参考：<http://www.cnblogs.com/bojuetech/p/5844413.html>

[使用IDEA和gradle搭建Spring MVC和MyBatis开发环境](http://www.cnblogs.com/bojuetech/p/5844413.html)

**1. 概述**

[Gradle](http://gradle.org/)是一个基于Apache Ant和Apache Maven概念（me：新技术的出现肯定要借鉴之前的技术）的项目自动化建构工具。 它使用一种基于Groovy的特定领域语言(DSL)（me：该语言的也可以编译成class文件然后到jvm上执行）来声明项目设置，抛弃了基于XML的各种繁琐配置（me：maven即为xml的配置）。

[Spring MVC](http://spring.io/)属于SpringFrameWork的后续产品，已经融合 在Spring Web Flow里面。Spring 框架提供了构建 Web 应用程序的全功能 MVC 模块。

[MyBatis](http://blog.mybatis.org/)本是apache的一个开源项目iBatis, 2010年这个项目 由apache software foundation 迁移到了google code，并且改名为MyBatis 。 2013年11月迁移到Github。

**2. 准备工作**

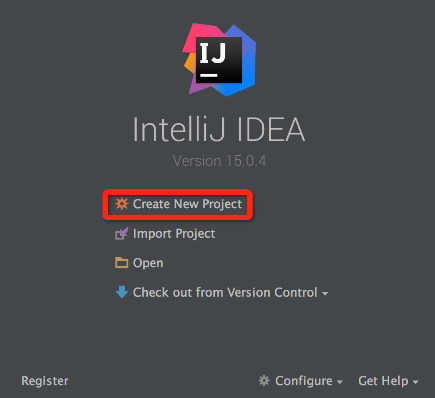
1.安装mysql，并完成相应数据库的建立

2.安装IDEA完整版

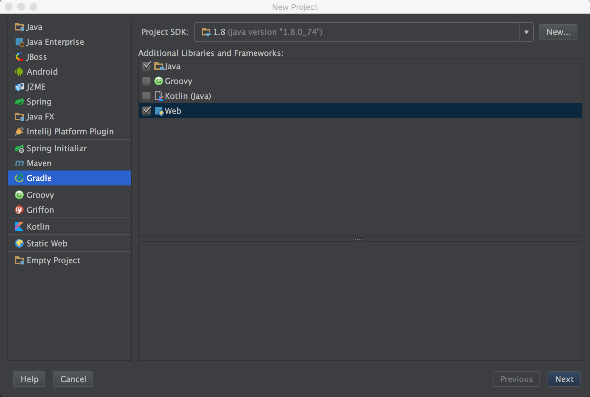
3.下载tomcat或其他服务器

**3. 创建工程**

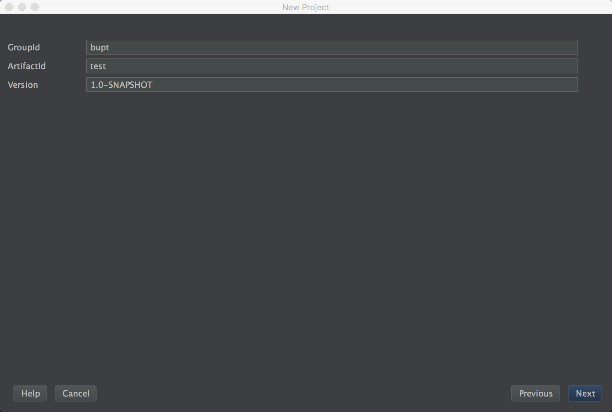
打开[IDEA](https://www.jetbrains.com/idea/)软件，选择Create New Project：



