Mtools1.0v使用说明

2014/7/21

张广海

修改记录：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 内容 | 修改人 | 修改日期 |
| 1.0v | 新增使用手册 | 张广海 | 2014/7/22 |
|  |  |  |  |
|  |  |  |  |

目录

[1.Mtools介绍 4](#_Toc394673095)

[1.1插件结构设计 5](#_Toc394673096)

[２．插件中各包的用处 5](#_Toc394673097)

[2.1　com.mtools.core.plugin 5](#_Toc394673098)

[2.2　com.mtools.core.plugin.annotation 5](#_Toc394673099)

[2.3　com.mtools.core.plugin.aop.aspect 5](#_Toc394673100)

[2.4　com.mtools.core.plugin.auth 5](#_Toc394673101)

[2.5　com.mtools.core.plugin.constant 6](#_Toc394673102)

[2.6　com.mtools.core.plugin.db 6](#_Toc394673103)

[2.7　com.mtools.core.plugin.ehcache 6](#_Toc394673104)

[2.8　com.mtools.core.plugin.entity 6](#_Toc394673105)

[2.9　com.mtools.core.plugin.entiy.vo 6](#_Toc394673106)

[2.10　com.mtools.core.plugin.freemark 6](#_Toc394673107)

[2.11　com.mtools.core.plugin.helper 6](#_Toc394673108)

[2.12　com.mtools.core.plugin.lucene 6](#_Toc394673109)

[2.13　com.mtools.core.plugin.mail 6](#_Toc394673110)

[2.14　com.mtools.core.plugin.monitor 6](#_Toc394673111)

[2.15　com.mtools.core.plugin.notify 7](#_Toc394673112)

[2.16　com.mtools.core.plugin.optlog 7](#_Toc394673113)

[2.17　com.mtools.core.plugin.properties 7](#_Toc394673114)

[2.18　com.mtools.core.plugin.security 7](#_Toc394673115)

[2.19　com.mtools.core.plugin.solr 7](#_Toc394673116)

[2.20　com.mtools.core.plugin.spring.resolver 7](#_Toc394673117)

[2.21　com.mtools.core.plugin.staticres 7](#_Toc394673118)

[2.22　com.mtools.core.plugin.sys 7](#_Toc394673119)

[2.23　com.mtools.core.plugin.task 7](#_Toc394673120)

[2.24　com.mtools.core.plugin.web.filter 7](#_Toc394673121)

[2.25　com.mtools.core.plugin.web.interceptor 8](#_Toc394673122)

[2.26　com.mtools.core.plugin.web.jcaptcha 8](#_Toc394673123)

[2.27　com.mtools.core.plugin.web.listener 8](#_Toc394673124)

[2.28　com.mtools.core.plugin.web.servlet.init 8](#_Toc394673125)

[2.29　com.mtools.core.plugin.netty.client 8](#_Toc394673126)

[2.30　com.mtools.core.plugin.netty.server 8](#_Toc394673127)

[3．下列配置文件的用处 8](#_Toc394673128)

[3.1 .src/main/resources 8](#_Toc394673129)

[1、common-mvc.xml 8](#_Toc394673130)

[2、mtools.params.properties 8](#_Toc394673131)

[3、mtools.default.properties 9](#_Toc394673132)

[4、log4j.properties.template 9](#_Toc394673133)

[5、mtools-monitor.xml 9](#_Toc394673134)

[6、mtools.beans.xml 9](#_Toc394673135)

[7、mtools.cache.xml 9](#_Toc394673136)

[8、alipay-druid-datasource.xml 9](#_Toc394673137)

[9、mtools.datasource01.xml 9](#_Toc394673138)

[10、mtools.datasource02.xml 9](#_Toc394673139)

[11、mtools.task.xml 9](#_Toc394673140)

[3.2 conf目录下的template 10](#_Toc394673141)

[1. core-config.xml.template 10](#_Toc394673142)

[2. db.params.properties.template 10](#_Toc394673143)

[3. ehcache.xml.template 10](#_Toc394673144)

[4. ext.dic.template 10](#_Toc394673145)

[5. IKAnalyzer.cfg.xml 10](#_Toc394673146)

[6. log4j.properties.template 10](#_Toc394673147)

[7. schema.xml.template 10](#_Toc394673148)

[8. stopword.dic 10](#_Toc394673149)

[9. web-mvc.xml.template 10](#_Toc394673150)

[10. web-root.xml.template 11](#_Toc394673151)

[11. web.params.properties 11](#_Toc394673152)

[12. web.xml.template 11](#_Toc394673153)

[4.使用教程 12](#_Toc394673154)

[１．打开eclipse，分别导入两个maven工程：mparent和mtoools 12](#_Toc394673155)

