

IntelliJ IDEA 14.1 上 JFinal 开发环境搭建手册

目 录

一、 新建项目.....	2
二、 项目参数配置.....	5
1. 新建 Module.....	5
2. 建 Jetty 运行模块.....	7
(1) 新建模块.....	7
(2) 导入类库.....	9
3. 建 Tomcat 运行模块.....	18
(1) 新建模块.....	18
(2) 导入类库.....	21
三、 修改运行配置.....	27
1. 创建 Jetty 运行配置.....	27
2. 创建 tomcat 运行配置.....	30
四、 添加源文件.....	33
五、 运行项目.....	36
1. 在 jetty 下运行.....	36
2. 在 tomcat 下运行.....	38
六、 项目实际部署.....	39

JFinal 官方的教程都是使用 Eclipse 进行开发的，而使用 IntelliJ IDEA 来做开发，不少地方和 Eclipse 区别还是很大的。

本文参考了网上不少文章，主要沿用了网友的内容，一步一步做了尝试后记录下来，感谢万能的互联网。

分别做了两个 Module，分别使用 Jetty 和 tomcat 来运行。均测试均可正常跑起来。

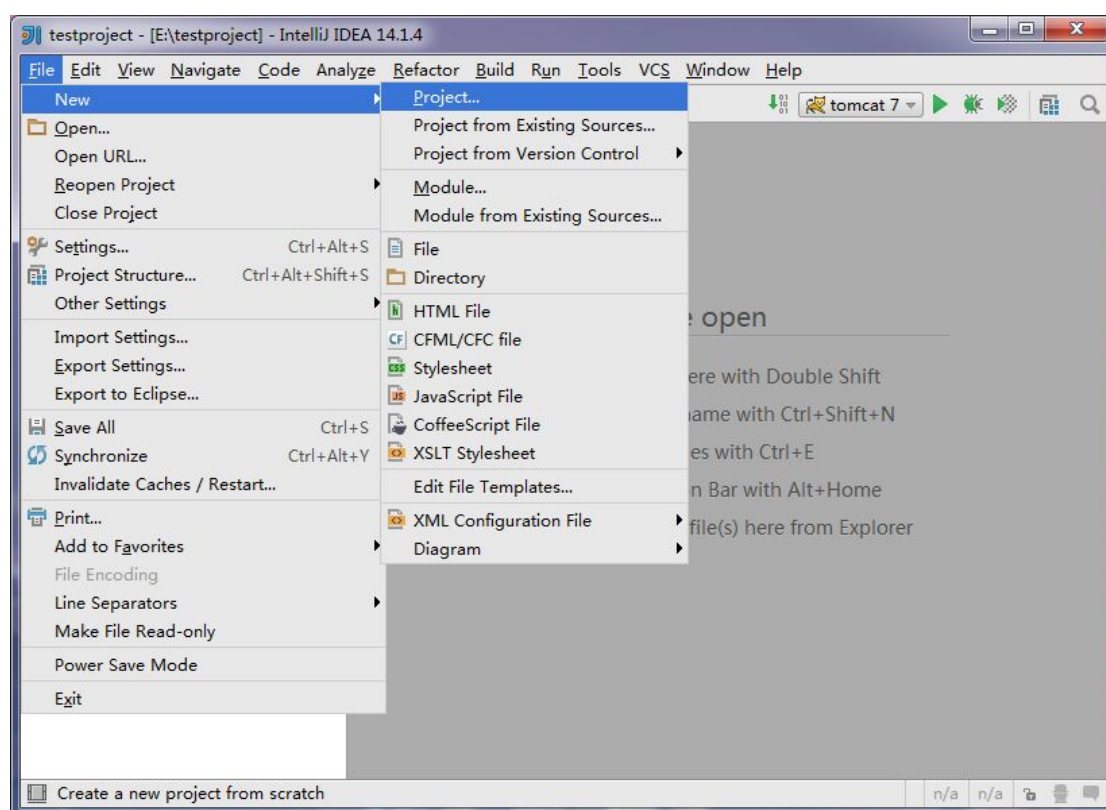
本文看起来虽然很长，但实际上步骤不多，为便于初学者了解 IntelliJ IDEA，所以细节写的非常完善，每一步都有截图。

本文使用的开发环境是 IntelliJ IDEA 14.1.4 版本。JFinal 是 2.0 版。

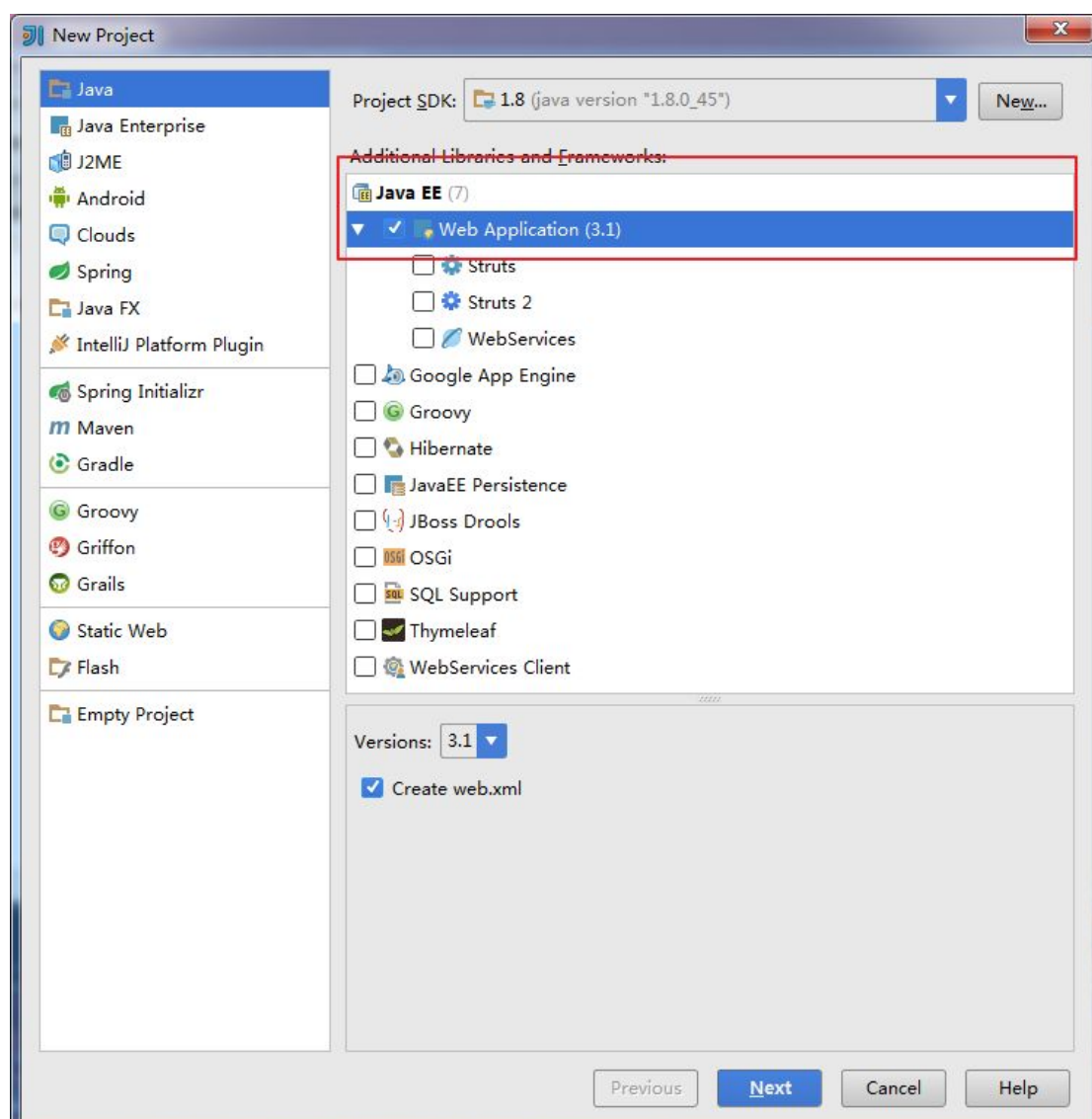
另：本文博客发布在 <http://my.oschina.net/chixn/blog/471755>。

一、新建项目

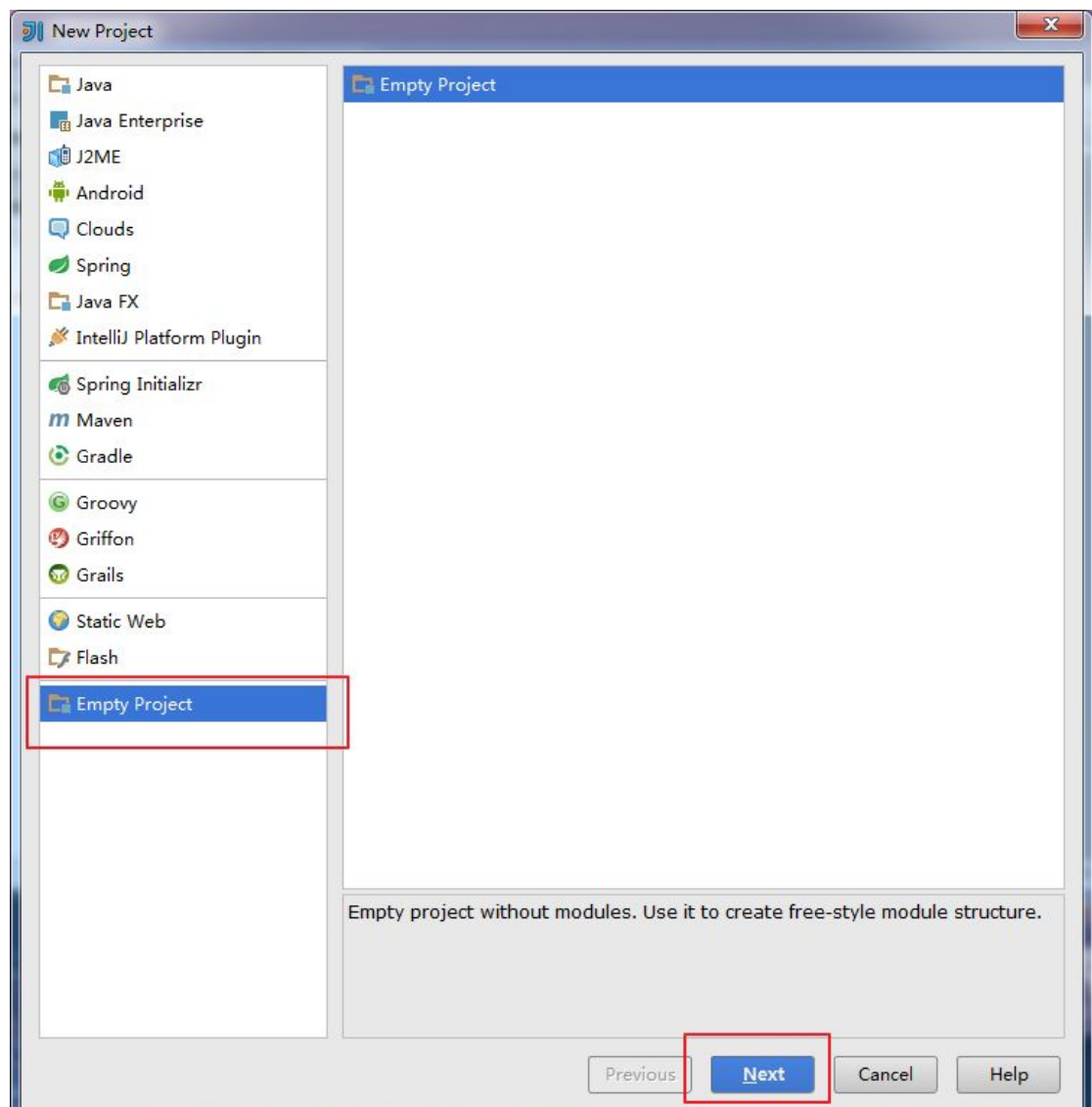
新建一个项目，可以是空项目，也可以是连模块一起建的项目，本文为了演示 Jetty 和 tomcat 均能运行的效果，所以先建一个空项目，再分别建两个不同的 module，以便区分。



