}/controller/\${controller-name}
 - ex. "http://my-app-namespace/controller/country"
 http://my-app-namespace/controller/person

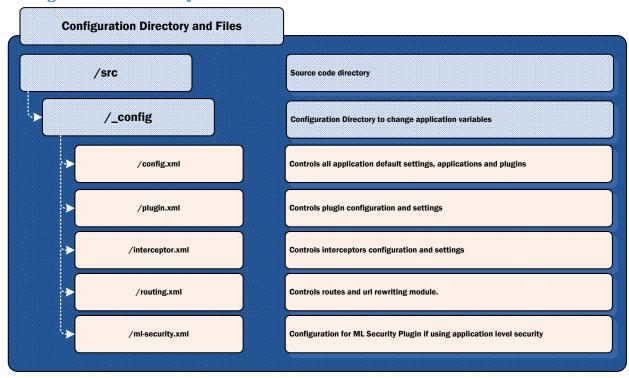
Understanding the Request/Response pipeline



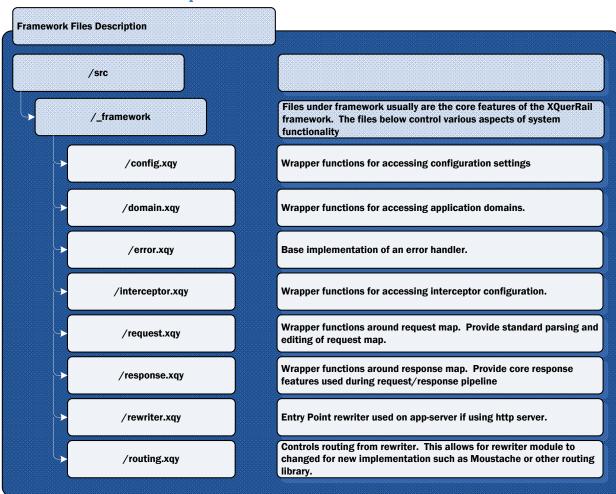
Framework Directory Structure



Configuration Directory Structure



Framework Files Description



Application Directory Structure

Application Directory Structure /\${application-name} Application Directory. Can have directory for each application Main application controllers. The dispatcher will call corresponding named controller /controllers module. Controllers for application usually in \$controller-name[dot|dash]controller.xqy Files should be formatted as \${controller-name}-controller.xqy /xxxx-controller.xqy Namespace: \$(app-namespace)/controller/\$(controller-name) Shortcut Key:\$(application-name):controller:\$(controller-name):action(\$(params)) /domain Controls all dynamic scaffolding for application in application-domain.xml /application-domain.xml Controls all dynamic scaffolding for application in application-domain.xml /lib Libraries directory commonly shared in your application code. Files should be formatted as \${library-name}-lib.xqy /xxxx-lib.xqy Namespace: \${app-namespace}/lib/\${library-name} Shortcut Key:\${application-name}:lib:\${library-name}:{function-name}(\${params}) /models **Models for CRUD and Search operations** Files should be formatted as \${model-name}-lib.xqy Namespace : \${app-namespace}/model/\${model-name} /xxxx-model.xqy Shortcut Key :\${application-name}:model:\${model-name}:(function-name)(\${params}) Resources should be common web resources or code that you would like to include in your /resources application. Also certain tags will look in this folder for assets such as <?controller-script?> or <controller-stylesheet?> Application Specific Resources including scripts, stylesheets and images /scripts Script directory used to dynamically insert <?controller-script?> tag. Ensure your file is named exactly as your controller name /stylesheets Stylesheet directory used to dynamically insert <?controller-stylesheet?>. Filename must match name of controller in order for it to be included. /images Basic image library for application /tags Application Tag Libraries. Tags are controlled by using specific names and creating them in your application directory as xxxx-tag.xqy. When a tag with names matches will call the tag render method. /views Views directory with each controllers views listed under their name. The view filename should be in the form of \$action.\$format.xgy Controller Views are stored under the controller name folder and can correspond to a given /\${controller-name} action or a named view you would like to render from controller. The response:set-view() function will proper view routing.