[２．新建maven工程（也就是将要开发的项目工程） 12](#_Toc394673156)

[３．加入依赖工程mtools 14](#_Toc394673157)

[４．复制配置文件 15](#_Toc394673158)

[５．根据项目需要修改配置 15](#_Toc394673159)

[6 .　设置数据源参数 16](#_Toc394673160)

[7.设置web.params.propertices参数 17](#_Toc394673161)

[8.配置web.xml 17](#_Toc394673162)

[5.关于作者 18](#_Toc394673163)

# 1.Mtools介绍

Mmtools是一个开源的，插件式的，多功能的Java开发工具框架，mtools 的目的是简化spring项目的配置，但不重复造轮子，充分利用现有的一些框架技术,适用于Java se项目以及Java ee 项目，该插件整合了springMVC框架，以及ehcache、freemarker、lucene/solr等开源框架，简单简洁，各种项目中应用到的技术，都在该插件中融合，不需要做过多复杂的配置，项目中已经对各种配置进行了整合，仅仅需要在新的工程中，引入core-config.xml配置文件即可，当然，可以自由配置各自项目中需要的个性化配置，对于初学或者初用ehcache、freemarker、lucenu、solr等技术的开发人员来说，插件中已经提供了现成的使用方法，只要依葫芦画瓢即可，插件中还提供了常用的工具包，如安全方面的加解密，签名验签的工具类，字符串处理，字节处理，日期处理等等日常开发中应用到的基本工具包

