**Public CMS 2016**

使用手册

目录

[第一章 PublicCMS介绍 1](#_Toc450682097)

[1. 概述 1](#_Toc450682098)

[2. 版本更新 1](#_Toc450682099)

[3. 获取PublicCMS 1](#_Toc450682100)

[4. 目录结构 2](#_Toc450682101)

[第二章 开发工具集成 3](#_Toc450682102)

[1. 工程导入 3](#_Toc450682103)

[2. 开发工具配置 9](#_Toc450682104)

[3. 工程调试 13](#_Toc450682105)

[4. 工程发布 18](#_Toc450682106)

[第三章 二次开发 23](#_Toc450682107)

[1. 工程代码简介 23](#_Toc450682108)

[2. 开发流程 24](#_Toc450682109)

[3. 获取支持 29](#_Toc450682110)

# PublicCMS介绍

## 概述

PublicCMS是采用2016年最新主流技术开发的免费开源JAVACMS系统。商用免费，架构科学。无需任何数据库优化，即可支持上千万数据；支持全站静态化，SSI，0xml配置，扩展指令自动加载，动态页面局部静态化等为您快速建站，建设大规模站点提供强大驱动，也是企业级项目产品原型的良好选择。

PublicCMS同时支持动态站点，静态站点的建设。

PublicCMS支持多站点建设,各个站点之间数据、模板完全分离。

PublicCMS开源版本采用最宽松的授权协议MIT。这意味着:

* 你可以自由使用，复制，修改，可以用于自己的项目。
* 可以免费商用。
* 唯一的限制是必须包含许可声明。

## 版本更新

### 待完成任务

1. 插件管理
2. 抽奖管理
3. 抽奖用户管理
4. 投票管理
5. 推荐位前台投稿
6. 内容前台投稿
7. 用户注册
8. 用户评论集成

## 获取PublicCMS

代码托管

开源中国:<http://git.oschina.net/sanluan/PublicCMS>

GITHUB:<https://github.com/sanluan/PublicCMS>

CSDN:<https://code.csdn.net/zyyy358/publiccms>

如果您没有使用过gradle或者maven，并且不打算使用它们，也可以在PublicCMS-war中war\WEB-INF\lib目录下复制PublicCMS依赖的库

## 目录结构

|  |
| --- |
| data  database  publiccms-by-gradle  publiccms-by-maven  LICENSE  README.md |

* data目录

data/ publiccms/文件夹中包含以下目录

|  |
| --- |
| indexes  staticfile  template |

其中indexes目录为数据全文索引文件，staticfile目录为静态化文件和资源文件存放目录，template为模板文件目录

* database目录

其中的Database Init.sql为PublicCMS项目数据库初始化脚本， Database Change Log.sql为数据库变更记录，随着PublicCMS版本的不断更新，Database Change Log文件中将有大量的数据结构变更记录，Database Init则是当前最新版本的初始化数据，旧版本升级到新版本可以选择直接用最新初始化数据覆盖旧数据结构，也可以选择执行Database Change Log中的变更sql更新。

* publiccms-by-gradle为Gradle管理的工程
* publiccms-by-maven为Maven管理的工程
* LICENSE为PublicCMS授权协议
* README.md为markdown语言编写的简单介绍文档

# 开发工具集成

以下步骤Eclipse为例，下载地址 <https://www.eclipse.org/downloads/>，具体版本信息如下:

Eclipse Java EE IDE for Web Developers.

Version: Luna Service Release 2 (4.4.2)

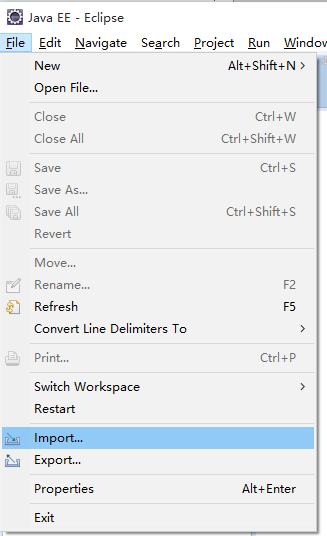
Build id: 20150219-0600

## 工程导入

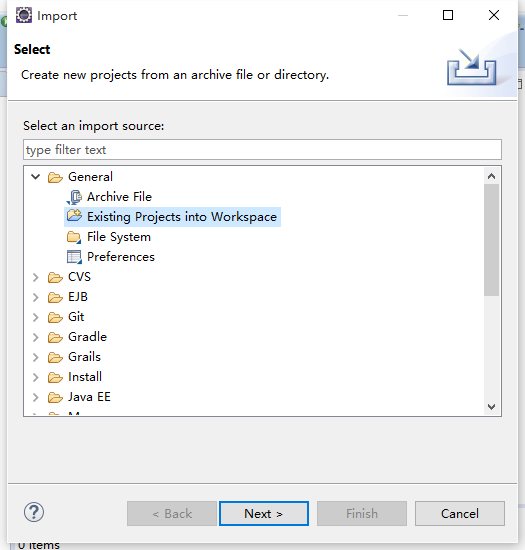
导入到开发工具中后，您可以自由的编译和修改PublicCMS源码，进行二次开发等。

### 操作步骤

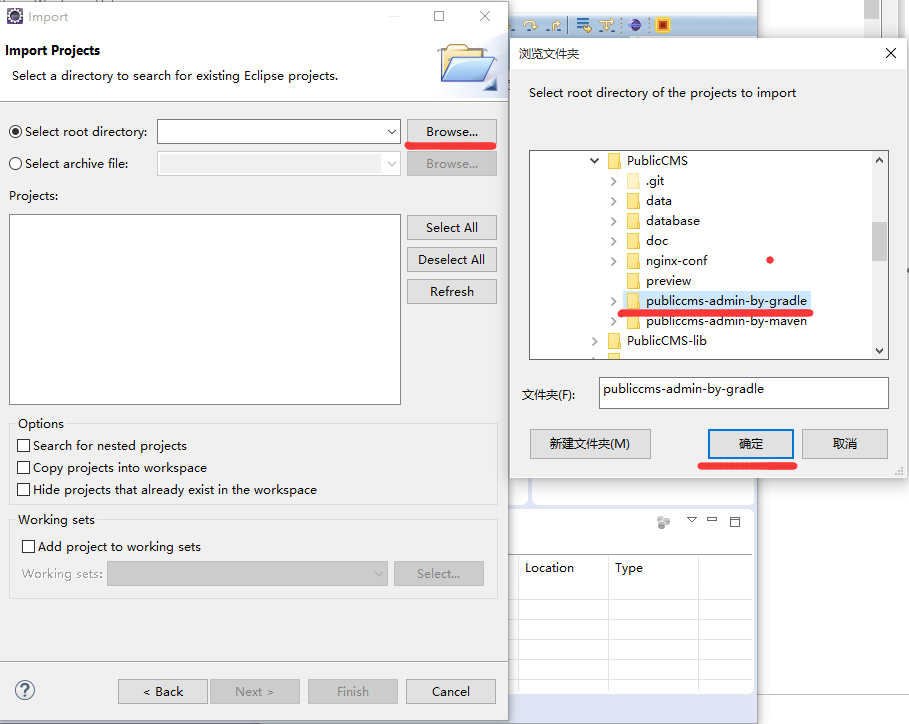
* 首先选择File菜单中的Import(导入)