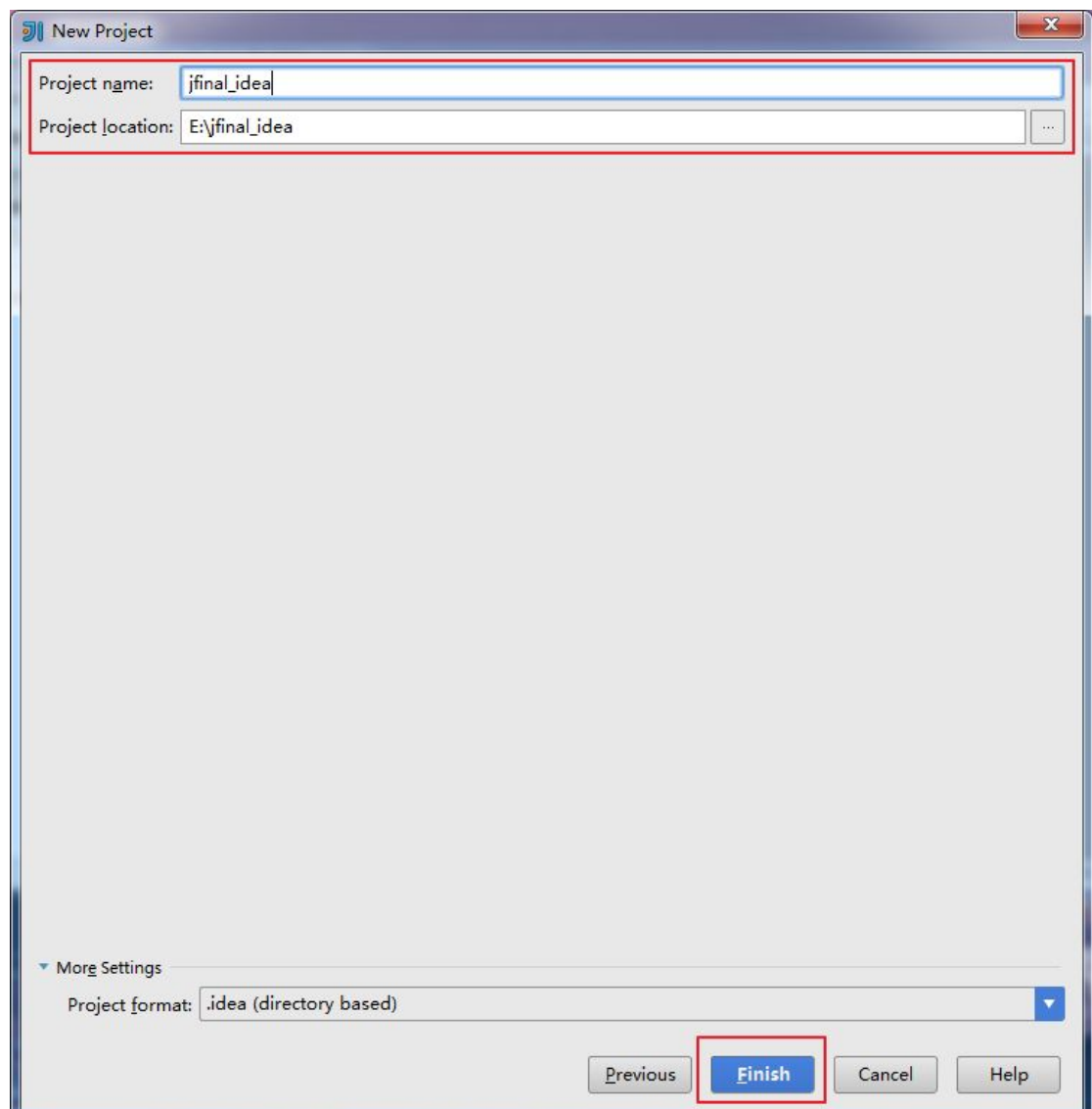
如果项目和模块一起建，可以把 **Web Application** 选上，其他的默认就行。



因为我们要分别测试 **jetty** 和 **tomcat** 的效果，要建两个模块，所以先建一个空项目。



点击“Next”进入下一步。

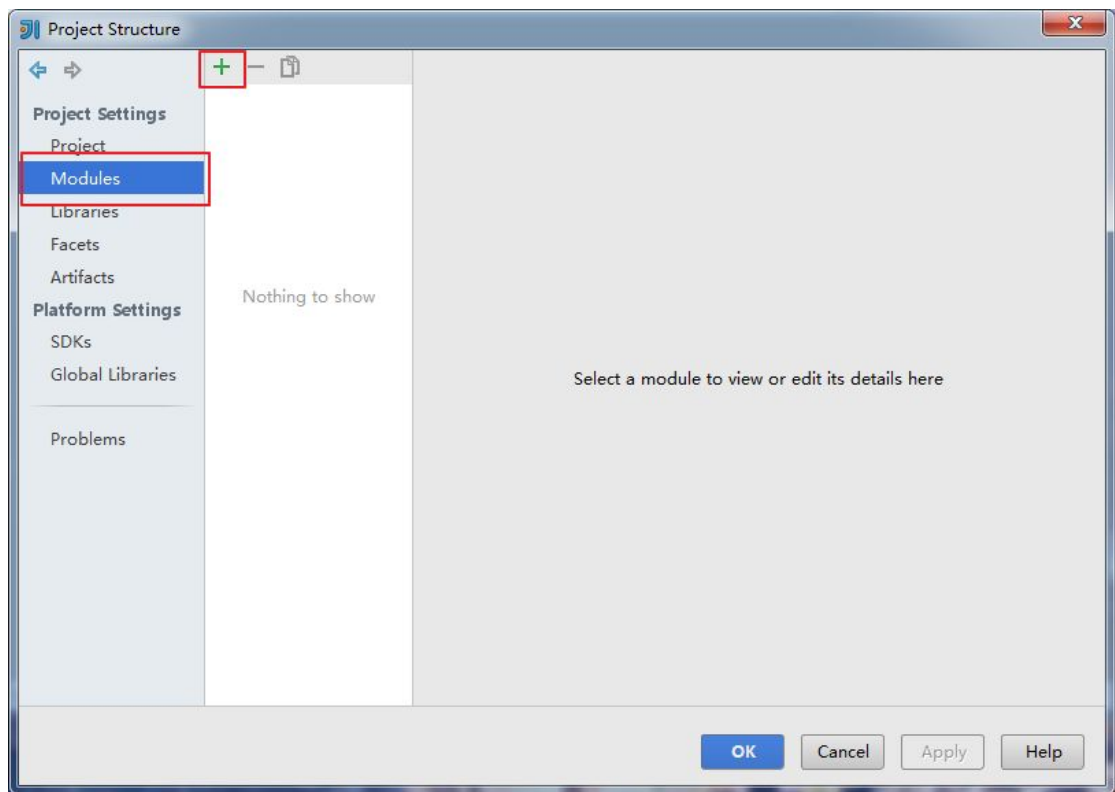


输入项目名称和项目所在目录，点击“Finish”即可。

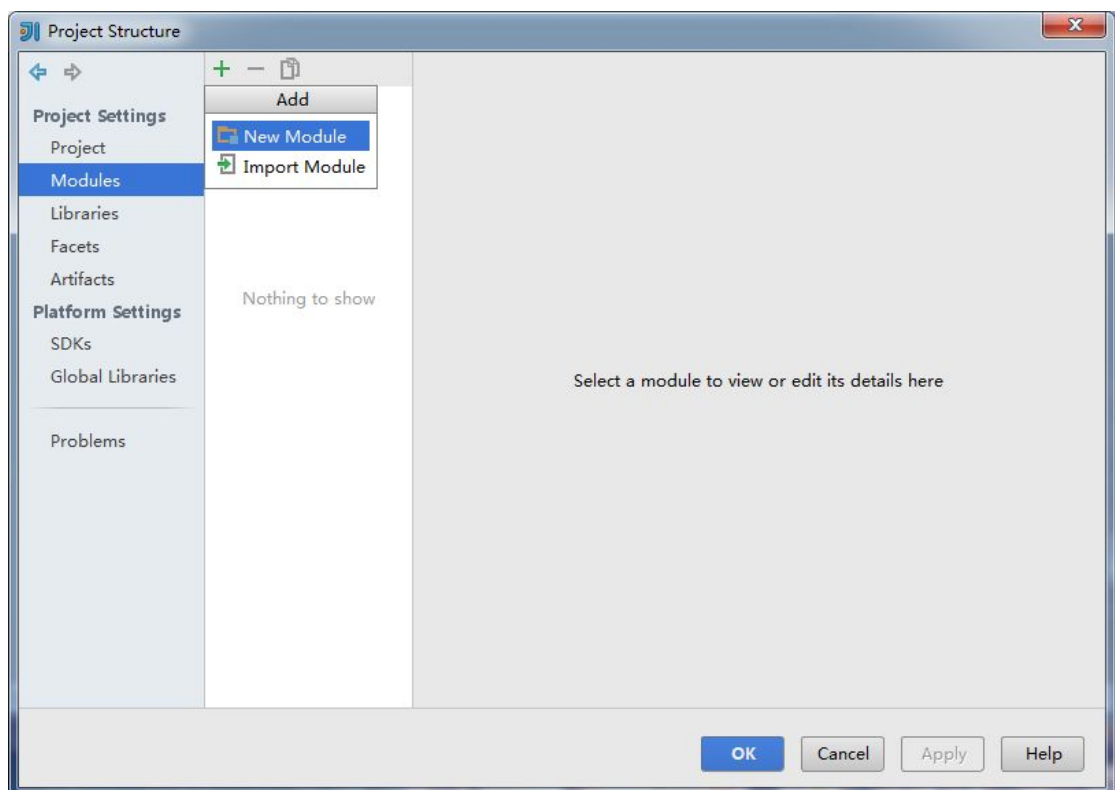
二、项目参数配置

1. 新建 Module

如果新建一个空项目，会立即出来一个 Project Structure 的配置窗口。如果是连模块一起建的，请从【File】-【Project Structure】中选择，对项目参数进行配置。