## 1.1插件结构设计



# ２．插件中各包的用处

## 2.1　com.mtools.core.plugin

插件基类，除web包下面的插件以外，其他的插件都可以继承该基类，以便具备一些基本的能力

## 2.2　com.mtools.core.plugin.annotation

该包下是一些自定义的注解，如：免登录注解@AuthLogin 免权限验证注解@AuthAccess

以及获取当前登录用户对象注解：@CurrentUser

## 2.3　com.mtools.core.plugin.aop.aspect

切面编程例子

## 2.4　com.mtools.core.plugin.auth

登录授权验证、角色管理等

## 2.5　com.mtools.core.plugin.constant

插件用到的常量

## 2.6　com.mtools.core.plugin.db

数据库工具

## 2.7　com.mtools.core.plugin.ehcache

ehcache缓存工具

## 2.8　com.mtools.core.plugin.entity

插件用到的实体类

## 2.9　com.mtools.core.plugin.entiy.vo

用于返回页面显示的视图类

## 2.10　com.mtools.core.plugin.freemark

页面静态化工具类，用于生成静态html

## 2.11　com.mtools.core.plugin.helper

常用工具箱，如日期处理，xml处理，json处理，加解密处理等等

## 2.12　com.mtools.core.plugin.lucene

lucene搜索工具类，用于建立索引，全文搜索

## 2.13　com.mtools.core.plugin.mail

邮件工具，用于发送邮件通知

## 2.14　com.mtools.core.plugin.monitor

系统监控插件，监控系统内存，线程数等等

## 2.15　com.mtools.core.plugin.notify

通知服务，如一般邮件通知，系统异常通知

## 2.16　com.mtools.core.plugin.optlog

系统日志插件，用于记录访问痕迹

## 2.17　com.mtools.core.plugin.properties

系统参数配置

## 2.18　com.mtools.core.plugin.security

系统安全插件，用于加解密

## 2.19　com.mtools.core.plugin.solr

solr搜索客户端插件，用于建立solr索引，全文查询

## 2.20　com.mtools.core.plugin.spring.resolver

使用springresolver的方式自定义注解

## 2.21　com.mtools.core.plugin.staticres

静态文件，html生成插件

## 2.22　com.mtools.core.plugin.sys

## 2.23　com.mtools.core.plugin.task

系统自动任务插件，用于执行某些定时任务

## 2.24　com.mtools.core.plugin.web.filter

系统过滤器，如异常过滤器，缓存过滤器

## 2.25　com.mtools.core.plugin.web.interceptor

系统拦截器，如session拦截器

## 2.26　com.mtools.core.plugin.web.jcaptcha

验证码插件

## 2.27　com.mtools.core.plugin.web.listener

系统监听器

## 2.28　com.mtools.core.plugin.web.servlet.init

系统初始化servlet，用于初始化基础信息，配置参数等

## 2.29　com.mtools.core.plugin.netty.client

高性能异步请求netty客户端

## 2.30　com.mtools.core.plugin.netty.server

高性能异步请求netty服务端

# 3．下列配置文件的用处

## 3.1 .src/main/resources

## 1、common-mvc.xml

springMvc公共配置文件，在mvc配置文件中引入该公共配置，该配置已经包含了一些基础拦截器的配置，视图的配置，静态资源映射等等

## 2、mtools.params.properties

系统基础属性，如，线程池的参数值

## 3、mtools.default.properties

默认配置，在新的工程里面应当覆盖这些参数

## 4、log4j.properties.template

log4j配置文件模版，每一个应用应该有且只有一个log4j配置文件

## 5、mtools-monitor.xml

aop配置文件

## 6、mtools.beans.xml

该组件用到的基本bean配置

## 7、mtools.cache.xml

系统缓存配置文件，对ehcache的配置

## 8、alipay-druid-datasource.xml

数据库的配置，事务配置等，使用了druid数据连接池

## 9、mtools.datasource01.xml

数据库的配置，事务配置等

## 10、mtools.datasource02.xml

数据库的配置，事务配置等

## 11、mtools.task.xml

自动任务配置，无特殊情况，无需修改该配置文件

## 3.2 conf目录下的template

在建立新的工程时，把该目录的template复制到新工程的src/main/resources目录下，并把“.template”后缀去掉，然后修改相关参数即可使用

## core-config.xml.template

总配置文件，该配置文件引入了该组件所有需要的配置，

当新的工程引入该组件时，直接在新工程的配置文件中引入该配置文件即可，此时不需要在引入下面的配置文件

## db.params.properties.template

数据库配置，新工程中，修改相应的数据库用户名密码地址数据库类型即可

## ehcache.xml.template

新项目中需要用到的缓存配置，仿照配置中的testCache配置，复制并修改缓存名即可完成添加新的缓存配置

## ext.dic.template

IKAnalyzer分词器的拓展词汇字典，可以在该文件中添加新的词汇

## IKAnalyzer.cfg.xml

IKAnalyzer分词器的核心配置，直接复制过去即可（）

## log4j.properties.template

log4j日志模版，如果只是简单的打印日志，直接去掉“.template”即可使用

## schema.xml.template

solr collection 中的conf目录下的schema,此模版加入了IKAnlyzer分词

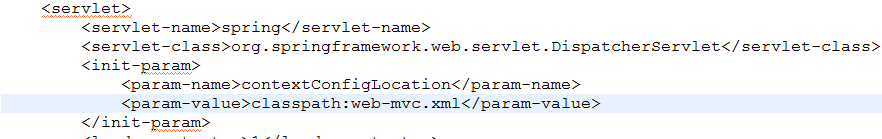
## stopword.dic

IKAnalyzer分词器的默认停止词典，不必修改，直接复制过去即可

## web-mvc.xml.template

spring mvc 配置，在web.xml中将引入该配置文件

如下图：



## web-root.xml.template

web工程中引入配置文件的入口，在web.xml中配置

如下图：



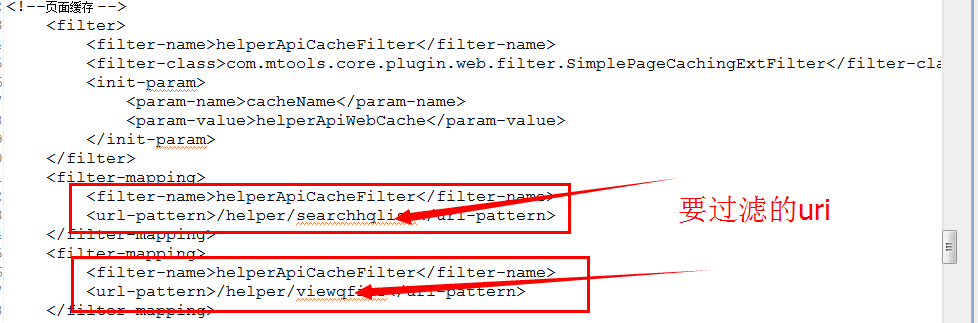
## web.params.properties

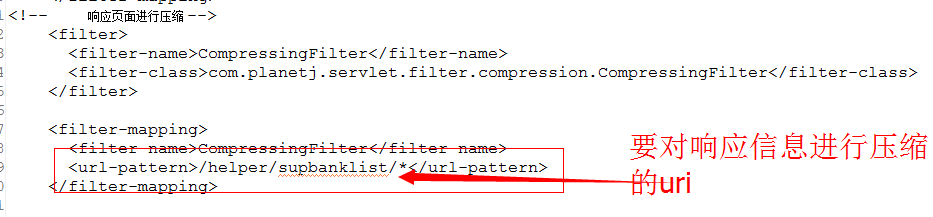
该属性配置文件，包含一些如：静态资源目录的配置，缓存地址的配置，solr地址的配置等等，如果需要使用到相关功能，请去掉注释，并设置相应的值