* 在弹出的导入对话框中选择Existing Projects into Workspace(导入已有的工程)



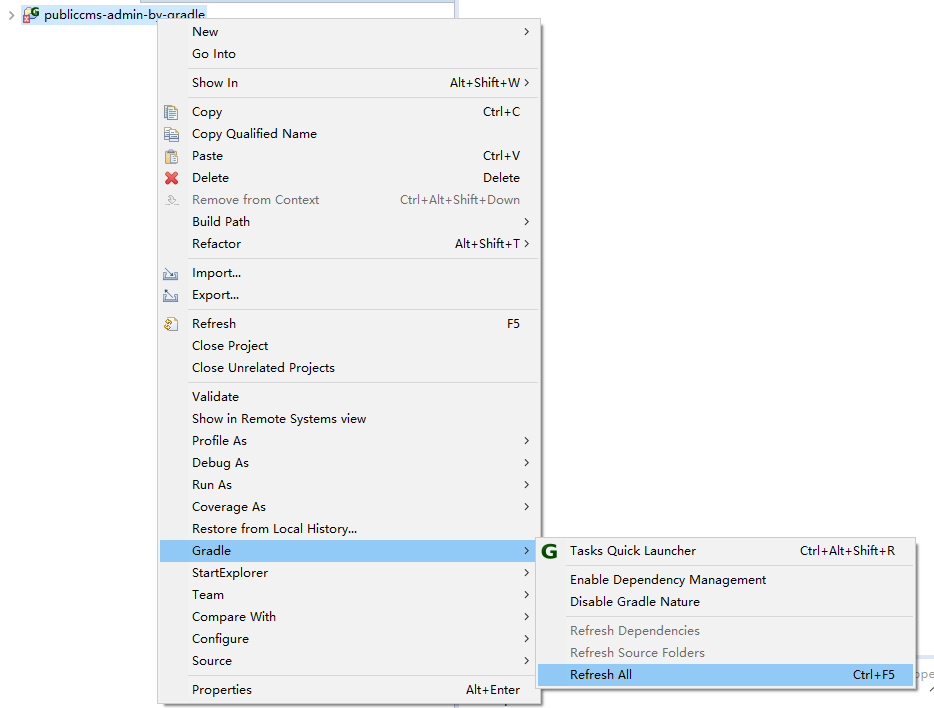
* 使用Select root directory(选择跟目录)的方式浏览选择PublicCMS目录或某个工程目录



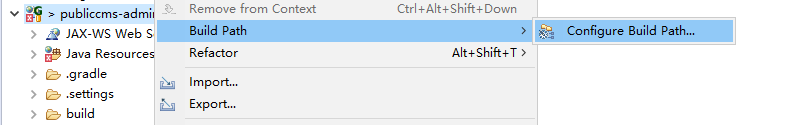
* 在根据需要选择gradle工程或者maven工程，建议勾选Copy projects into workspace(复制到工作空间)

### 注意事项

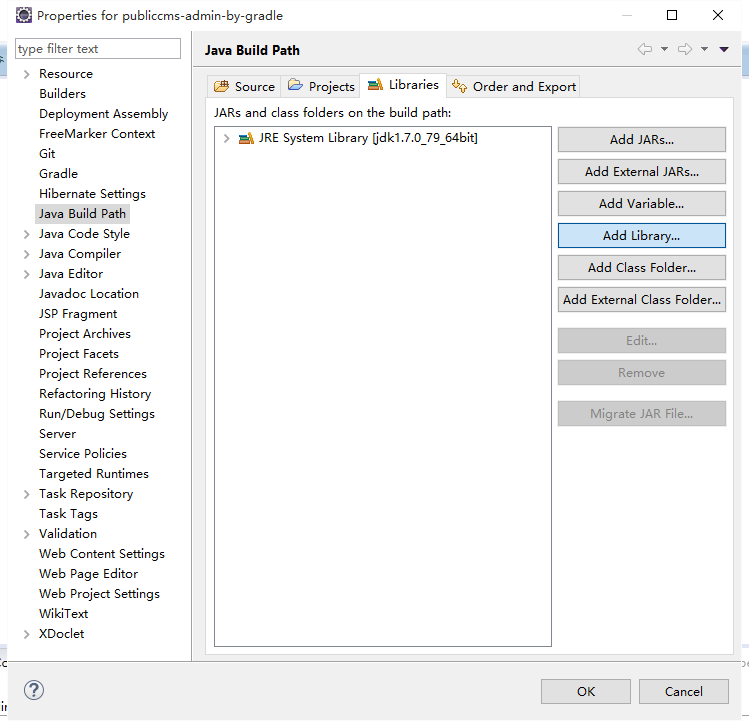
1. 如果是gradle工程，需要执行Gradle->Refresh All来解决一些导入后的报错信息



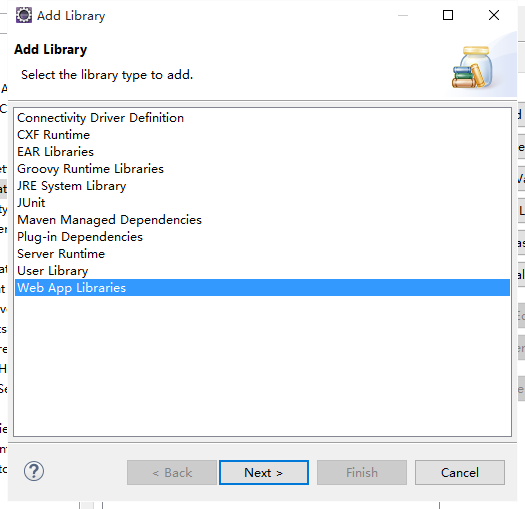
* 如果您没有使用gradle或者maven，请使用导入gradle工程到开发工具中，然后在工程WebContent/WEB-INF/目录下创建lib文件夹，并将PublicCMS-lib目录下所有jar包复制到lib文件夹下，并修改build path(编译路径)