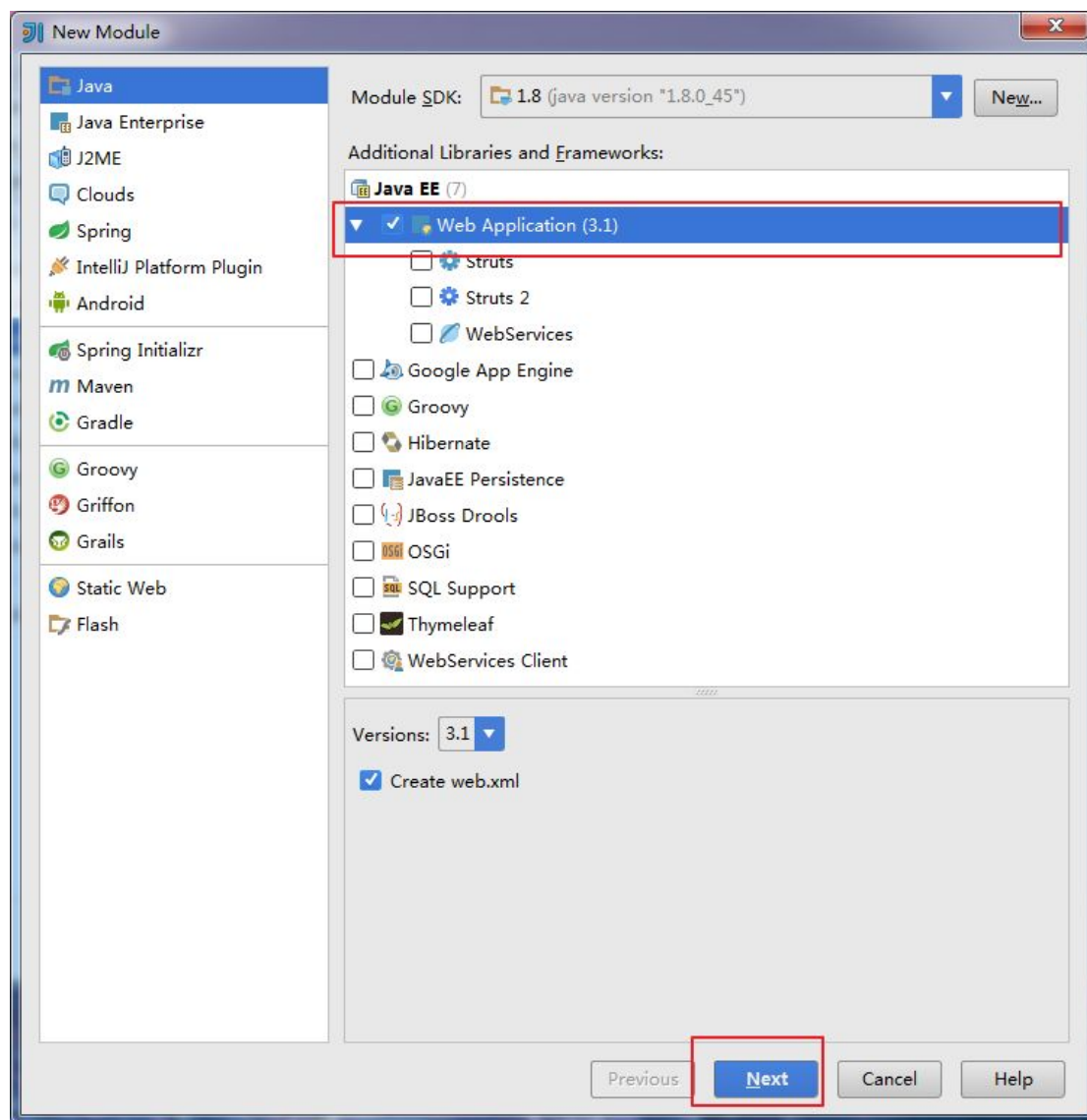
选择 Modules，准备新建 Module。



2. 建 Jetty 运行模块

(1) 新建模块

我们先新建一个 module，用于使用 jetty 来运行。

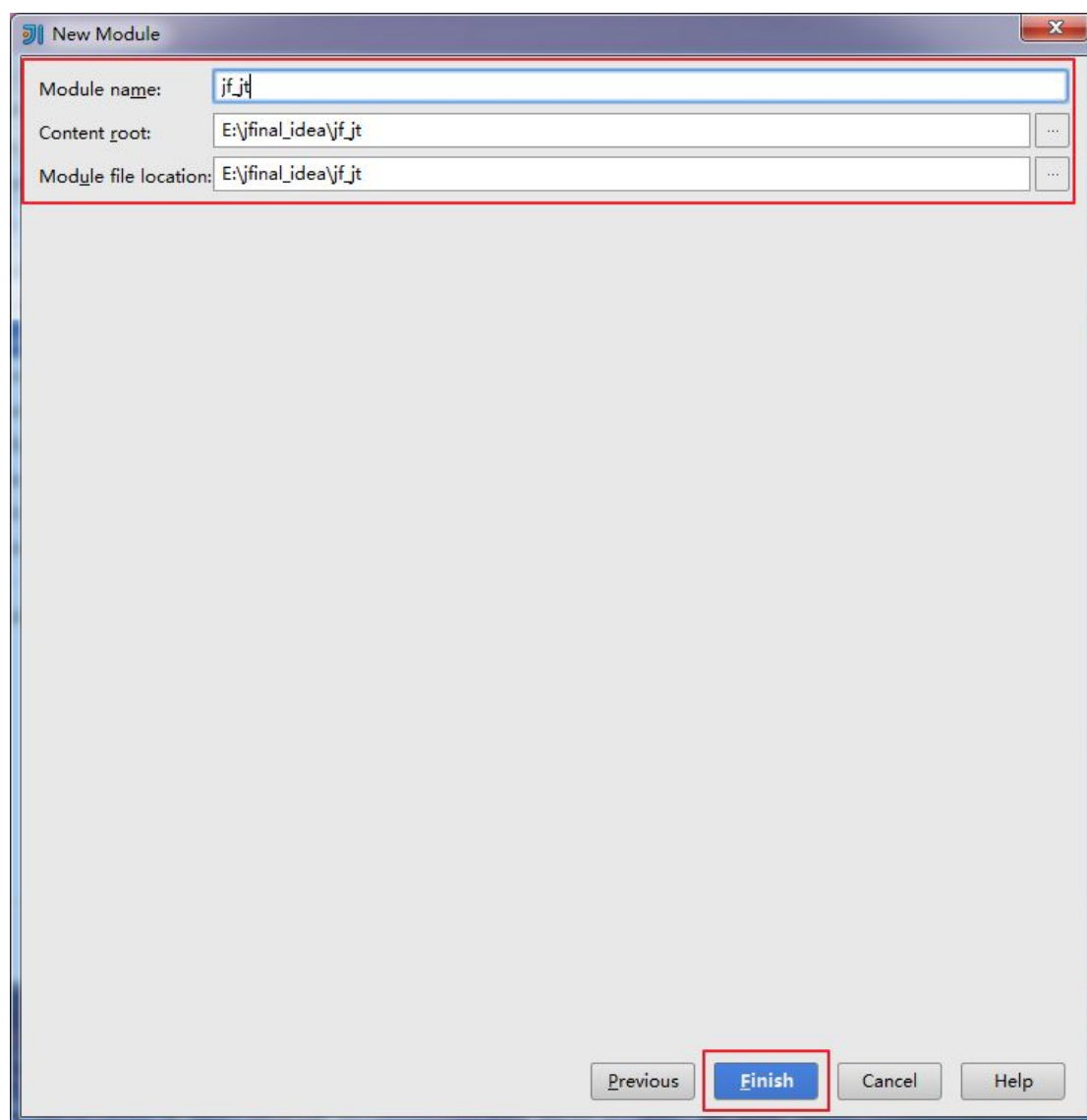


选上“Web Application”后，点击 Next。

在出现的窗口中，直接在 Module name 中输入想要新建的 module 名字，
下面的 Content root 和 Module file location 中会自动把路径填进去。

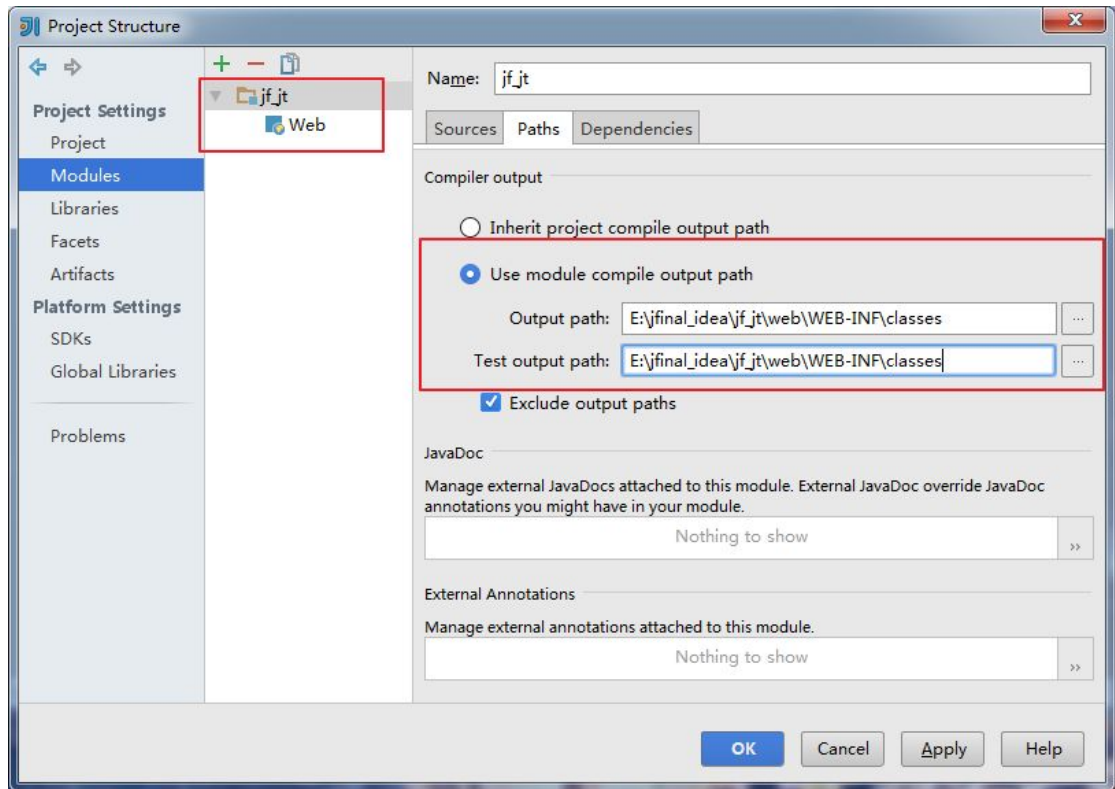
为了便于区分，我们把 jetty 运行的 module 命名为 jf_jt。

点击“Finish”。



此时会出现如下界面；

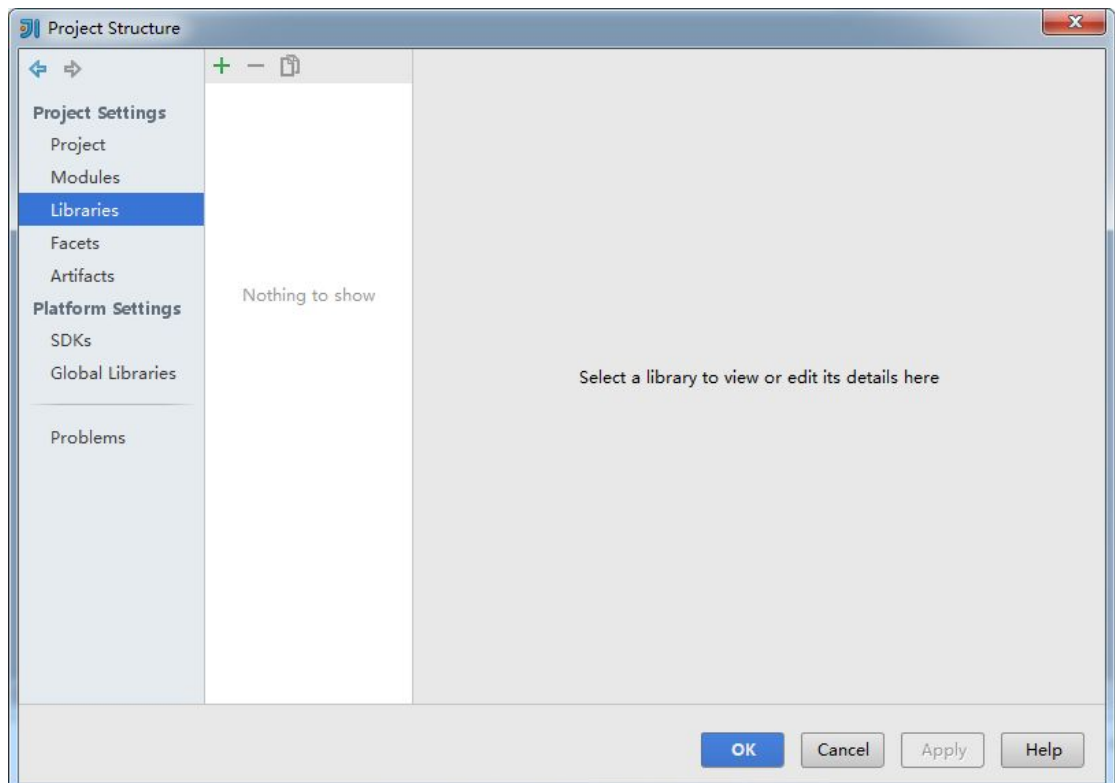
选择 **Paths**”选项卡，选中“**Use module compile output path**”后，在 **Output path**”和 **Test output path**”中均写上类输出的路径。按照一般常规写法，我把这个目录放在 **module** 下，**web\WEB-INF** 路径下的 **classes** 目录下。