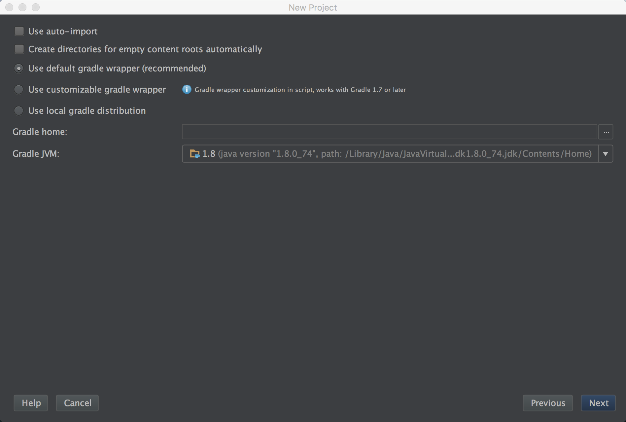
选择创建基于gradle的java web项目：



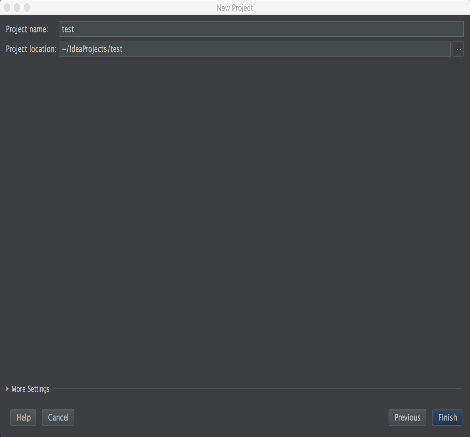
输入GroupId和ArtifactId：

s

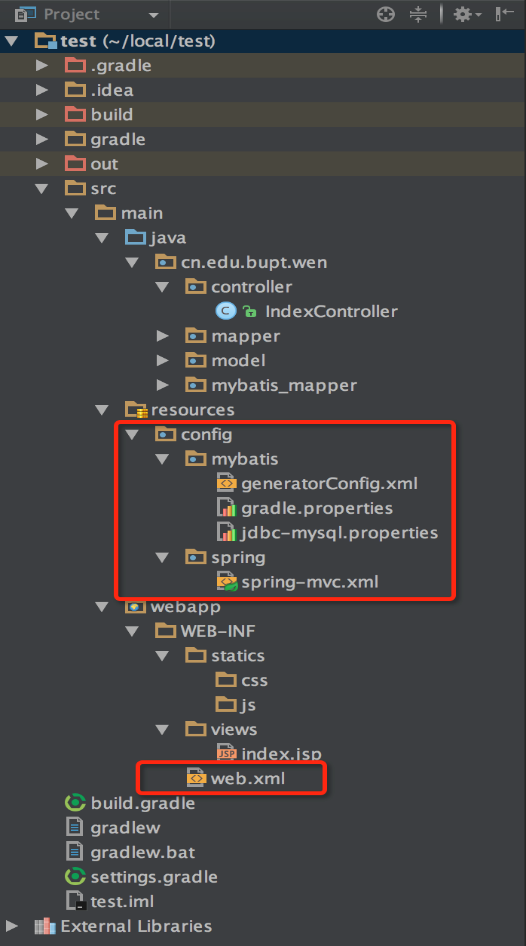
选择gradle的来源，可以用自己下载的gradle，也可以使用IDEA内建的gradle版本：



选择项目的目录：



点击完成，会生成一个最基本的java web目录，增加文件夹和相应文件，将其目录结构改为以下结构：



其中resources目录中的config文件夹中的文件和WEB-INF文件夹下的web.xml， 以及build.gradle都是自己进行配置的。

**配置文件**

**1. generatorConfig.xml**

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

<!DOCTYPE generatorConfiguration

PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"

"http://mybatis.org/dtd/mybatis-generator-config\_1\_0.dtd">

<generatorConfiguration>

<classPathEntry location="${jarDirection}"/>

<context id="mysql" targetRuntime="MyBatis3">

<jdbcConnection driverClass="${driverClass}"

connectionURL="${connectionURL}"

userId="${userId}"

password="${password}">

</jdbcConnection>

<javaTypeResolver >

<property name="forceBigDecimals" value="false" />

</javaTypeResolver>

<javaModelGenerator

targetPackage="${modelPackage}"

targetProject="${src\_main\_java}">

<property name="enableSubPackages" value="true" />

<property name="trimStrings" value="true" />

</javaModelGenerator>

<sqlMapGenerator

targetPackage="${sqlMapperPackage}"

targetProject="${src\_main\_java}">

<property name="enableSubPackages" value="true" />

</sqlMapGenerator>

<javaClientGenerator type="XMLMAPPER"

targetPackage="${mapperPackage}"