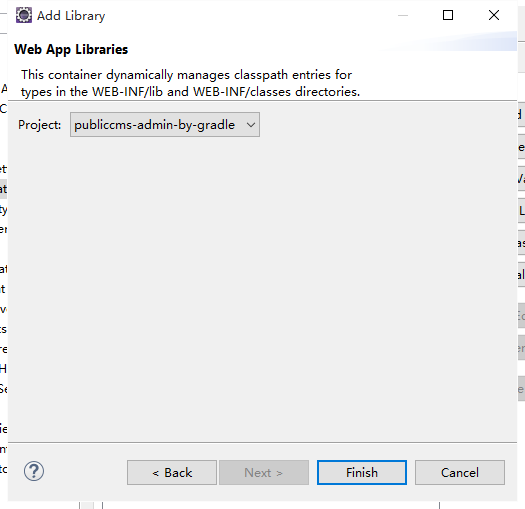
点击右侧Add Library按钮



选择Web App Libraries(Web应用类库)，然后点击next >(下一步)



选择publiccms工程，点击Finish



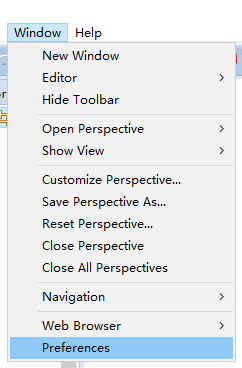
回到Build Path对话框后点击Ok保存配置

## 开发工具配置

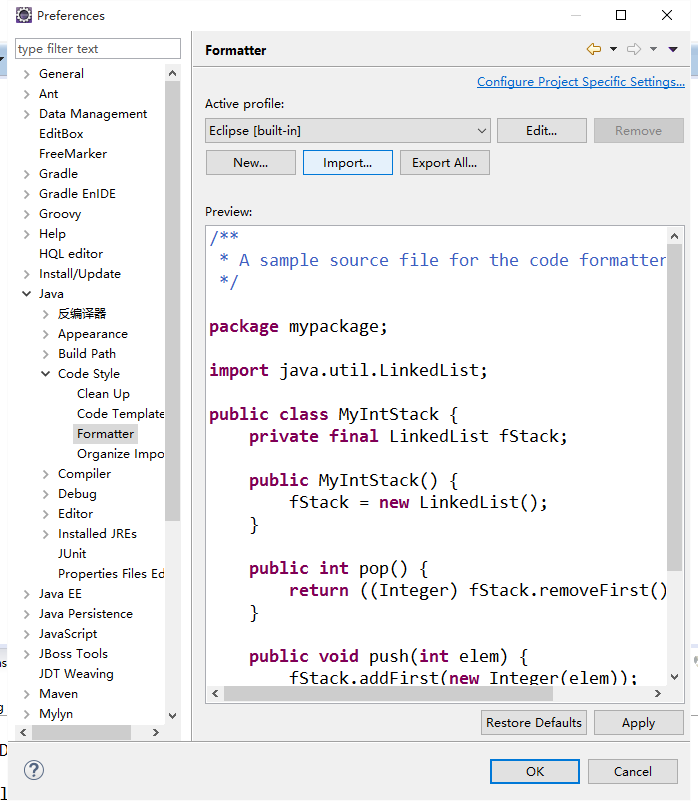
开发工具配置选项是为了您更好的对PublicCMS进行开发，并保持代码风格的一致性

