Restful服务框架(Athena) - 移植老服务用户向导

作者：李艳鹏

日期: 2015.05.09

[Restful服务框架(Athena) - 移植老服务用户向导 1](#_Toc418926531)

[引言 2](#_Toc418926532)

[背景 2](#_Toc418926533)

[本地调试环境配置 2](#_Toc418926534)

[CI集成一键部署配置 6](#_Toc418926535)

[FAQ 9](#_Toc418926536)

# 引言

如果你正在使用Athena Restful服务框架构建一个新的后台服务，请参考同文件夹下的另一份文档：*Restful服务框架(Athena) - 构建新服务用户向导.docx*

这份文档为老服务移植向导，目标读者是业务开发人员。。

# 背景

Athena Restful服务框架最适用于之前使用Spring Web MVC开发的服务项目，这些项目除了使用Spring Web MVC以外，还内部集成了嵌入式的Jetty或者Tomcat，并且集成了部分的配合上线下操作的CI脚本，以及Spring容器，HTTP容器，GC配置，线程池配置和属性管理等功能，这些框架级别功能全部耦合在业务代码中，这大大的增加了业务开发人员的开发成本和维护成本，如果你的项目正处于这种情况，Athena Restful服务框架则是你的最佳选择，它通过合理的分层来整理这些框架级别的需求，解放业务人员使其不必再关注框架级别的技术细节，使业务人员能够集中精力解决业务架构的问题，并且通过简单的几步骤就可以完成老业务到Athena Restful服务框架的移植，并且可以很容易的配置本地调试环境和远程CI集成的环境达到一键发布的功能。