targetProject="${src\_main\_java}">

<property name="enableSubPackages" value="true" />

</javaClientGenerator>

<!-- 每一个数据库表都需要建立一个对应的table字段 -->

<table schema="test" tableName="user" domainObjectName="User" >

<!-- <property name="useActualColumnNames" value="true"/>

<generatedKey column="ID" sqlStatement="DB2" identity="true" />

<columnOverride column="DATE\_FIELD" property="startDate" />

<ignoreColumn column="FRED" />

<columnOverride column="LONG\_VARCHAR\_FIELD" jdbcType="VARCHAR" /> -->

</table>

</context>

</generatorConfiguration>

**2. gradle.properties**

# 生成的model类所在包

modelPackage=cn.edu.bupt.wen.model

# 生成的mapper接口类所在包

mapperPackage=cn.edu.bupt.wen.mapper

# 生成的mapper xml文件所在包

sqlMapperPackage=cn.edu.bupt.wen.mybatis\_mapper

**3. jdbc-mysql.properties**

# 将jdbc.jarDirection后的内容更改为gradle下载的

# mysql-connector-java-5.1.x.jar所在的路径，gradle自动下载的路径如下所示

jdbc.jarDirection=/Users/wenzhiquan/.gradle/caches/modules-2/files-2.1/mysql

/mysql-connector-java/5.1.38/dbbd7cd309ce167ec8367de4e41c63c2c8593cc5

/mysql-connector-java-5.1.38.jar

jdbc.driverClassName=com.mysql.jdbc.Driver

jdbc.url=jdbc:mysql://localhost:3306/bookbest

jdbc.user=root

jdbc.pass=root

**4. spring-mvc.xml**

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/mvc

http://www.springframework.org/schema/mvc/spring-mvc.xsd">

<!-- 自动扫描控制器 -->

<context:component-scan base-package="cn.edu.bupt.wen"/>

<!-- 视图渲染 -->

<bean id="internalResourceViewResolver"

class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="prefix" value="/WEB-INF/views/"/>

<property name="suffix" value=".jsp"/>

</bean>

<!-- 控制器映射器和控制器适配器 -->

<mvc:annotation-driven></mvc:annotation-driven>

<!-- 静态资源映射器 -->

<mvc:resources mapping="/statics/\*\*" location="/WEB-INF/statics/" />

</beans>

**5. web.xml**

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

<web-app xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"

version="3.0">

<servlet>

<servlet-name>index-dispather</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath\*:config/spring/spring-mvc.xml</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>index-dispather</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

**6. build.gradle**

group 'cn.edu.bupt.wen'

version '1.0-SNAPSHOT'

apply plugin: 'groovy'

apply plugin: 'java'

apply plugin: 'war'

apply plugin: 'idea'

sourceCompatibility = 1.5

repositories {

mavenCentral()

}

configurations {

mybatisGenerator

}

// 读取config文件夹中对应的配置文件

ext{

def prop = new Properties();

file("src/main/resources/config/mybatis/jdbc-mysql.properties")

.withInputStream {

prop.load(it)

}

file("src/main/resources/config/mybatis/gradle.properties")

.withInputStream {

prop.load(it)

}

prop.each {

project.extensions.add("$it.key",it.value);

}

}

// 配置gradle的依赖项

dependencies {

compile 'org.codehaus.groovy:groovy-all:2.3.11'

compile 'jstl:jstl:1.2'

compile 'org.springframework:spring-aop:4.2.4.RELEASE'

compile 'org.springframework:spring-context:4.2.4.RELEASE'

compile 'org.springframework:spring-beans:4.2.4.RELEASE'

compile 'org.springframework:spring-web:4.2.4.RELEASE'

compile 'org.springframework:spring-webmvc:4.2.4.RELEASE'

compile 'org.springframework:spring-tx:4.2.4.RELEASE'

compile 'org.springframework:spring-jdbc:4.2.4.RELEASE'

compile 'org.springframework:spring-test:4.2.4.RELEASE'

compile 'mysql:mysql-connector-java:5.1.38'

compile 'org.mybatis.generator:mybatis-generator-core:1.3.2'

compile 'org.mybatis:mybatis-spring:1.2.3'

compile 'org.mybatis:mybatis:3.3.0'

compile 'junit:junit:4.12'

mybatisGenerator 'org.mybatis.generator:mybatis-generator-core:1.3.2'

mybatisGenerator 'mysql:mysql-connector-java:5.1.38'

testCompile group: 'junit', name: 'junit', version: '4.11'