### 导入格式化文件

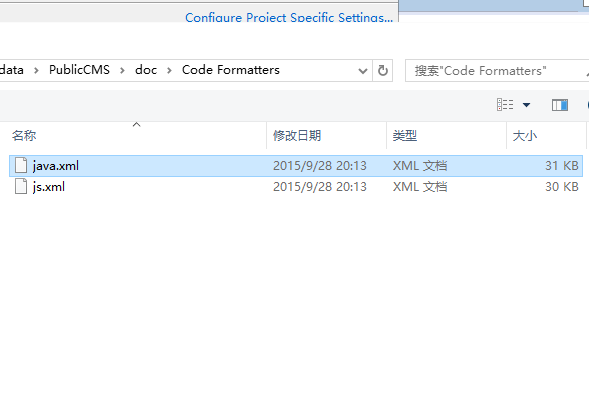
选择Window菜单下的Preferences菜单项



展开到Java->Code Style->Formatter选项卡点击Import按钮



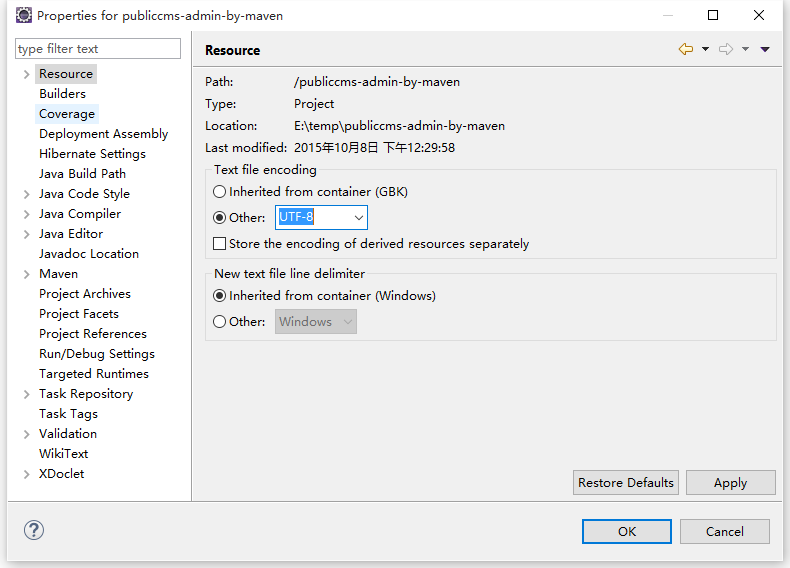
选择java.xml文件



使用相似的步骤导入js.xml到JavaScript->Code Style->Formatter中

### 字符编码设置

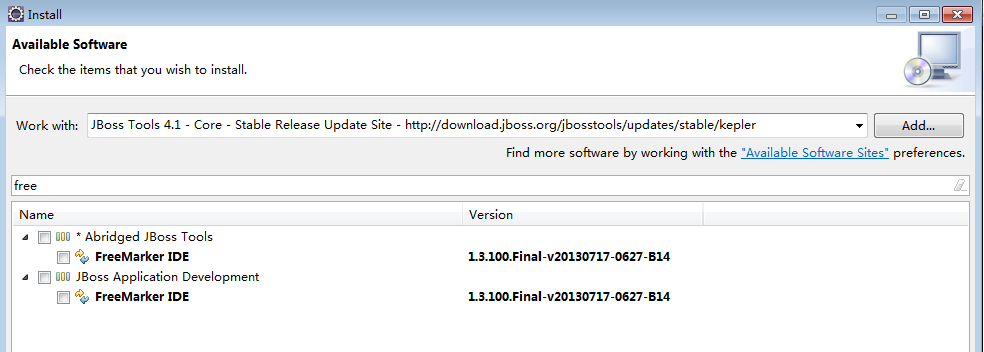
工程的编码都是UTF-8格式



### 开发工具插件

FreeMarker编辑器Eclipse插件被包含在jboss tools中

<http://download.jboss.org/jbosstools/updates/stable/> (需要选择IDE版本)



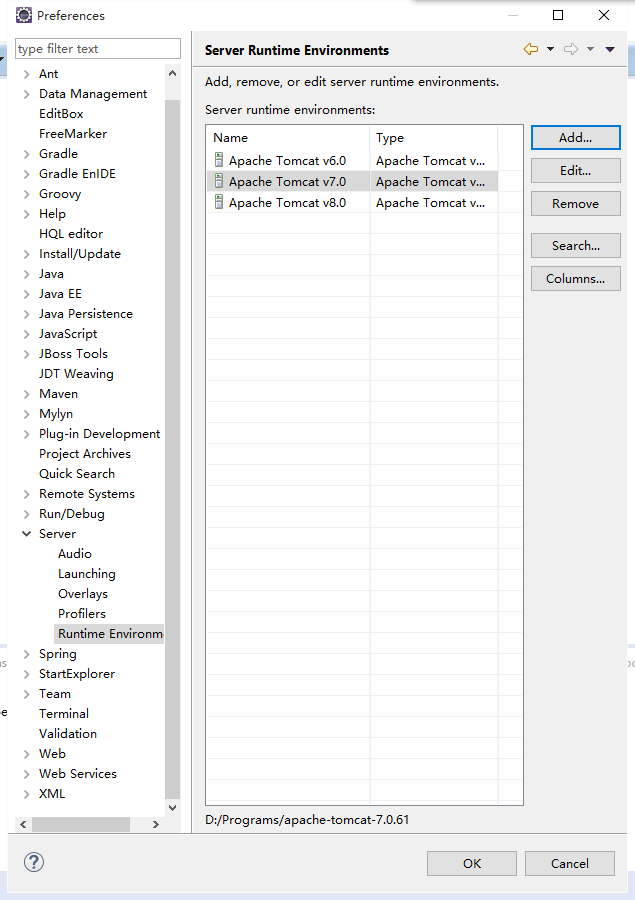
通过过滤选择FreeMarker IDE插件安装

需要注意的是该插件更新较慢，而且不是非常稳定，对于FreeMarker比较熟悉的人员来说，它并不能加快开发速度，因此只建议FreeMarker的初学者或对FreeMarker指令不熟悉的开发人员使用

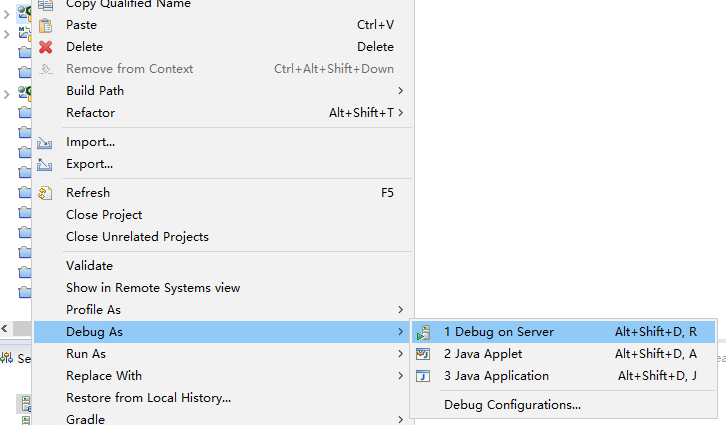
## 工程调试

在开发工具中调试时您可以选择编写测试代码进行单元测试，也可以直接在容器中运行工程，进行在线调试。模板的调试建议您选择在线进行。

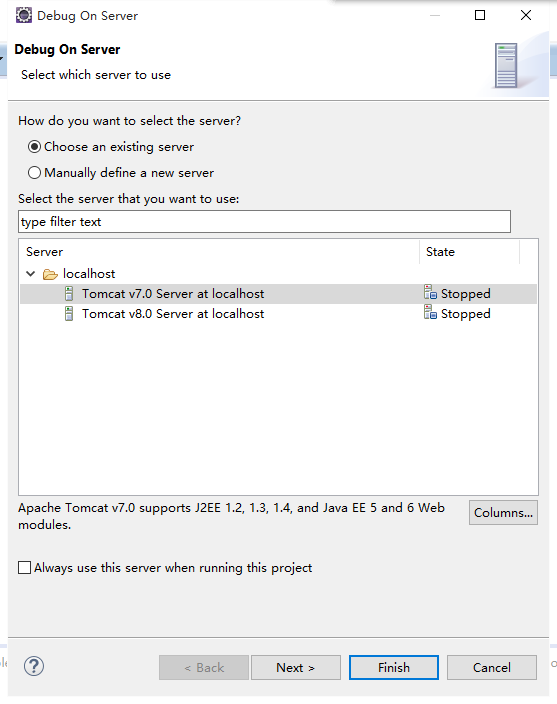
选择Window菜单下的Preferences菜单项, 展开到Server->Runtime Environments选项卡，新增一个Tomcat7.0以上版本的容器