## web.xml.template

web.xml配置文件，已经配置了spring的相关配置，以及缓存，压缩的配置等，但如果需要使缓存和压缩响应生效，须根据实际情况，修改里面的uri

如下图：

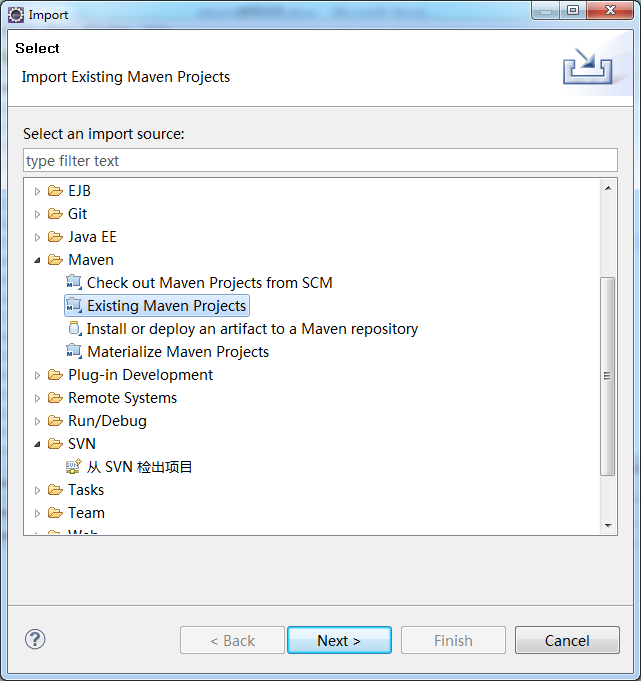




# 4.使用教程

## １．打开eclipse，分别导入两个maven工程：mparent和mtoools

注意，需要以maven工程的形式导入，如下图：



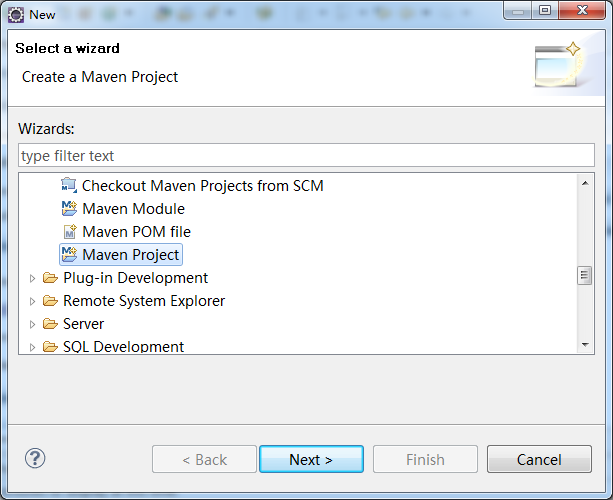
温馨提示：

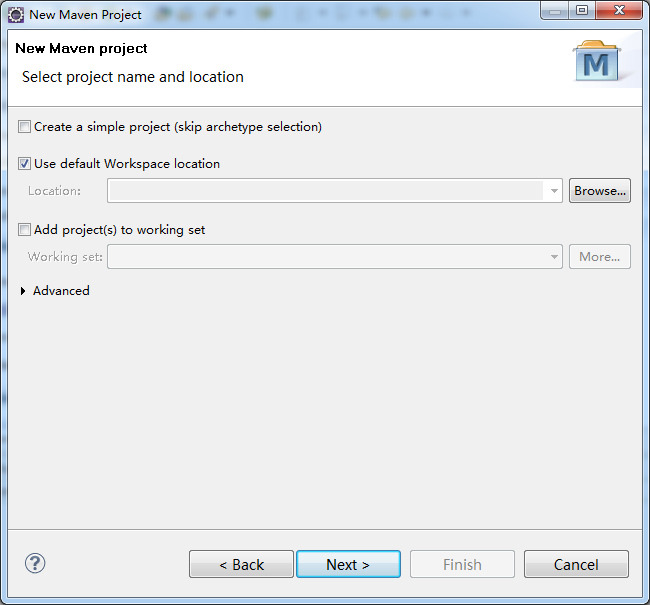
初次导入这两个工程，可能会由于本地资源仓库缺少一些包，或者maven第一次下载太多依赖jar时，导致其中一些jar没能完成下载成功，而导致工程报错