点击“Apply”，把配置启用起来；

(2) 导入类库

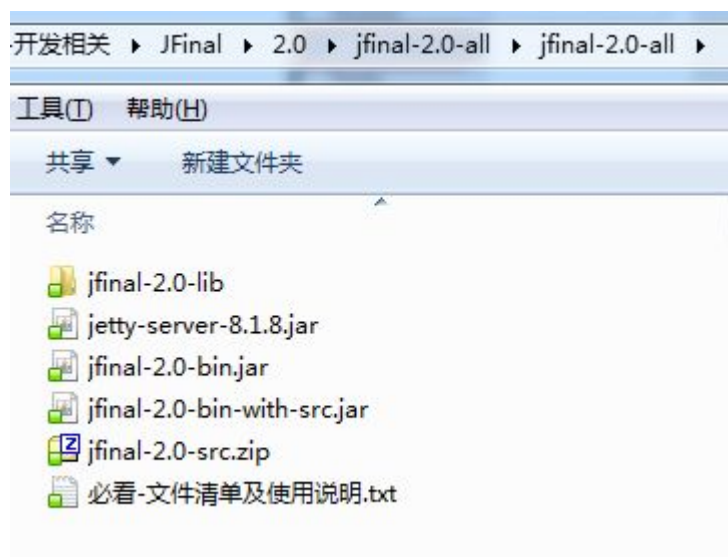
然后点击左侧的 Libraries 选项卡；



在做这一步之前，我们先要把需要的类库分别拷贝到我们建立的类库目录中。

这个模块是需要 **jetty** 来运行的，所以需要 **JFinal** 的类库和 **jetty** 的类库，事先准备好这几个类库。

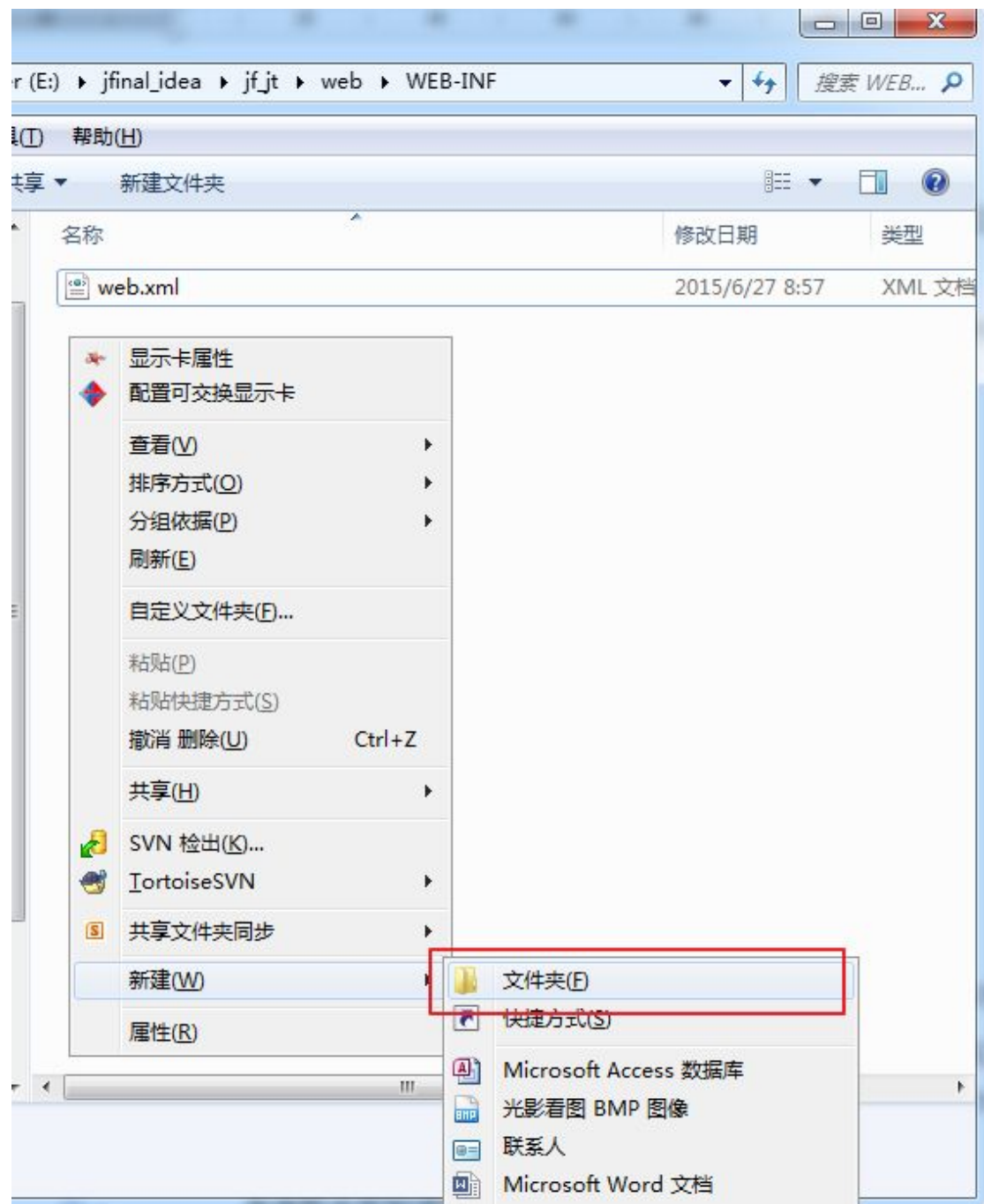
Jfinal-2.0-all 目录下有需要的类库文件。



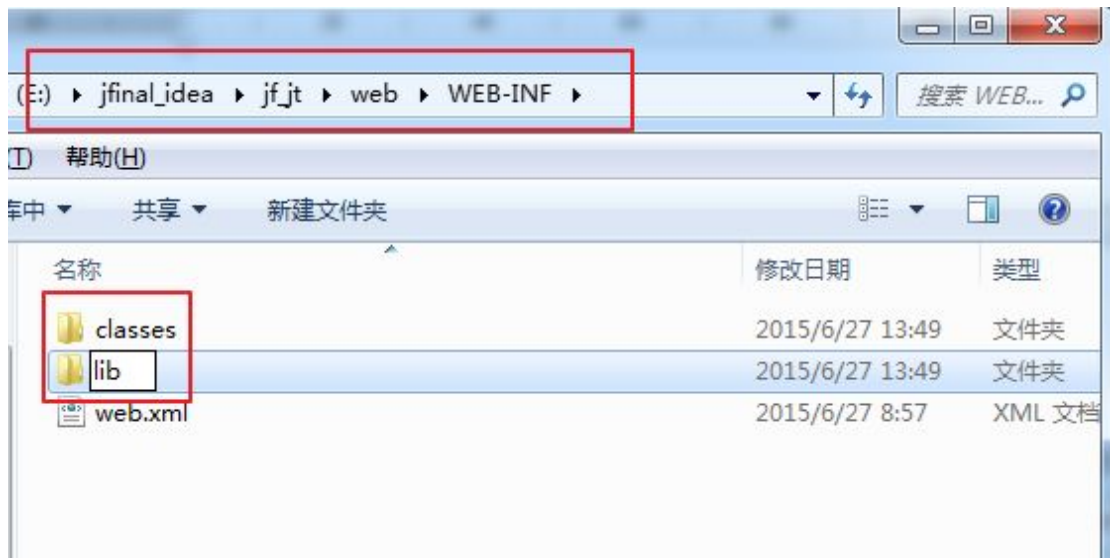
“jfinal-2.0-bin.jar”或“jfinal-2.0-bin-with-src.jar”是 jfinal 本身的 jar 包，任选一个都可以，为了方便调试，可以选择“jfinal-2.0-bin-with-src.jar”。

目前这个项目是为了 **Jetty** 而建的，所以要把“jetty-server-8.1.8.jar”也要包含进去。

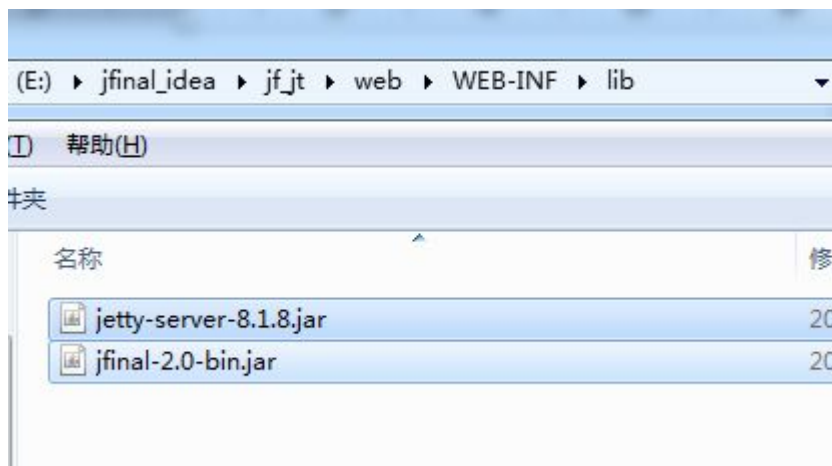
在电脑中找到项目目录，进入到 **Module** 路径中，新建文件夹；



新建一个 **classes** 目录(上文新建 **module** 时设置的输出目录)和 **lib** 目录;

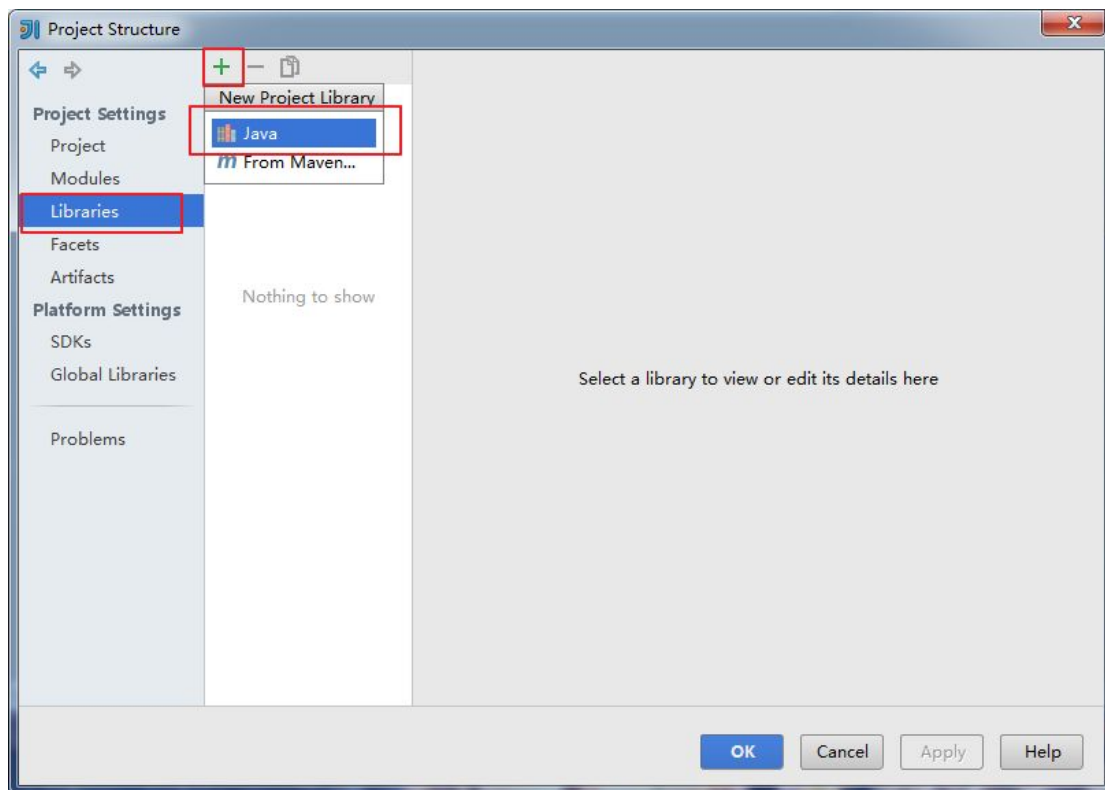


把“jfinal-2.0-bin.jar”和“jetty-server-8.1.8.jar”两个文件拷贝到刚才新建的 lib 目录下(注意，使用 jfinal-2.0-bin.jar 和使用 jfinal-2.0-bin-src.jar，后续界面会略有不同，但差异不大)。

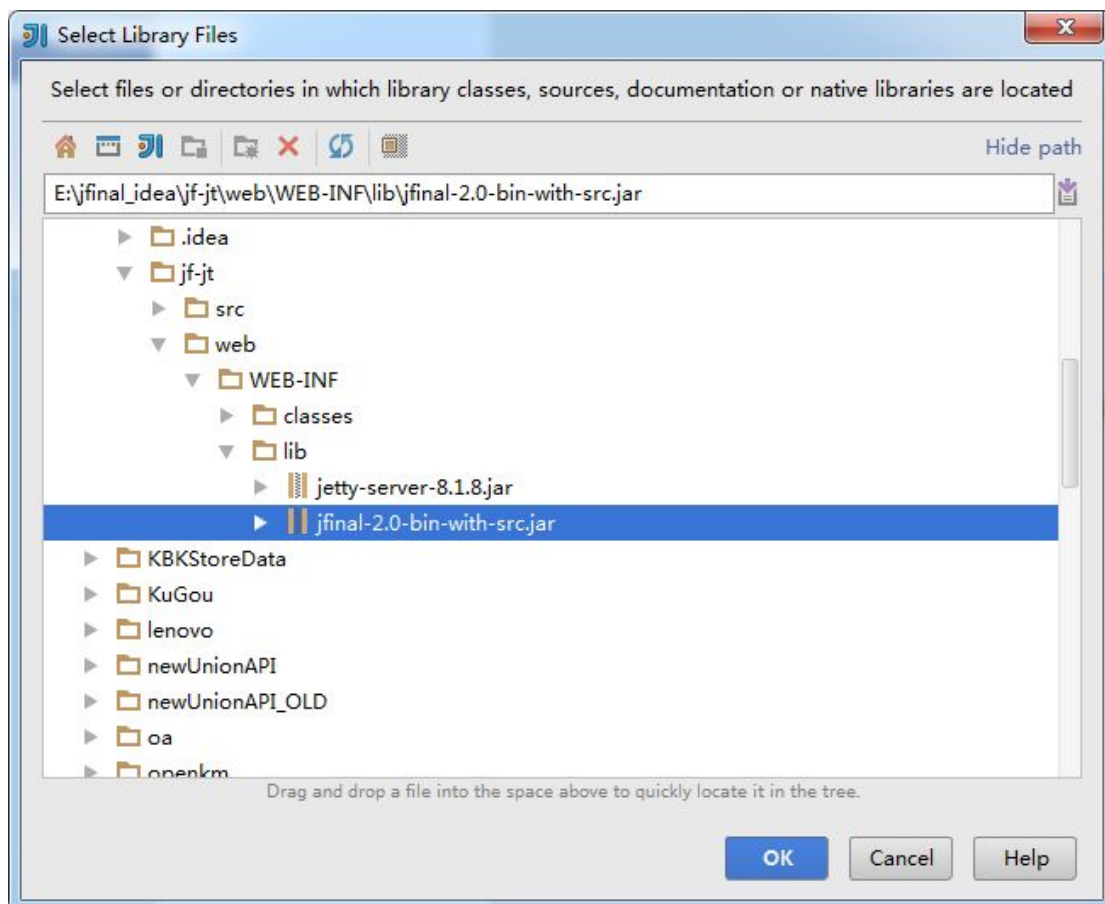


回到 IntelliJ IDEA 配置界面。

选择左侧的 Libraries 选项卡，点击中间的“+”号，新增 java 类库。



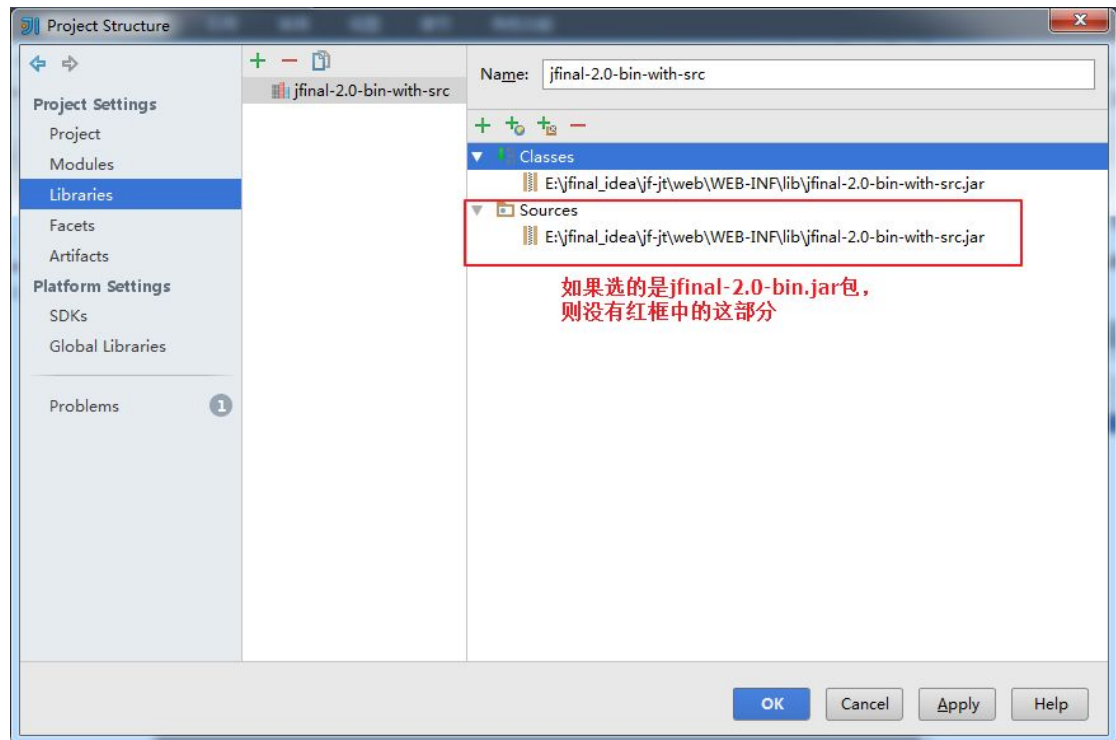
在弹出的窗口中，找到 lib 目录并选中刚才拷进去的“jfinal-2.0-bin-with-src.jar”包。