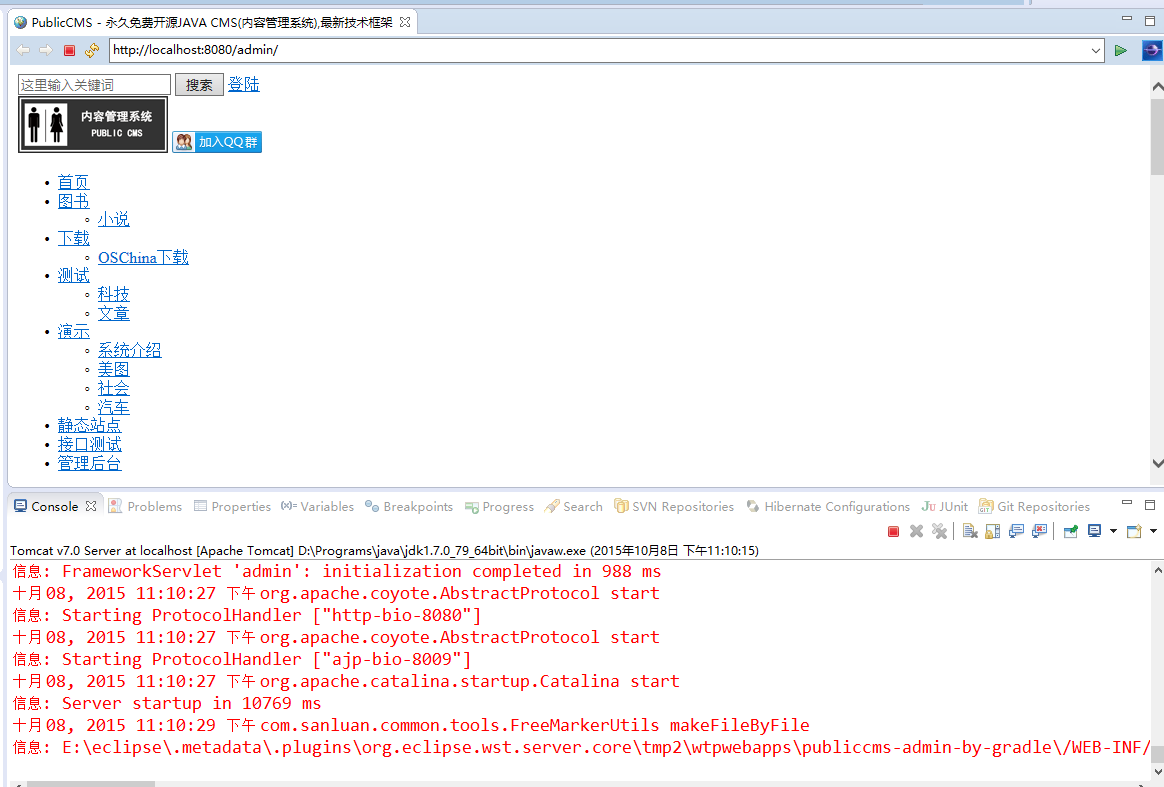
在工程上右键，选择Debug As菜单的Debug on Server子菜单项或者Run As菜单的Run on Server子菜单项