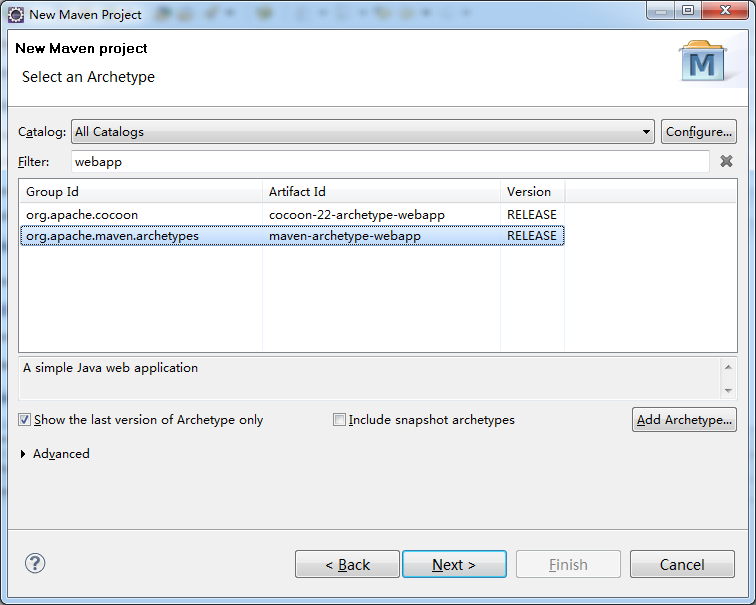
此时，从pom配置文件中找出缺少的jar包，然后到本地资源仓库中把该jar的目录删除，这个时候再次update dependence，那么maven会重新下载jar，一般再次下载的时候便可以成功