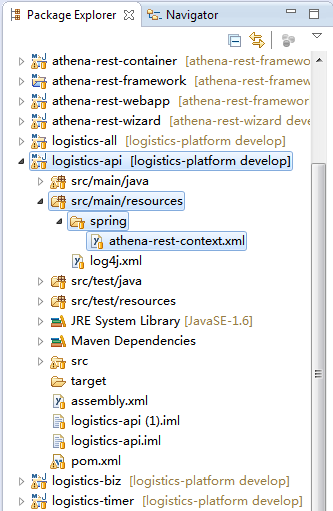
# 本地调试环境配置

1. 首先，业务项目与Athena Restful服务框架只需要一个单一的集成点，那就是Spring的跟环境文件，在你业务项目的主项目中导出如下跟环境文件：

spring/athena-rest-context.xml

Athena Restful服务框架默认查找此Spring环境文件并加载业务服务，你只需要将老的Spring环境移动到此类路径位置即可。

以物流项目为例：



1. 在你的主项目中的pom.xml添加对Athena Restful服务框架的依赖：

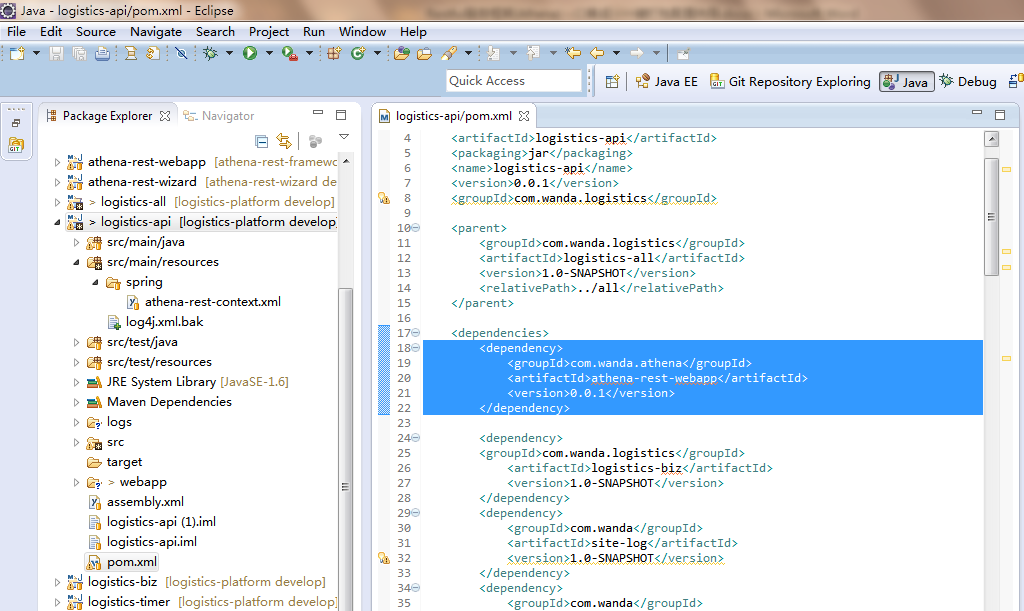
<dependency>

<groupId>com.robert.athena</groupId>

<artifactId>athena-rest-webapp</artifactId>

<version>0.0.1</version>

</dependency>

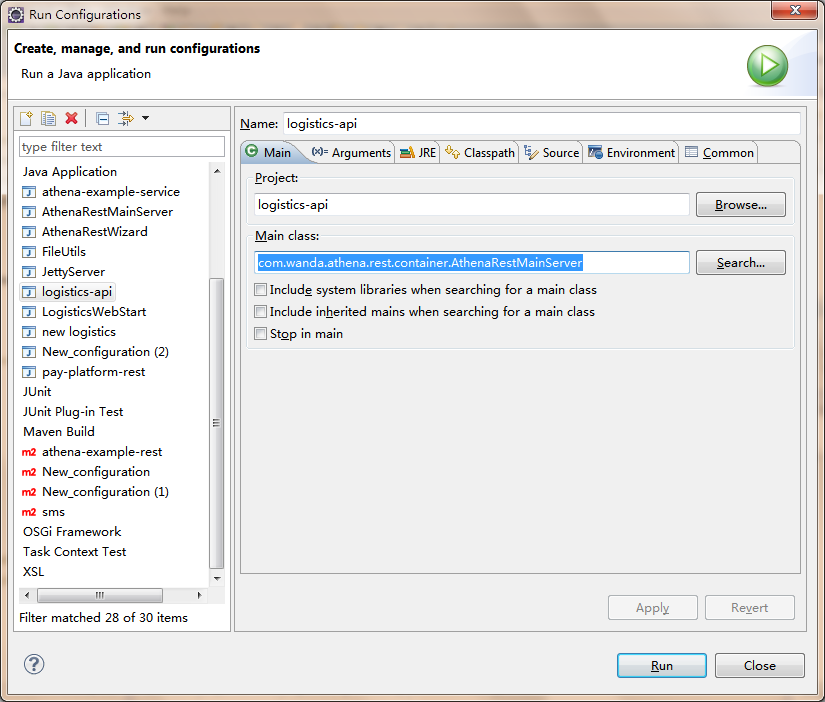


1. 为你的主项目配置启动选项：

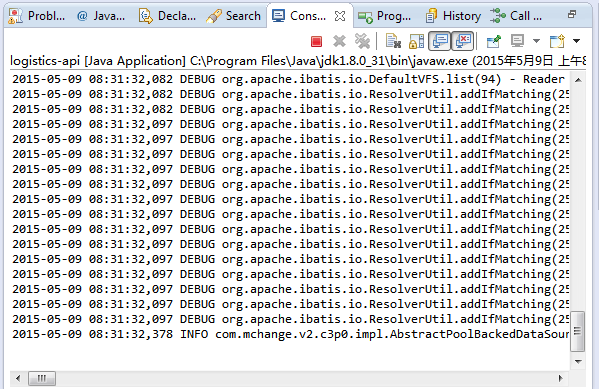
Name: logistics-api

Project: logistics-api

Main class: com.robert.athena.rest.container.AthenaRestMainServer

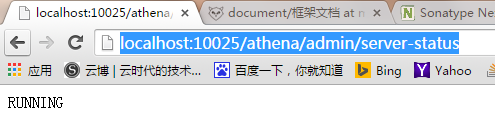


1. 此时，点击右下角的执行，在控制台窗口可以看见服务已经启动：



1. 这时，我们的服务已经构建完毕，你可以通过浏览器查看服务状态：

服务地址：<http://localhost:10025/athena/admin/server-status>



# CI集成一键部署配置

1. 在主项目的pom.xml中增加以下profiles和build配置：

<profiles>

<profile>

<id>dev</id>

<properties>

<env>dev</env>

</properties>

<activation>

<activeByDefault>true</activeByDefault>

</activation>

</profile>

<profile>

<id>test</id>

<properties>

<env>test</env>

</properties>

</profile>

<profile>

<id>integrate</id>

<properties>

<env>integrate</env>

</properties>

</profile>

<profile>

<id>release</id>

<properties>

<env>release</env>

</properties>

</profile>

</profiles>

<build>

<filters>

<filter>src/main/filters/filter-${env}.properties</filter>

</filters>

<resources>

<resource>

<directory>src/main/resources</directory>

<filtering>true</filtering>

</resource>

</resources>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-jar-plugin</artifactId>