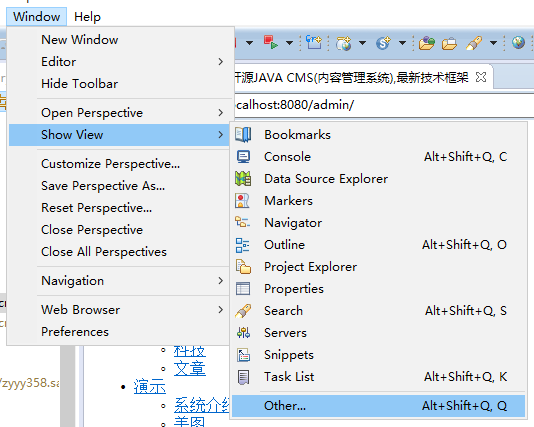
选择刚才添加的容器，然后点击Finish按钮



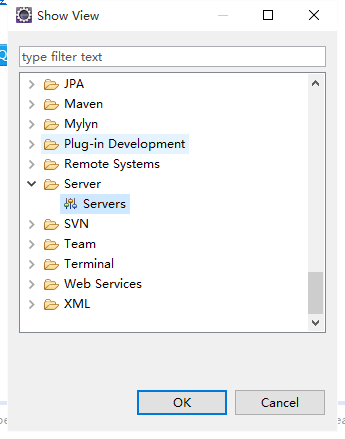
即可在工程中调试或运行工程



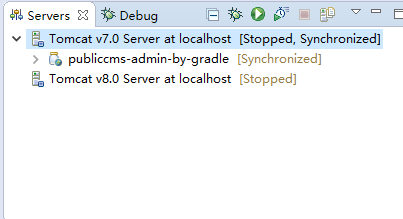
还可以在Window菜单中的Show View子菜单选择Other菜单项



打开Servers视图



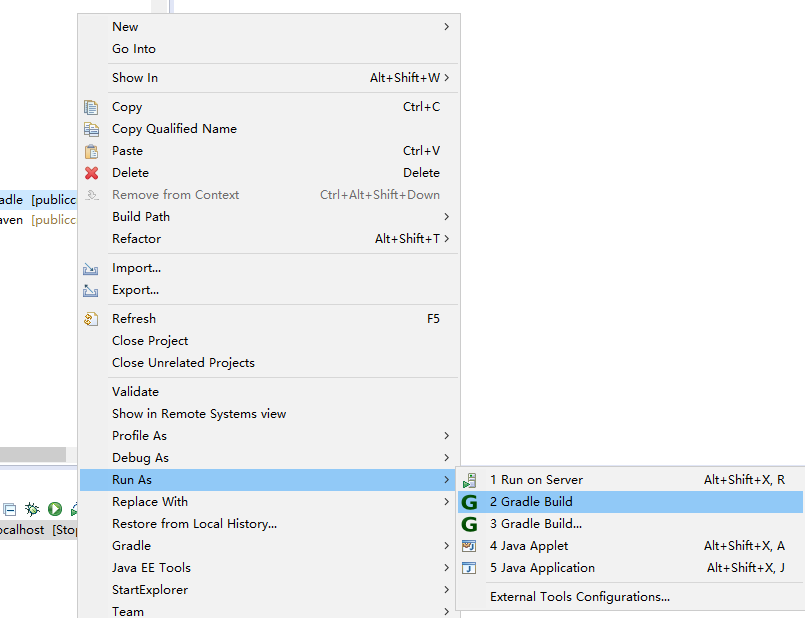
对所有容器和容器中运行或调试的工程进行管理



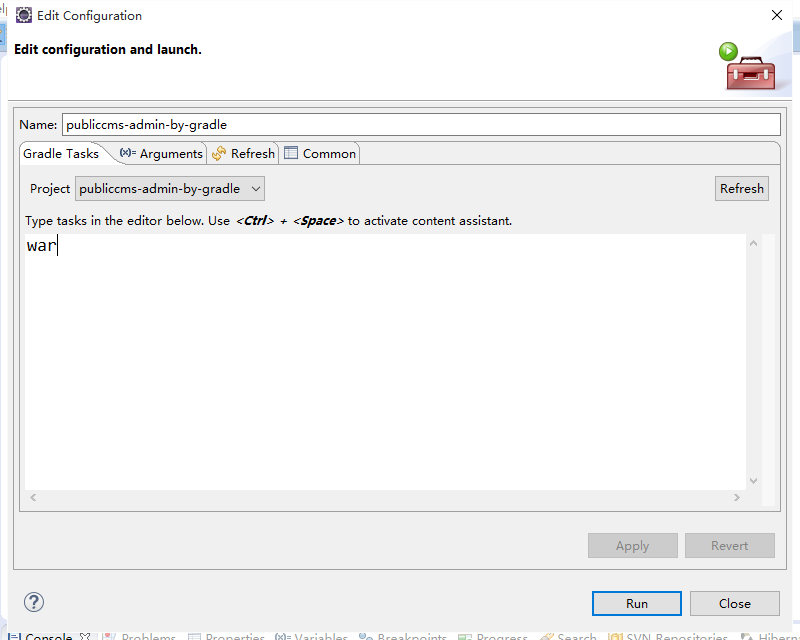
## 工程发布

### gradle工程发布

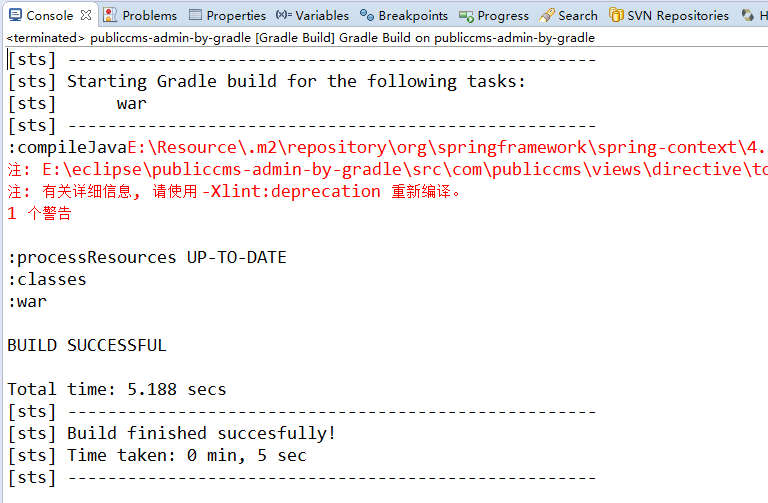
* 选择工程，右键弹出菜单中选择"Run As",在子菜单中选择"Gradle build"



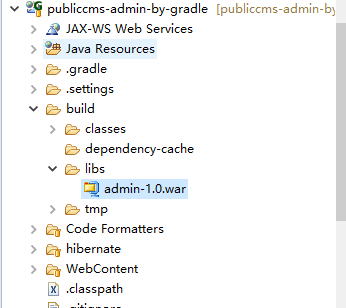
* 在弹出的对话框中输入"war",点击按钮"Run"



* 在Console中可以看到编译过程日志

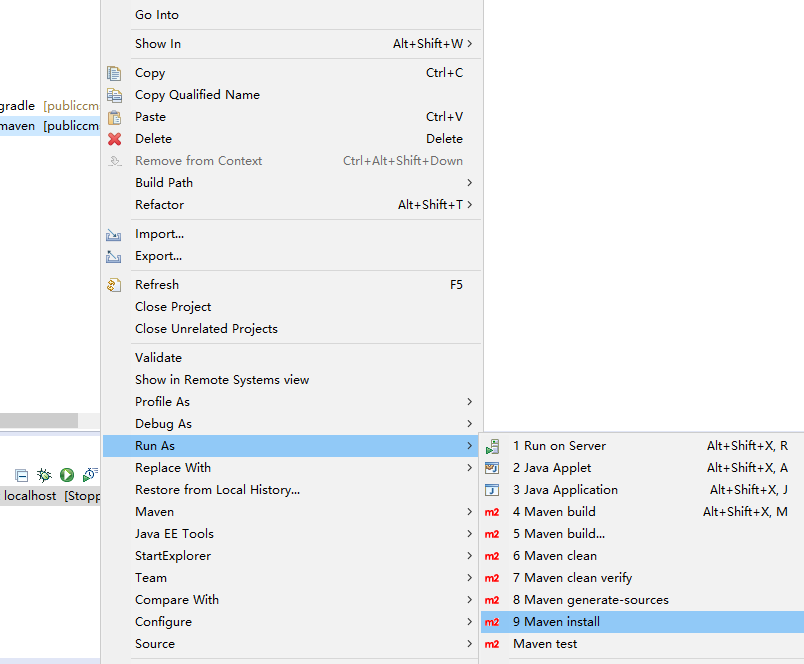


* 如果编译成功，您可以在工程下面 build/libs 目录下找到编译完成的 war包



### maven工程发布

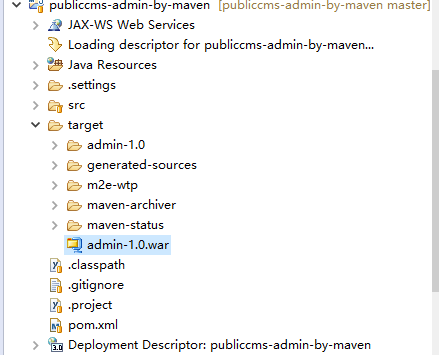
* 选择工程，右键弹出菜单中选择"Run As",在子菜单中选择"Maven install"



* 在Console中可以看到编译过程日志



* 如果编译成功，您可以在工程下面 target 目录下找到编译完成的 war包



# 二次开发

## 工程代码简介

### 架构

1. 软件基础架构

软件使用了FreeMarker 2.3.23, SpringMVC 4.2.4.RELEASE, Hibernate 5.1.0.Final, hibernate-search5.5.2.Final, Mysql jdbc5.1.35, c3p0 0.9.1.2, dom4j 1.6.1, log4j+bogback, Apache commons及其依赖的包,包依赖关系,工程编译等工作由Gradle或Maven处理

1. 软件核心机制

软件使用IndexAdminController, IndexController做统一模板分发。数据读取操作交给FreeMarker指令扩展来处理，实现了视图层JAVA代码的最大化复用，然后表单的处理采用了传统的SpringMVC Controller接收并处理的方式。