}

// 利用ant创建一个自动生成MyBatis目录文件的task

task mybatisGenerate << {

ant.properties['targetProject'] = projectDir.path

ant.properties['jarDirection'] = project['jdbc.jarDirection']

ant.properties['driverClass'] = project['jdbc.driverClassName']

ant.properties['connectionURL'] = project['jdbc.url']

ant.properties['userId'] = project['jdbc.user']

ant.properties['password'] = project['jdbc.pass']

ant.properties['src\_main\_java'] = sourceSets.main.java.srcDirs[0].path

ant.properties['src\_main\_resources'] = sourceSets.main.resources.srcDirs[0].path

ant.properties['modelPackage'] = project['modelPackage']

ant.properties['mapperPackage'] = project['mapperPackage']

ant.properties['sqlMapperPackage'] = project['sqlMapperPackage']

ant.taskdef(

name: 'mbgenerator',

classname: 'org.mybatis.generator.ant.GeneratorAntTask',

classpath: configurations.mybatisGenerator.asPath

)

ant.mbgenerator(overwrite: true,

configfile: 'src/main/resources/config/mybatis/generatorConfig.xml',

verbose: true) {

propertyset {

propertyref(name: 'targetProject')

propertyref(name: 'jarDirection')

propertyref(name: 'userId')

propertyref(name: 'driverClass')

propertyref(name: 'connectionURL')

propertyref(name: 'password')

propertyref(name: 'src\_main\_java')

propertyref(name: 'src\_main\_resources')

propertyref(name: 'modelPackage')

propertyref(name: 'mapperPackage')

propertyref(name: 'sqlMapperPackage')

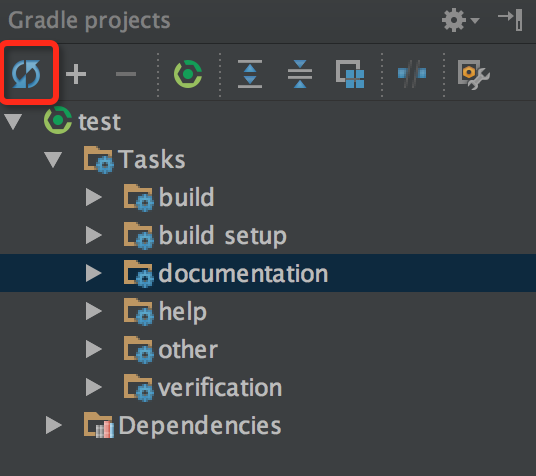
}

}

}

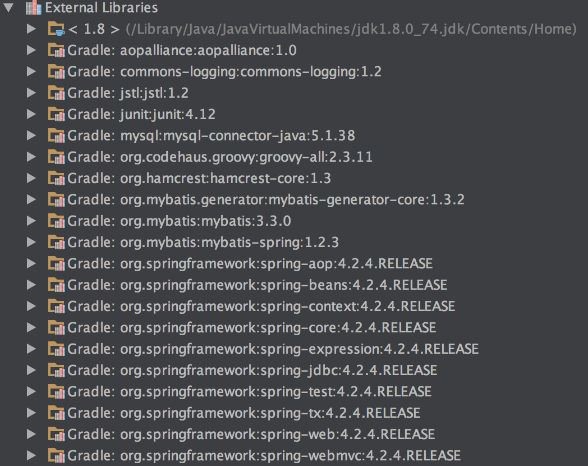
**下载gradle依赖包**

至此，基本的配置就完成了，接下来就需要运行gradle下载相应的依赖包了， 操作如下：打开IDEA > View > Tool Windows > gradle:刷新整个项目



点击图中的按钮，IDEA就会自动下载依赖包了。

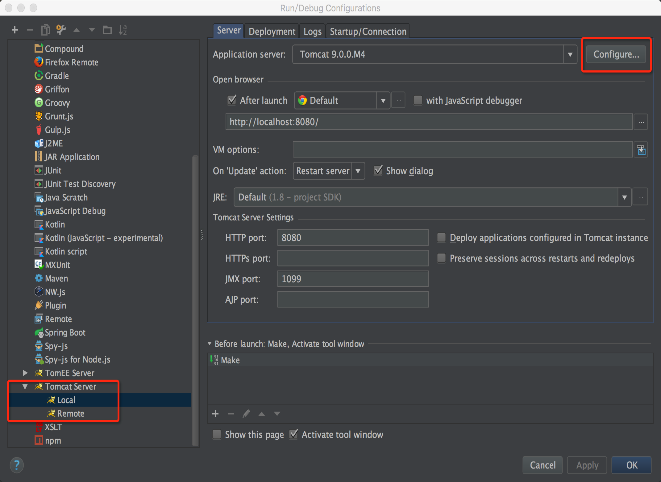
下载完成后，目录下的External Libraries文件夹会生成相应的目录：