## ２．新建maven工程（也就是将要开发的项目工程）

如图：



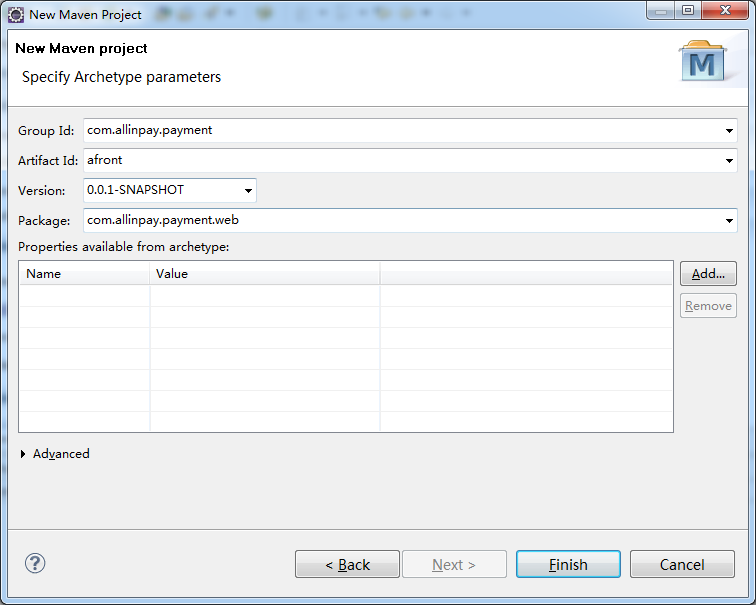




此处新建一个web工程，选择group id为：org.apache.maven.archetypes

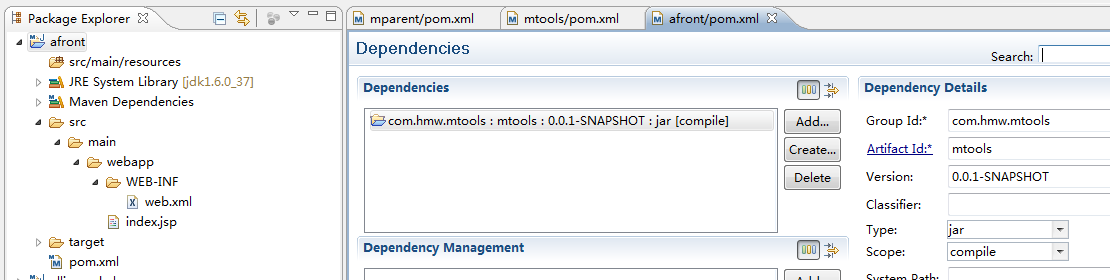
Artifactid 为 maven-archetype-webapp

然后下一步



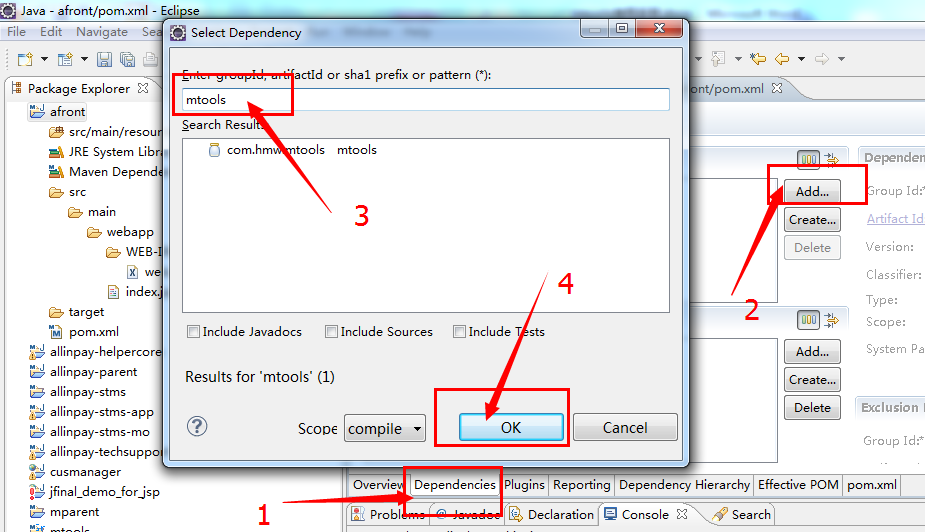
此处填写新工程的groupid artifactid 等信息，具体名字自己定义，然后finish

此时，eclipse会建立一个标准的maven webapp工程，如下图：

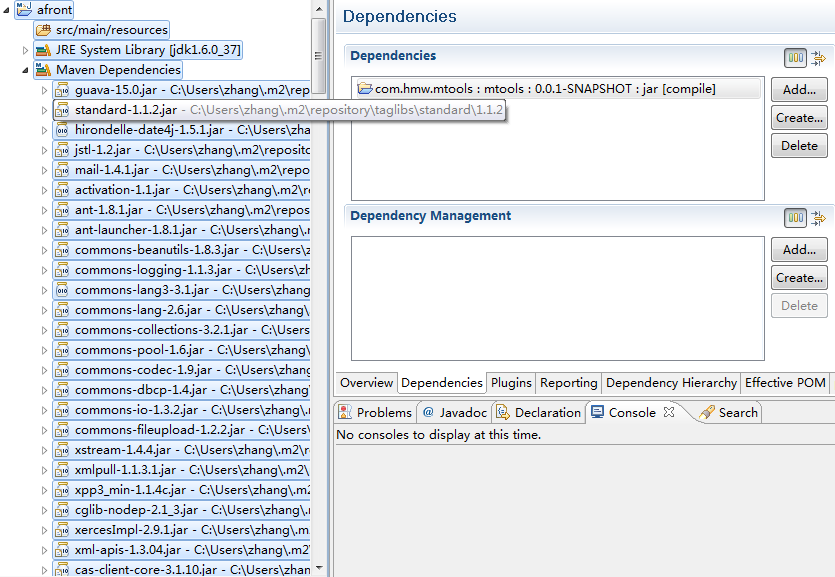


## ３．加入依赖工程mtools

以maven pom.xml的打开方式，打开afront工程下的pom.xml文件，然后点击Dependecices,然后点击add，输入mtools，选择search result 下的选项，然后点击ok

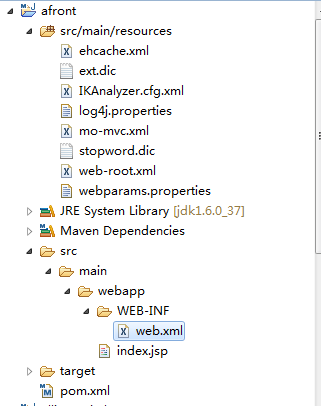


此时会发现，afront工程的maven dependence中把mtools所依赖的jar都关联进来了，包括mtools本身



## ４．复制配置文件

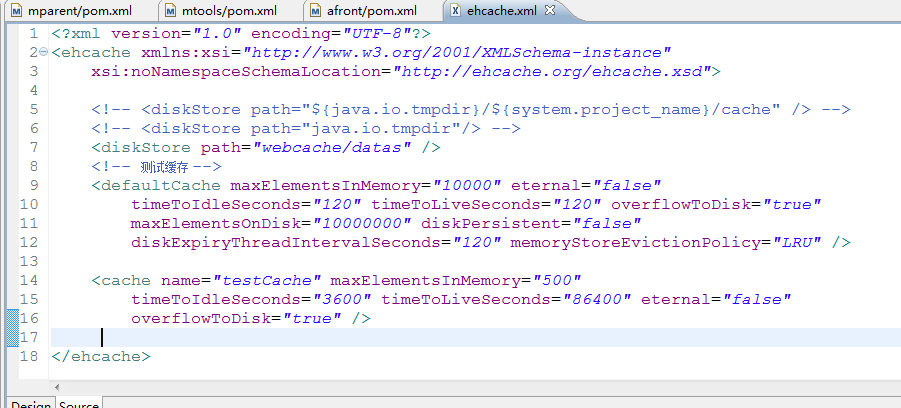
复制mtools工程下conf目录中的template文件到新工程afront的src/main/resources目录下，把”.template”后缀去掉，另外，把web.xml覆盖到WEB-INF下面的web.xml如下图：