点击“OK”。

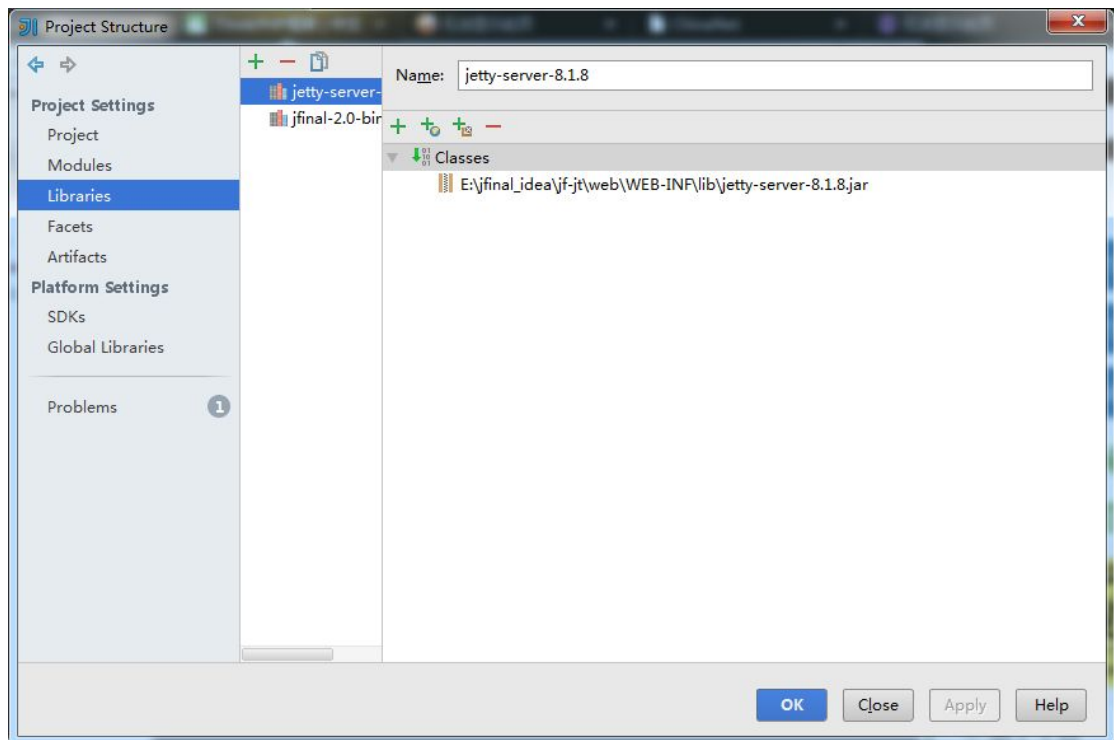
此时会让你选择这些类库将用于哪些 module（如果你在项目中有多 module,在此均会列出来）在这里，我们选择这些类库用于“jf_jt” module。

此时类库就会出现在列表里。

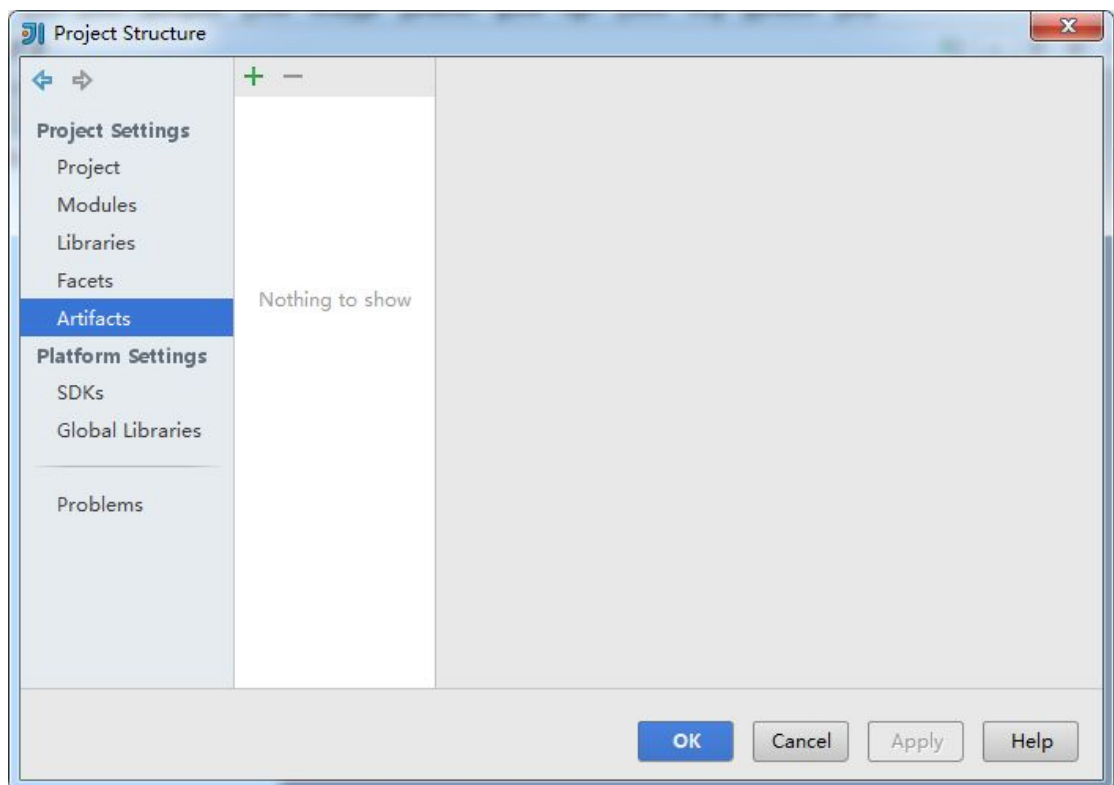


点击“Apply”，保存配置。

再重复上述步骤，导入 jetty 的 jar 包。导入后如下图所示：

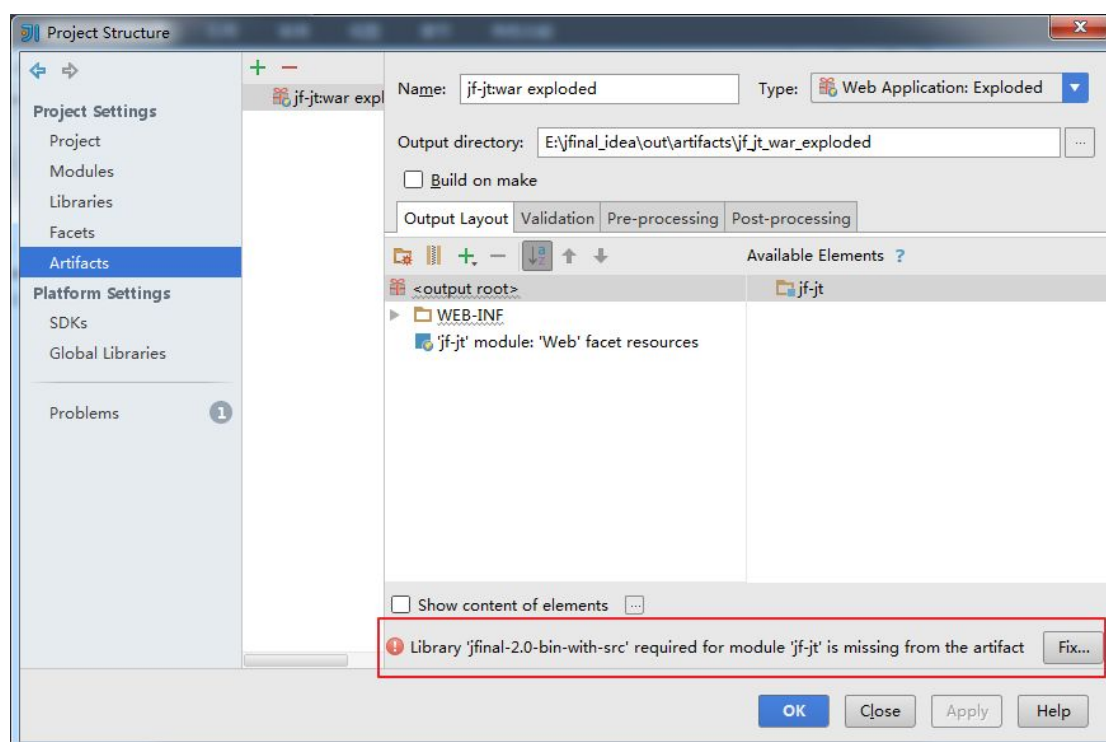


再选择左侧的“Artifacts”选项卡，此时会出现空白的 Artifacts 界面。



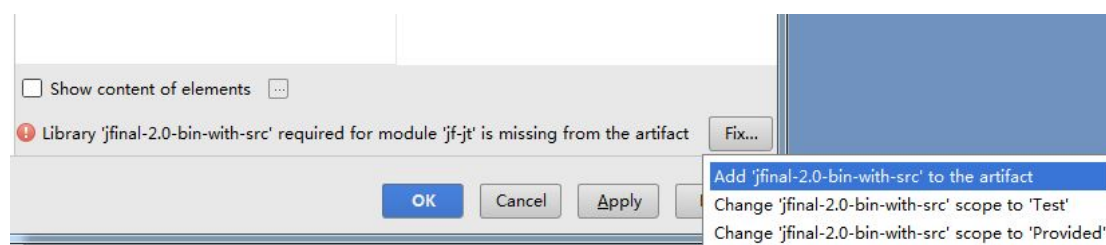
或出现已经有一个 `jf_ft.war exploded` 的界面(如果关闭后重新打开 Project Structure 界面就会出现)。建议：最好在导入类库后，点击“OK”，关闭本界面后，重新通过【File】-【Project Structure】菜单打开本界面后，选“Artifacts”选项卡进行操作，避免重复建立 `jf-ft.war exploded`。