<configuration>

<archive>

<manifest>

<addClasspath>true</addClasspath>

<mainClass>com.robert.athena.rest.container.AthenaRestMainServer</mainClass>

</manifest>

</archive>

</configuration>

</plugin>

<plugin>

<artifactId>maven-assembly-plugin</artifactId>

<configuration>

<appendAssemblyId>false</appendAssemblyId>

<encoding>utf-8</encoding>

<descriptors>

<descriptor>${basedir}/assembly.xml</descriptor>

</descriptors>

<outputDirectory>${basedir}/../distribution/</outputDirectory>

</configuration>

<executions>

<execution>

<id>make-assembly</id>

<phase>package</phase>

<goals>

<goal>single</goal>

</goals>

</execution>

</executions>

</plugin>

</plugins>

</build>

1. 在主项目的根目录新建文件assembly.xml，并填入如下内容：

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<assembly xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/assembly-1.0.0.xsd"*>

<id>package</id>

<formats>

<format>tar.gz</format>

</formats>

<includeBaseDirectory>false</includeBaseDirectory>

<dependencySets>

<dependencySet>

<outputDirectory>lib</outputDirectory>

<useProjectArtifact>true</useProjectArtifact>

<scope>runtime</scope>

</dependencySet>

</dependencySets>

</assembly>

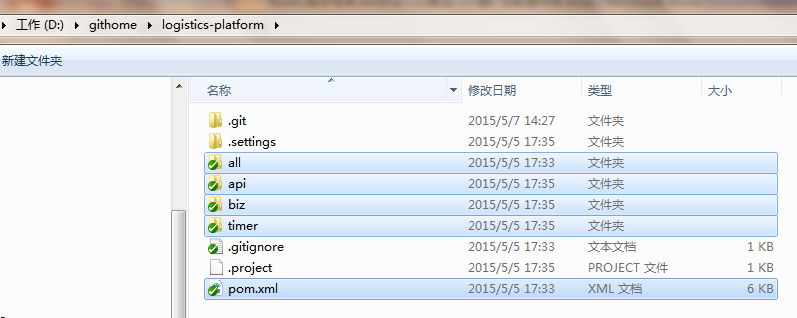
1. 到此为止，CI集成的工作在业务项目中已经完成，这些配置将把所有的依赖jar包打成tar.zip，由后续的CI集成流程处理，到此阶段没有任何容器的侵入，没有任何脚本的侵入，业务项目保持Java项目的整洁度和简单性。

可以使用如下命令验证打包：*mvn package*

1. 如果你还负责上线属性配置和CI一键发布的集成，请参考同文件夹下文档：*Restful服务框架(Athena) - CI集成 - 一键打包配置向导.docx*。

# FAQ

1. Athena Restful服务框架只兼容在项目的根目录存在父pom.xml文件的项目结构，因为这种视觉上的树型结构看起来最合理，如果把父项目放入单独文件夹看起来不太合理，也不实用。



1. 建议同时阅读同文件夹下的*Restful服务框架(Athena) - 构建新服务用户向导.docx*，这两份文档相辅相成，如果你遇到一些问题，可能在另一份文档中找到答案。