## ５．根据项目需要修改配置

５．１新增项目需要的cache

参考已有的testCache配置，复制，修改名称即可



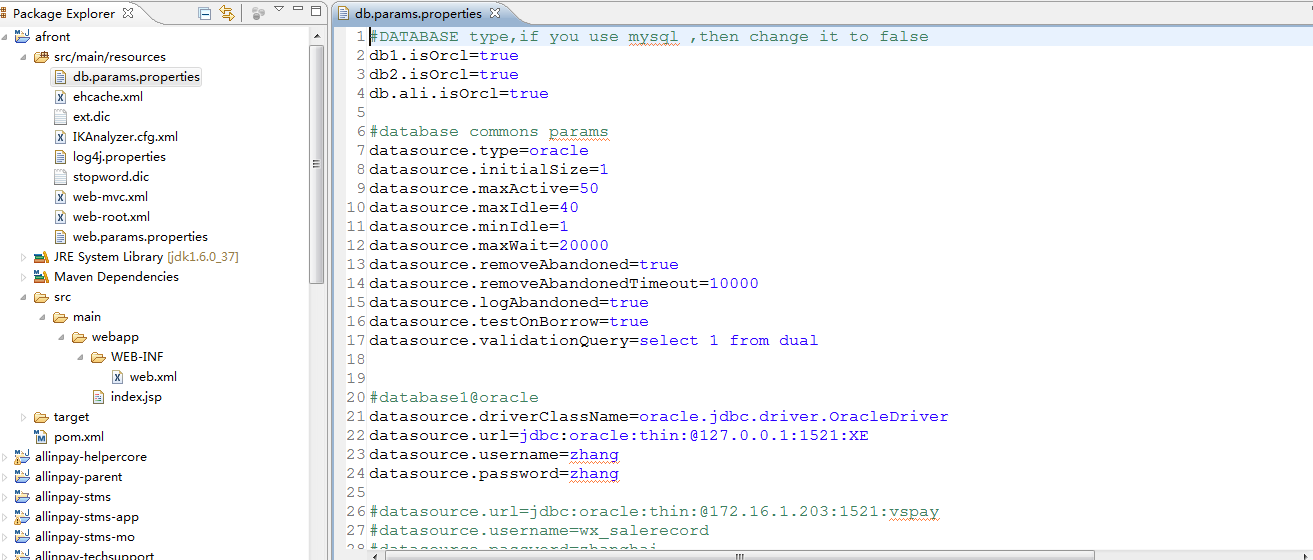
５．２修改web-mvc.xml

配置web工程controller包路路径，以及/跳转的视图



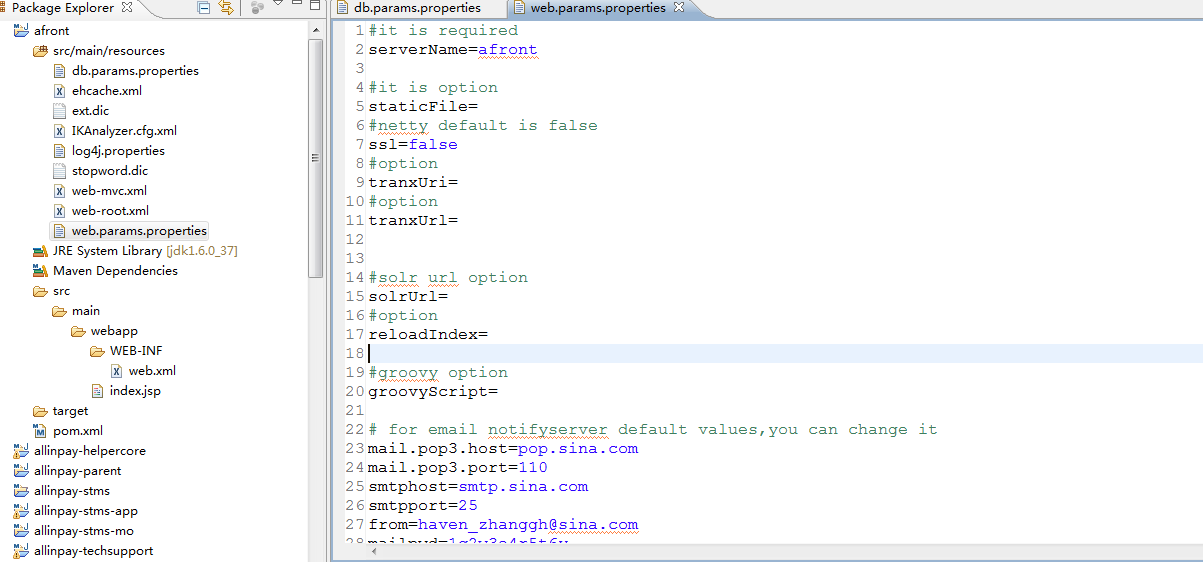
## 6 .　设置数据源参数

根据自己的环境，配置自己的数据库用户名地址等



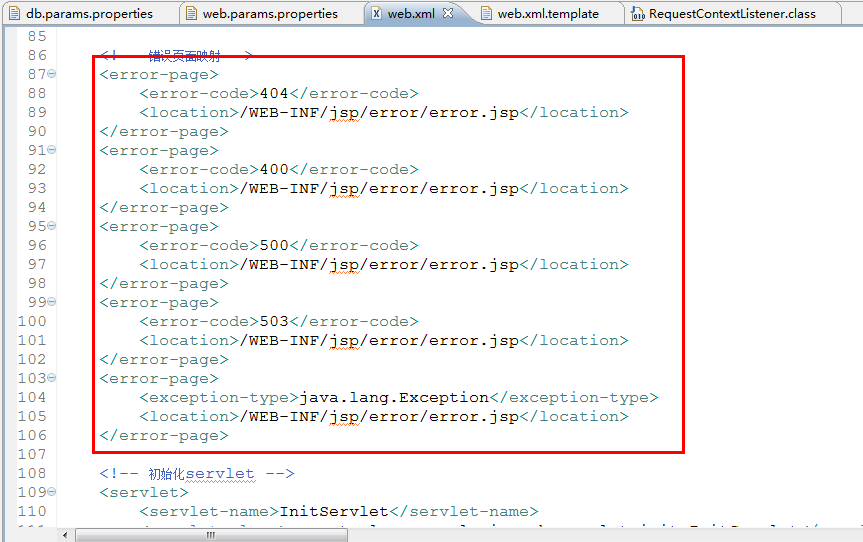
## 7.设置web.params.propertices参数

如下图，注释为option的参数均为可选参数，默认为空值



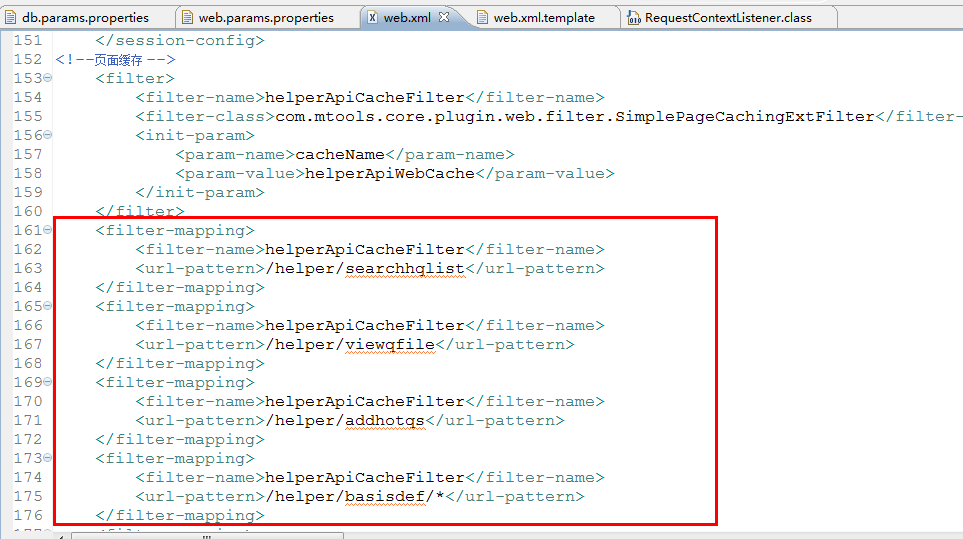
## 8.配置web.xml

a.根据项目需要，配置错误返回视图



b.根据需要，配置web缓存

注意，此处的缓存名必须先在ecache.xml文件中配置



C.配置服务端响应压缩过滤器



如上图是指过滤uri为：/helper/supbanklist/下面的所有请求，此时符合该过滤要求的响应都会以GZIP的压缩格式返回，大大提升了客户端的响应速度

到目前为止，整个项目的配置就完毕了

# 5.关于作者

Blog ： <http://kanckzhang.iteye.com/>

Email : [kanckzhang@163.com](mailto:kanckzhang@163.com)

Github: <https://github.com/zhanggh/mtools>