### 命名规范

.common 工具类、解决方案类等同用类包

.base 基础类包

.database 数据源包

.directive 指令基础包

.constants 常量定义类包

.handler 分页处理，查询处理，FreeMarker异常等处理器包

.index 索引组件

.servlet Servlet包

.source 源码工具包

.tools 工具类包

.view 视图层基础包

.entities 实体类包

.logic 逻辑类包

.component 组件包

.dao 数据库操作类包

.service 事务与逻辑封装类包

.views 视图类包

.controller 控制器类包

.directive freemaker自定义指令类包

.method freemaker自定义方法类包

.pojo 表单实体包

config.language 语言文件

config.properties 配置文件

config.initializer Servlet加载器

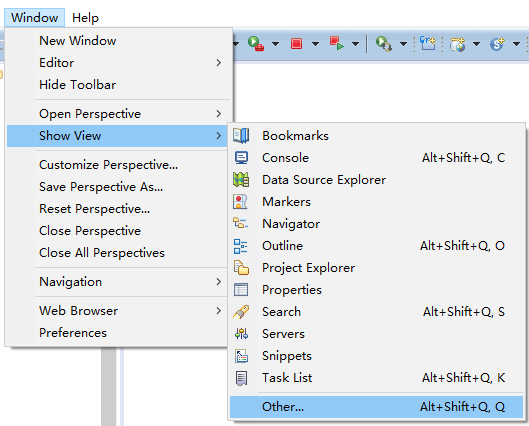
config Servlet,Spring配置项

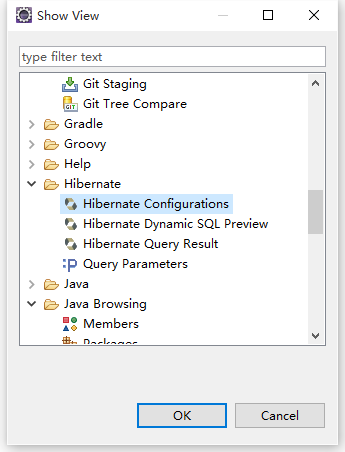
## 开发流程

1. 在开发工具中使用hibernate工具生成实体类，修改数据库配置文件

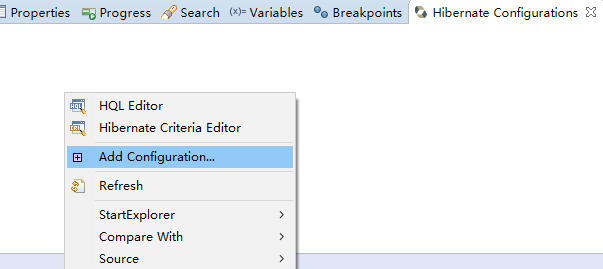
在Eclipse

将database\hibernate目录复制到工程中，也可以在后续操作中自己新建其中的配置文件，打开开发工具的Hibernate Configuration视图

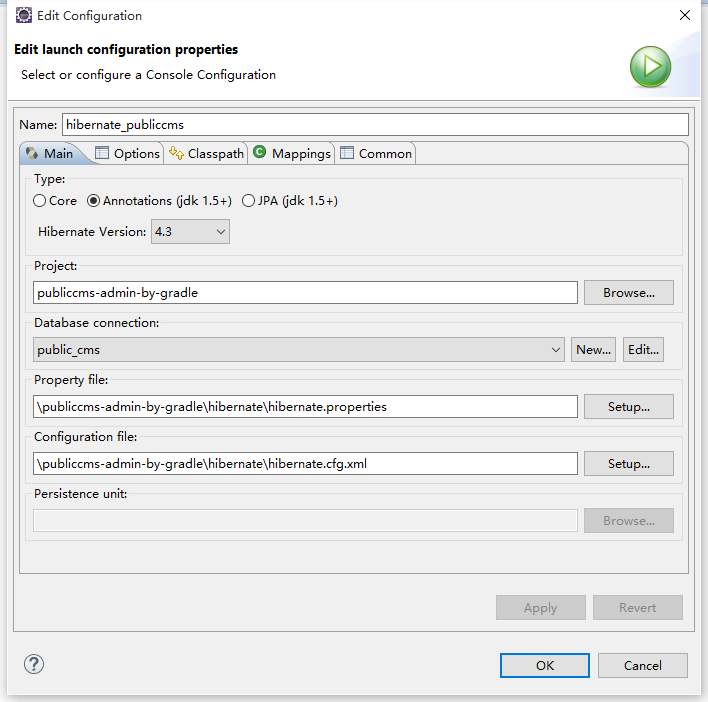




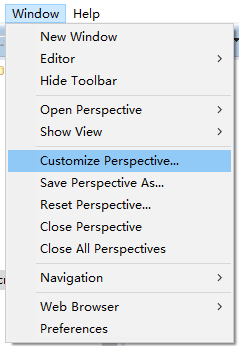
增加Hibernate配置



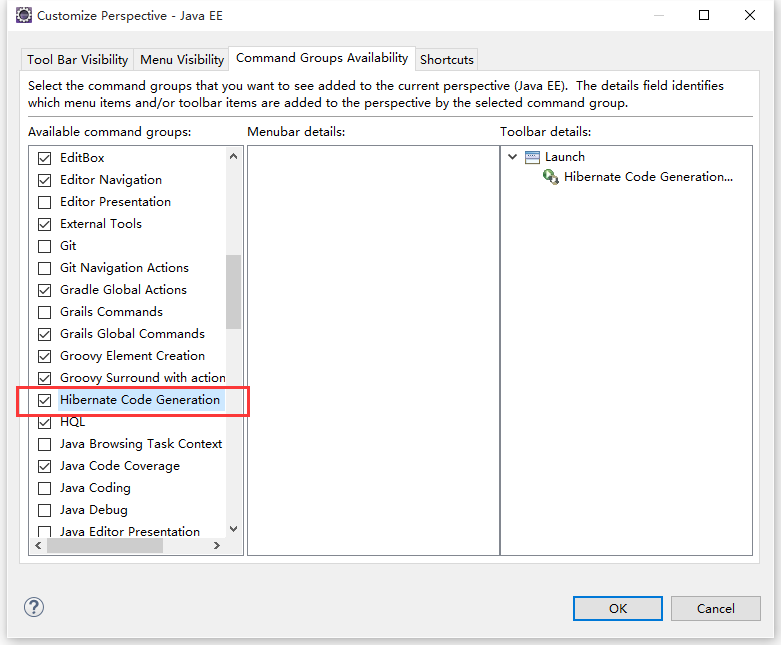
选择Annotations注解类型的实体，新建或者选择已有的数据库，选择hibernate目录下的properties和cfg.xml文件或者自己新建这两个文件，点击ok保存