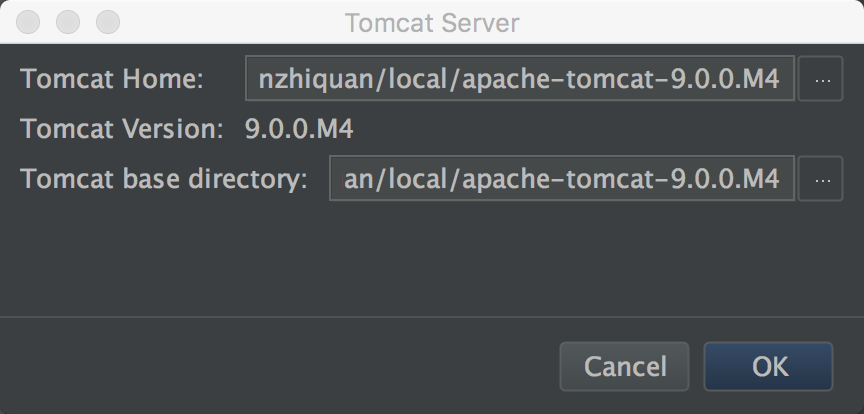
下载完依赖包之后，整体的配置基本上就完成了。

**3. 部署到Tomcat服务器**

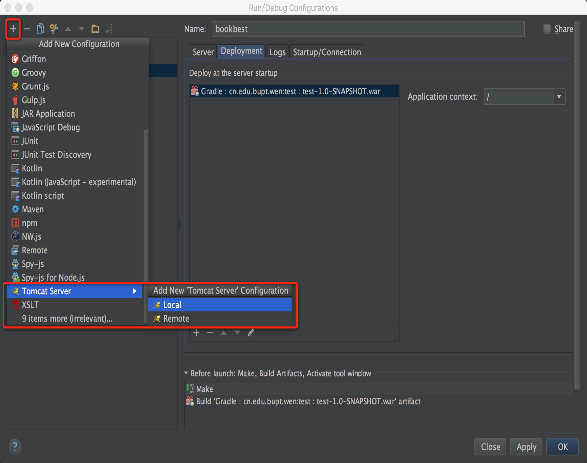
打开IDEA > Run > Edit Configurations...



点击tomcat server的local选项，点击右上角的configure...， 在弹出的对话框中选中下载的Tomcat根目录：



然后将配置好的tomcat添加到运行库当中：



**4. 编写代码并运行**

**编写controller代码**

在controller文件夹中创建IndexController文件，将代码修改为：

package cn.edu.bupt.wen.controller;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

/\*\*

\* Author wenzhiquan

\* Created on 16/4/12 下午12:25

\* File name: IndexController.java

\* Description: null

\*/

@Controller

public class IndexController {

@RequestMapping("/")

public String index(){

return "index";

}

}

**修改index.jsp**

将index.jsp文件修改为自己想要的样式：

<%--

Created by IntelliJ IDEA.

User: wenzhiquan

Date: 16/4/12

Time: 下午12:23

To change this template use File | Settings | File Templates.

--%>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<html>

<head>

<title>Index</title>

</head>

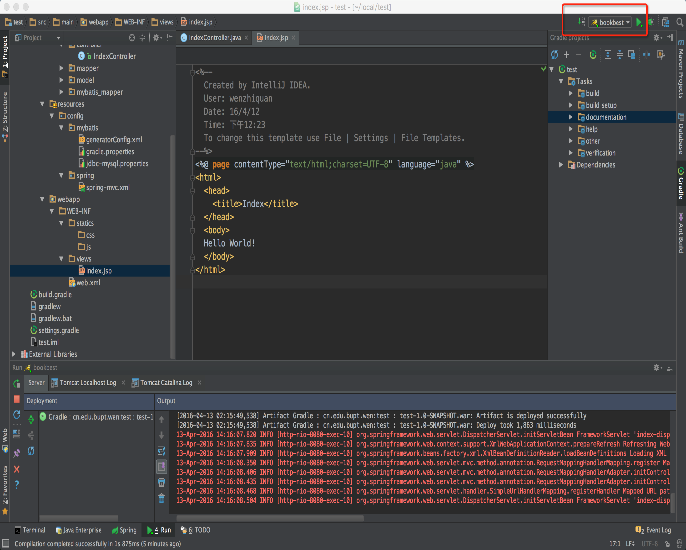
<body>

Hello World!

</body>

</html>

**运行Tomcat**



点击运行按钮，服务器启动后会自动启动浏览器进入主界面：

至此，hello world程序已经完成。

**5. 自动生成mybatis相关文件**

进入gradle的view，点击创建gradle任务：

创建一个mybatisGenerate任务：

点击OK，会在model，mapper和mybatis\_mapper文件夹中自动生成相应的文件：

至此，Spring MVC + MyBatis的Java Web开发环境配置完成。