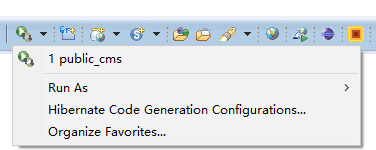
打开自定义视图



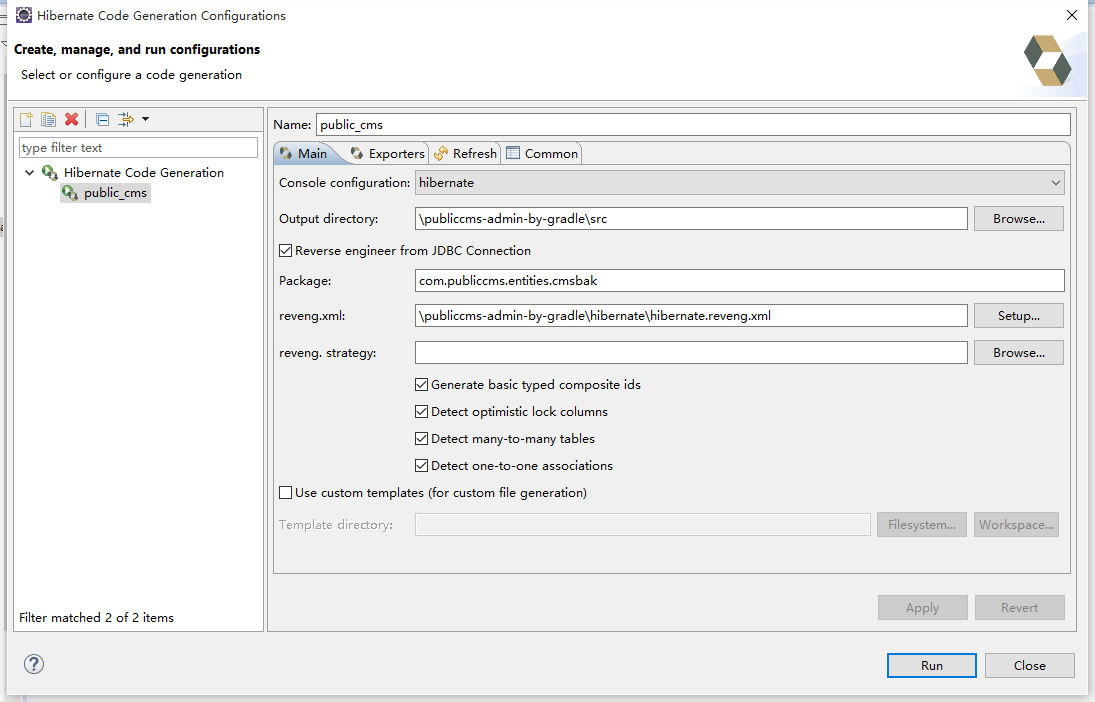
切换到Command Groups Availability选项卡，找到并勾选Hibernate Code Generation命令



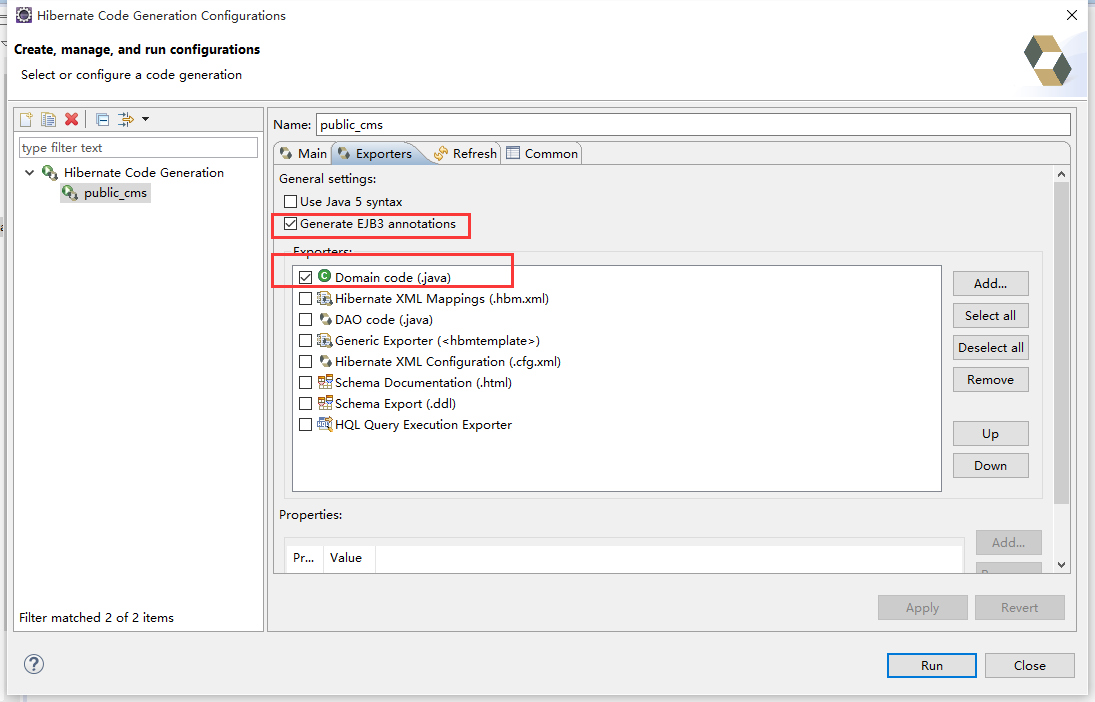
在eclipse工具条中即可找到Hibernate代码生成工具。点击Hibernate Code Generation Configurations新建代码生成配置



选择hibernate目录下的hibernate.reveng.xml.xml文件或者自己新建这个文件



切换到Exporters选项卡，勾选Generate EJB3 annotations和Domain code(.java)



然后点击run就可以生成带注解的实体类了。

1. 对实体类属性使用com.sanluan.common.source.entity.MyColumn注解类标记字段类型【condition:是否用于查询条件，order:是否用于排序，like:查询条件是否为like匹配，or:查询条件是否为or逻辑，name:字段名，当or为true时，相同name 的字段为同一组or条件，title:字段真实含义】
2. 使用com.sanluan.common.source.Maker工具类 生成自定义指令，Controller，Service，Dao，html
3. 根据实际业务调整代码中的逻辑

## 获取支持

在线调试FreeMarker <http://www.sanluan.com/freemarker_test.html>