重新打开界面，并选择 **Artifacts** 选项卡后，界面会如下图所示：



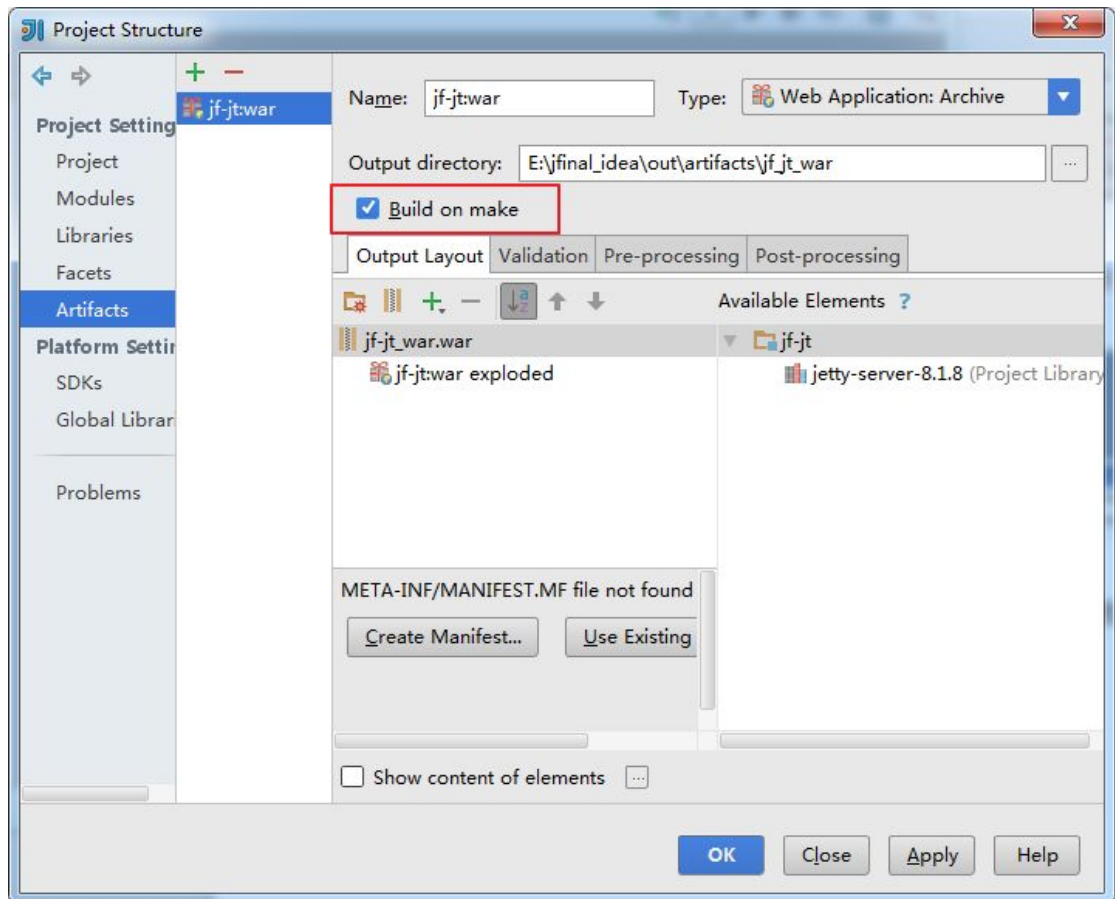
此时窗口下部会出现一条告警信息。如上图红框部分。

此时点击 “Fix...” 按钮，选择 “Add ‘jfinal-2.0-bin-with-src’ to the artifact”，即可。

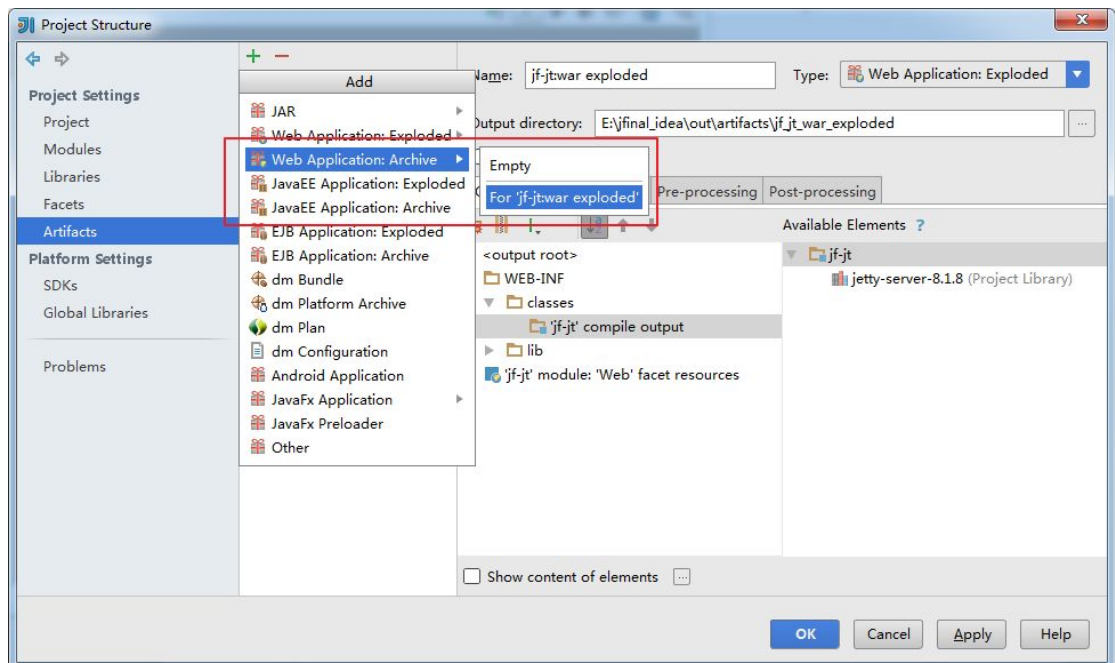


设置好的界面如下图所示（把 **Build on make** 选上）；

点击 “Apply” 按钮保存设置；

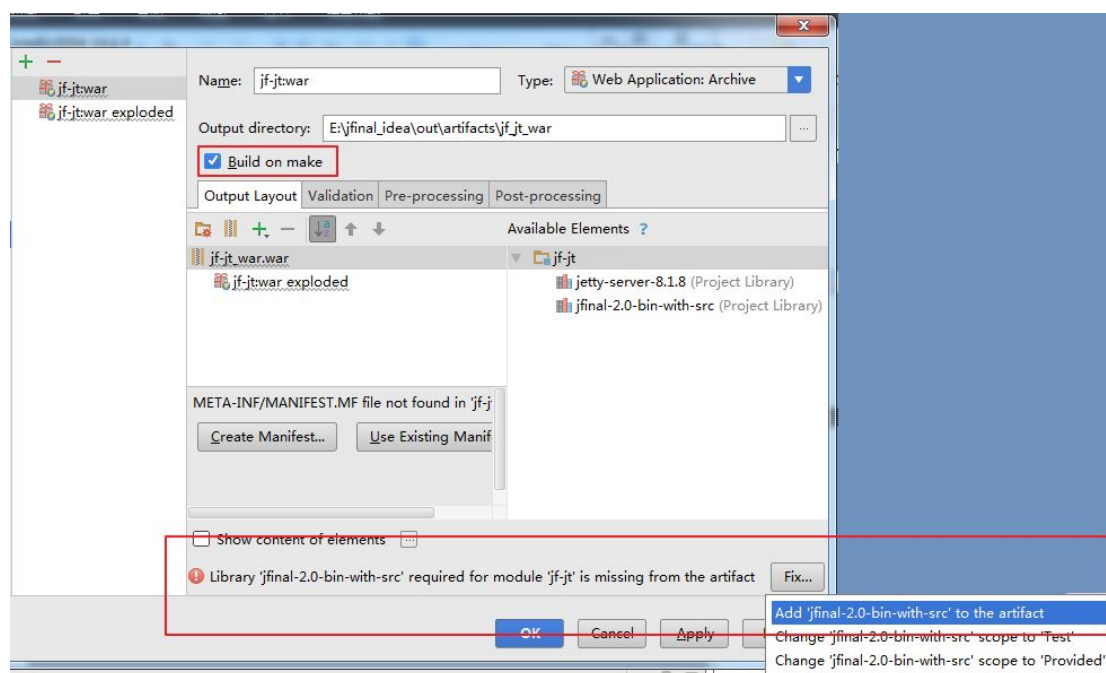


再点击“+”号，添加一个“Web Application: Archive”，此时选“For ‘jft:war exploded’”；



建好后，界面如下图所示(如果还有告警提示，按照上面步骤，点击“Fix”按钮把类库添加进 Artifact 中)；

选中“Build on make”选项后，点“OK”按钮保存并退出配置界面。

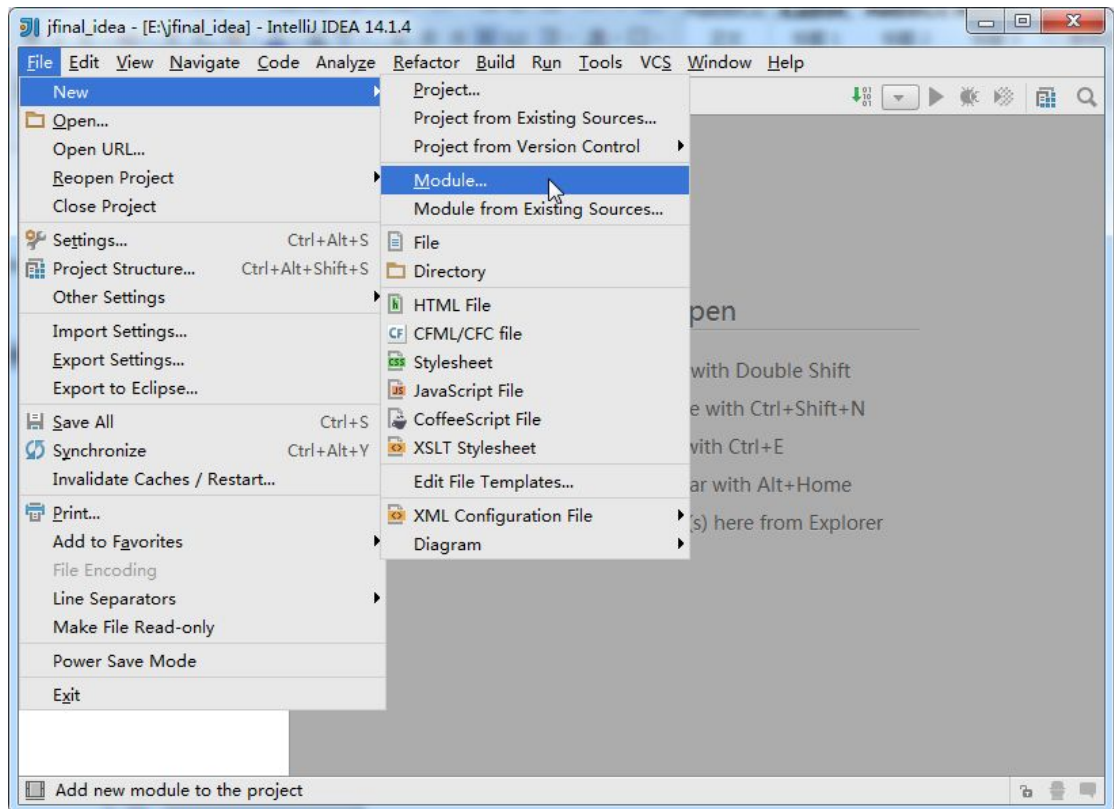


至此 Jetty 运行的模块已经建好。

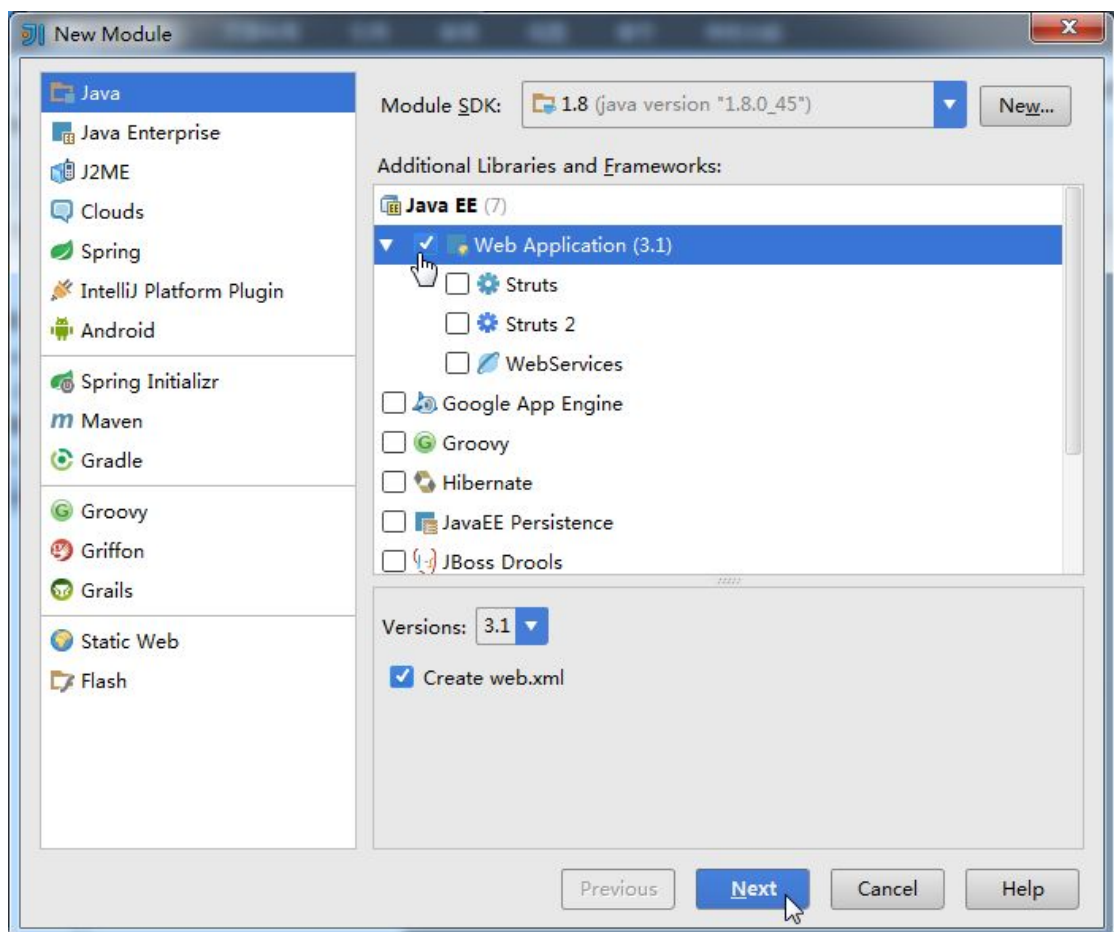
3. 建 Tomcat 运行模块

(1) 新建模块

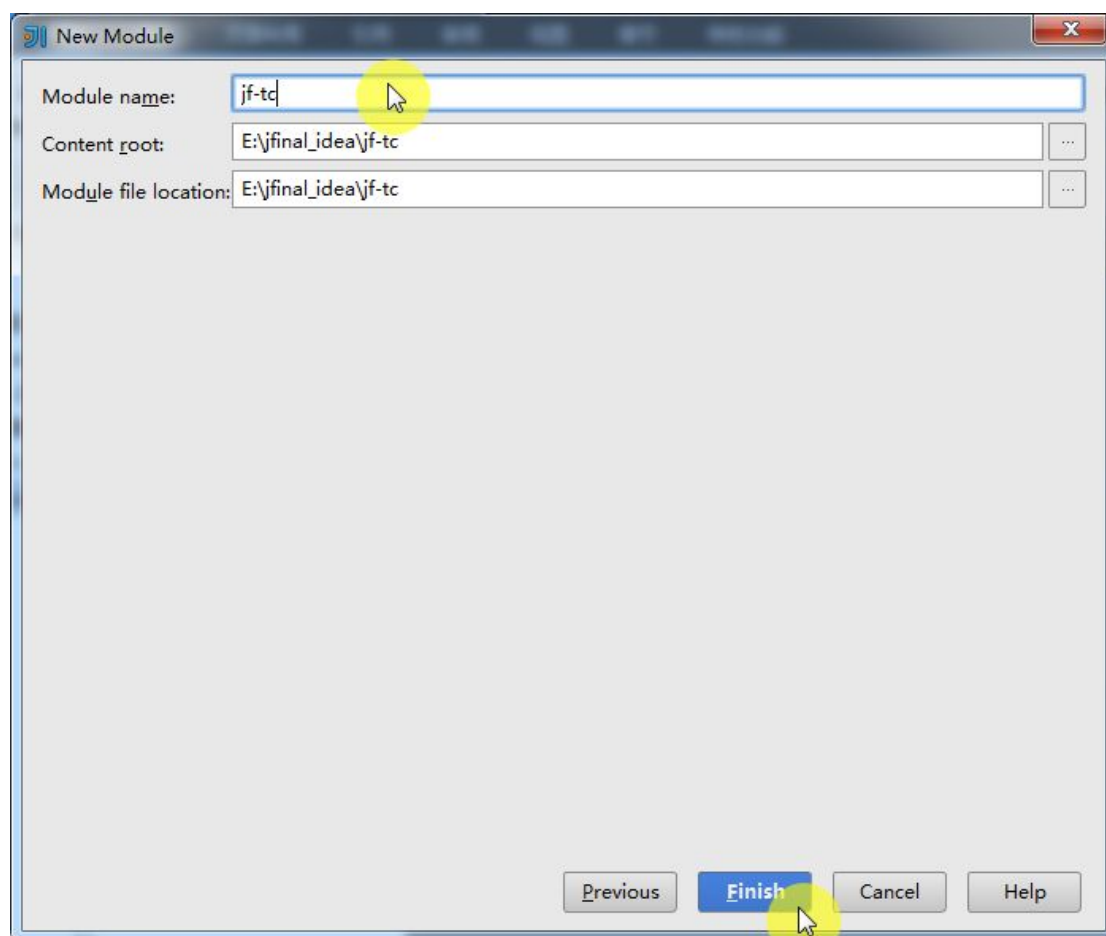
点击【File】-【New】-【Module】:



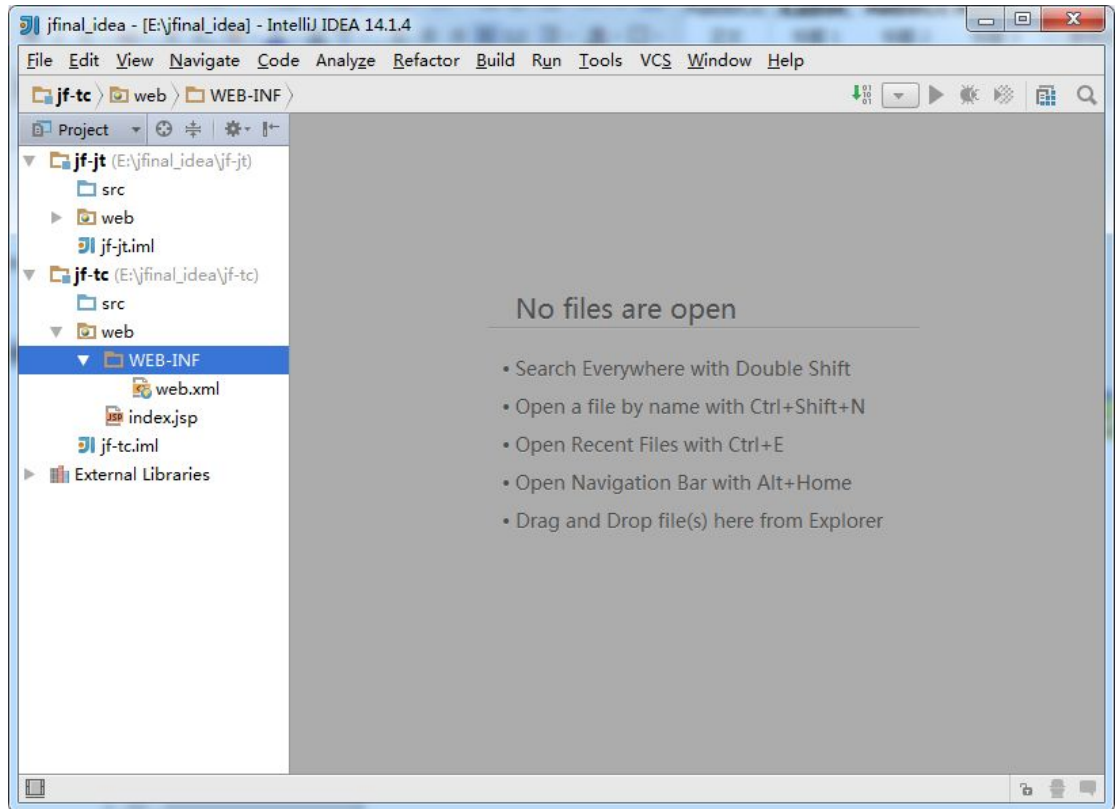
选上 “Web Application”，并点 “Next”：



填上 Module 的名称，这里起名为 jf-tc,然后点击 “Finish”:



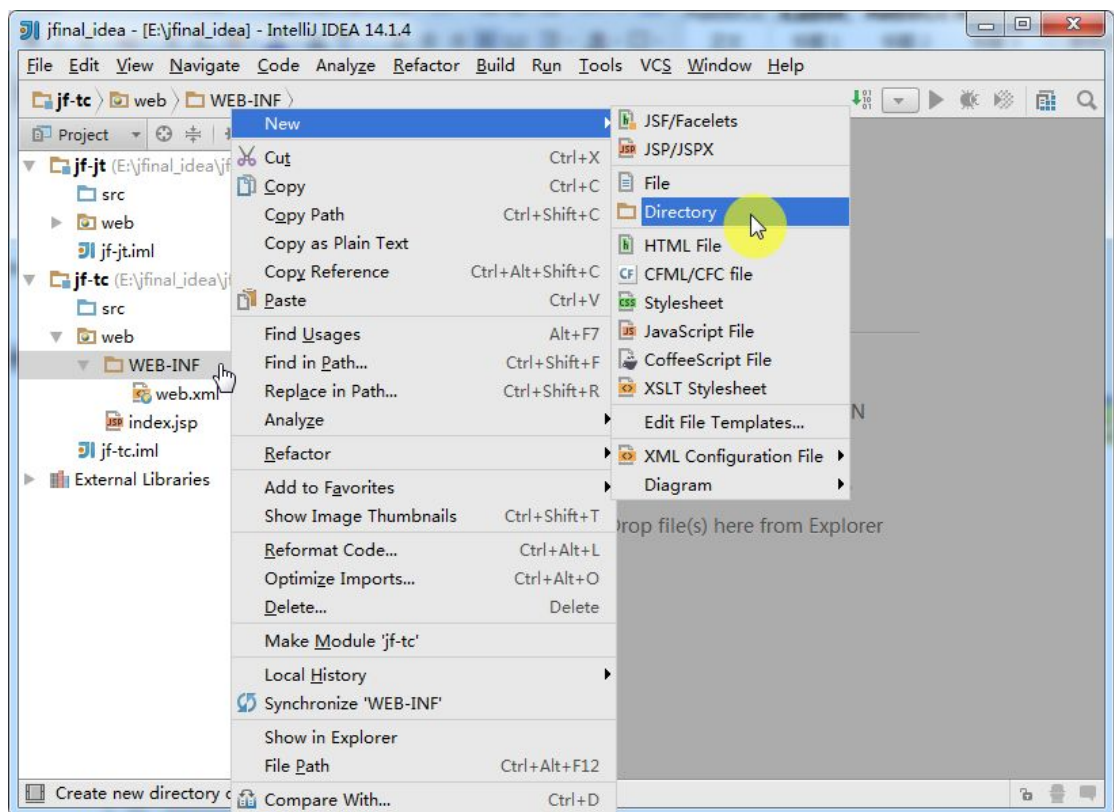
此时的界面是这样:



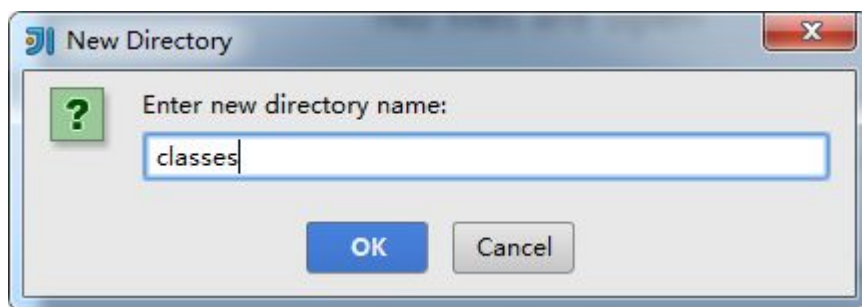
(2) 导入类库

现在可以直接在 IDEA 的界面建目录 `classes` 和 `lib`。

鼠标右键点击[jf-tc]-[web]-[WEB-INF]目录，在出来的菜单里点击【New】-【Directory】。



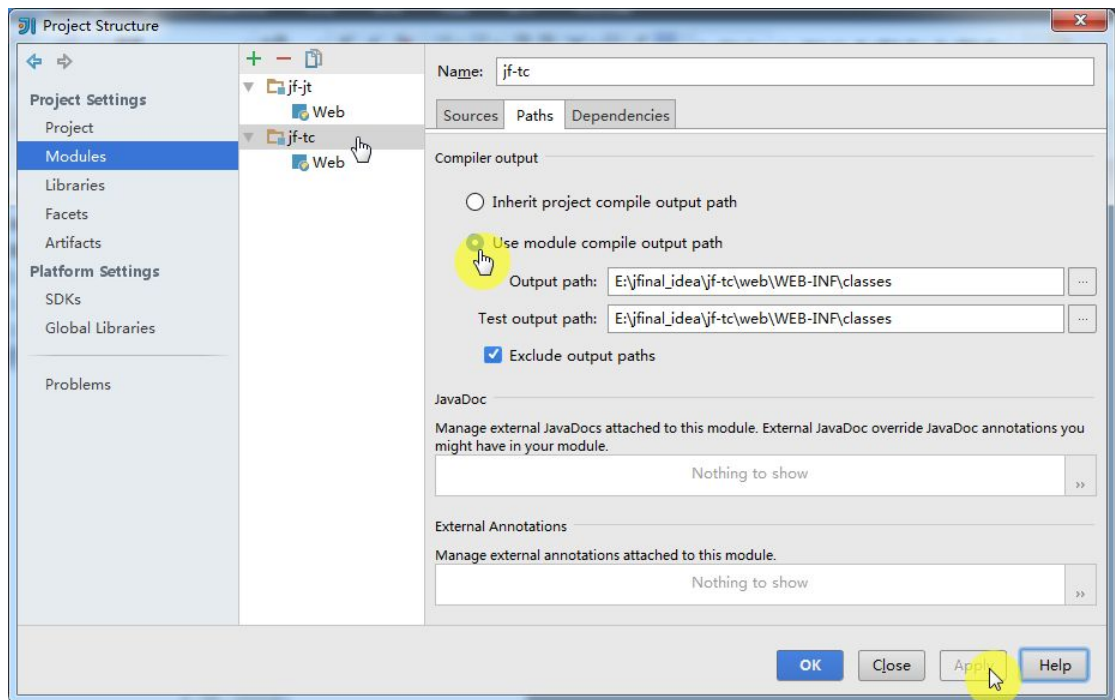
输入 classes 和 lib 目录的名称:



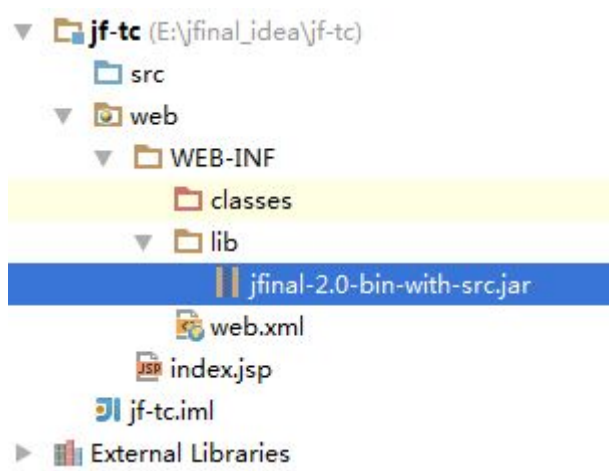
建好目录后的模块如下所示:



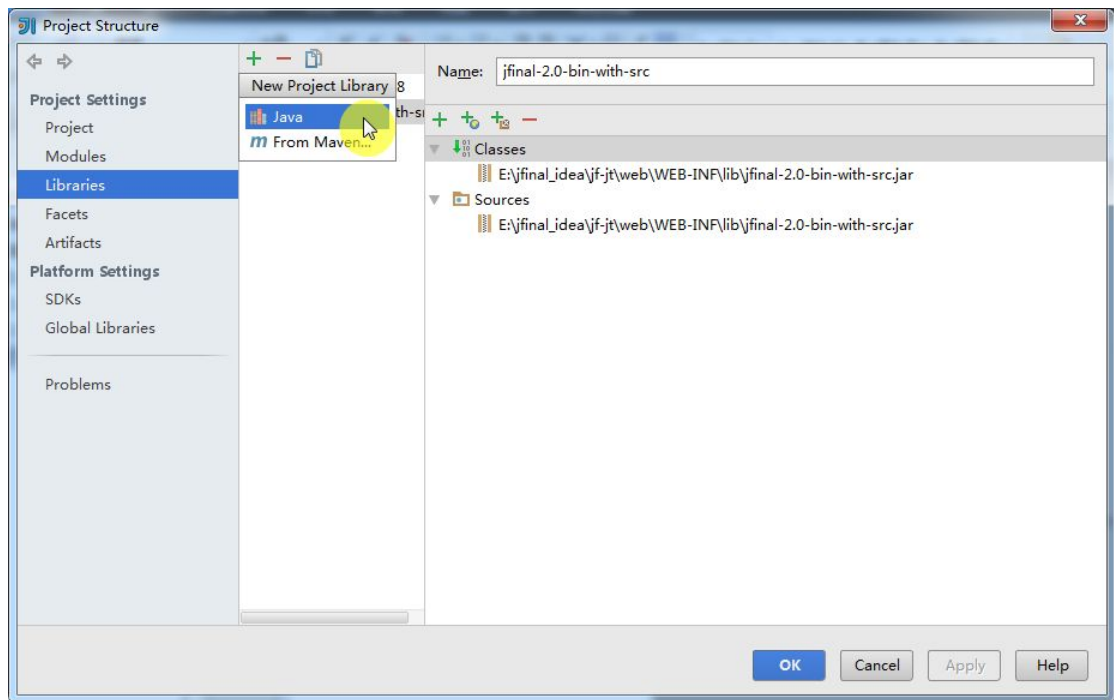
点击【File】-【Project Structure】进入项目，步骤和上一个模块建立的时候一样，此时界面里已经有两个模块，选中 jfinal_idea 模块后，设置 Path:



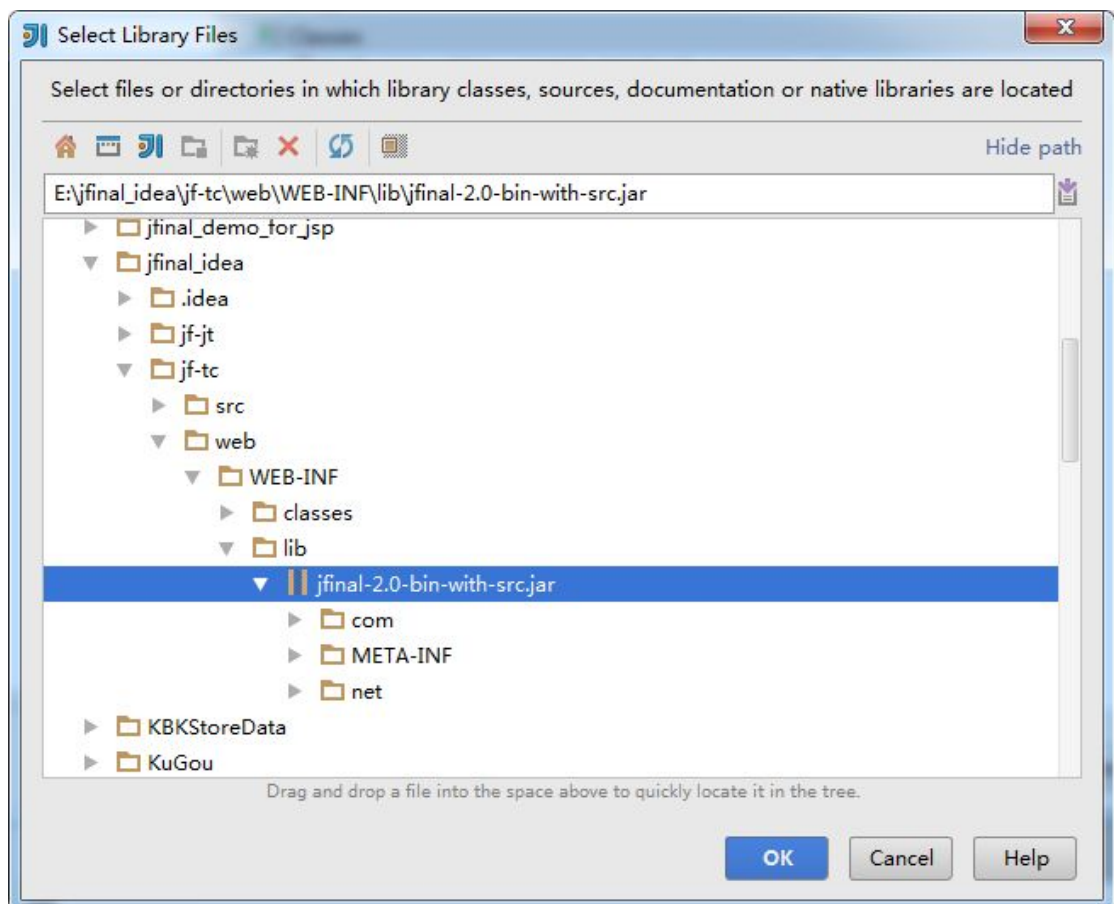
接下来再设置 Libraries，在此之前，需要把 **jfinal-2.0-bin-with-src.jar** 拷贝到 **lib** 目录下。



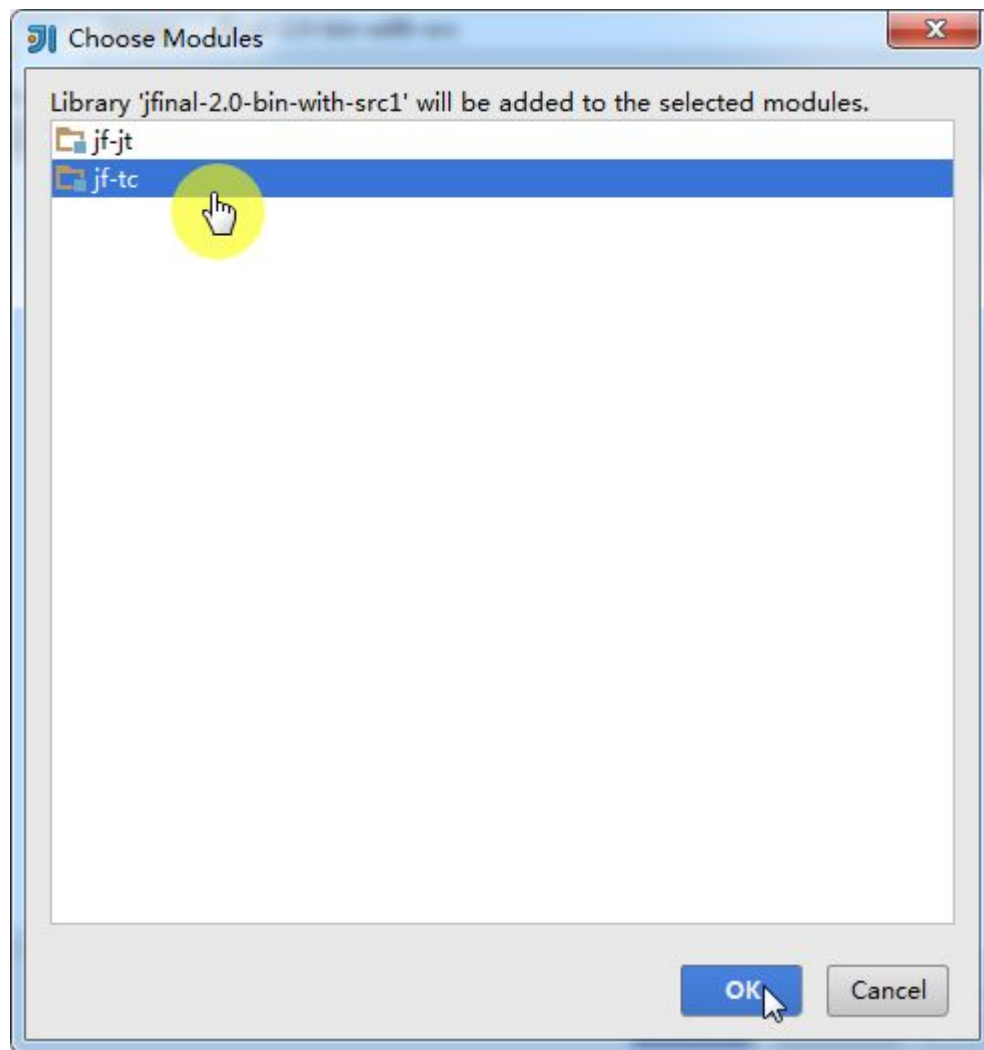
在“Libraries”选项卡中点“+”号，选“Java”：



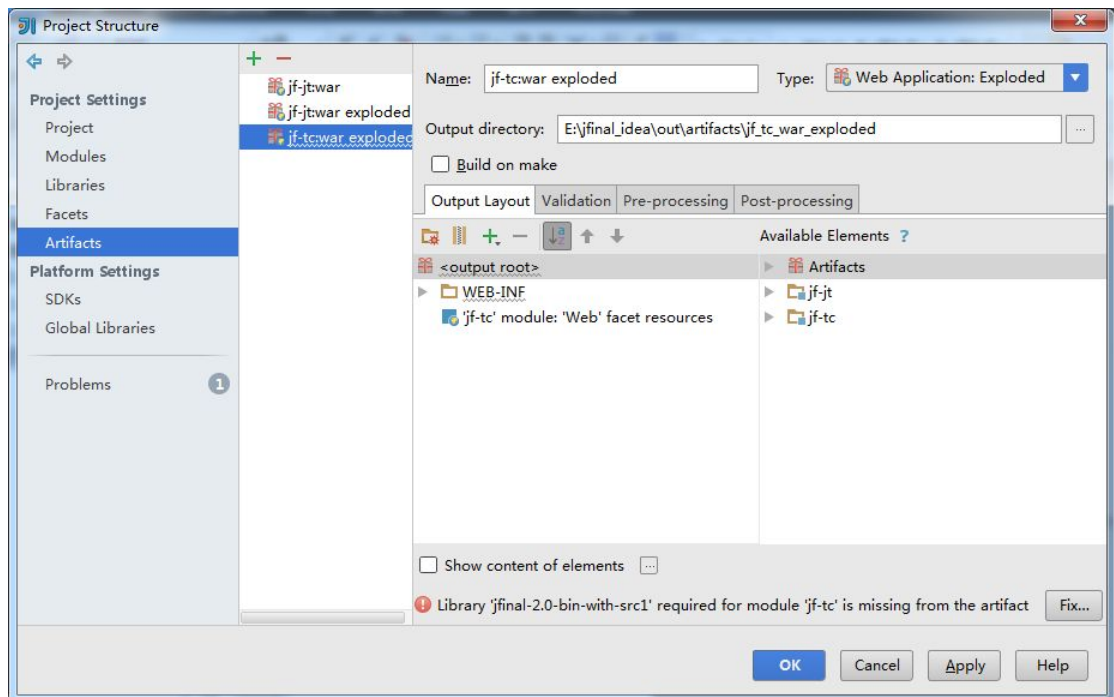
这里只需要导入一个 jar 包即可(记住,目录不要选错,要选刚才拷进去的 jf-tc 模块下 lib 目录的 jfinal-2.0-bin-with-src.jar 文件)。



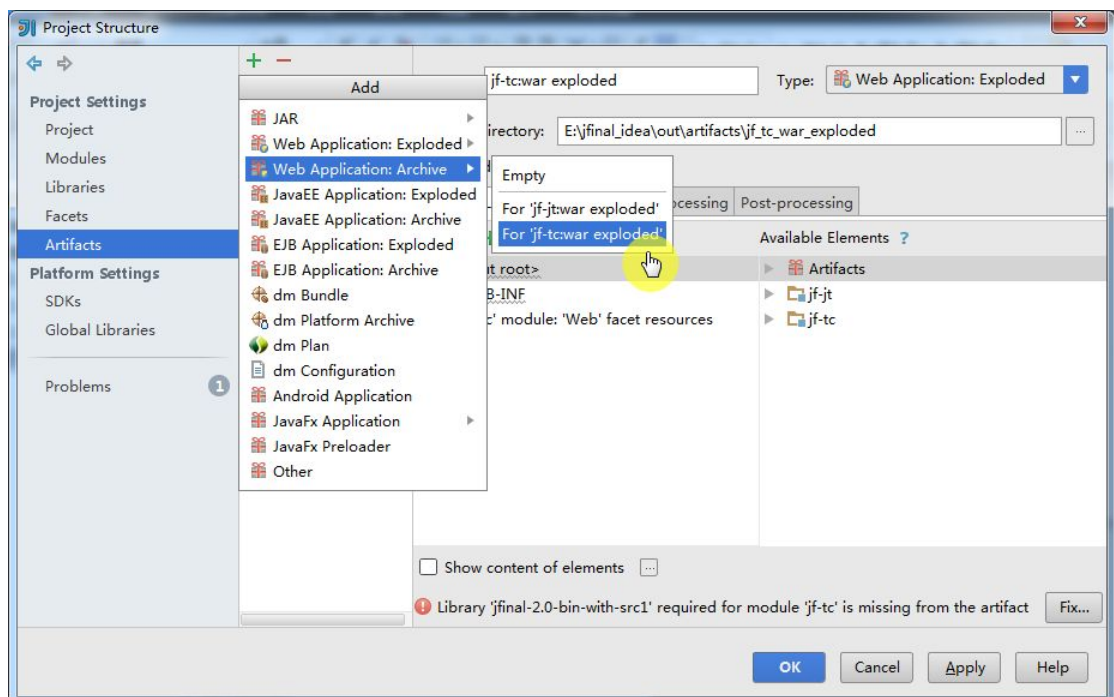
注意:选中目标 Jar 包后,选模块的时候,一定不要选错。



点“OK”后，来到 **Artifacts** 选项卡：

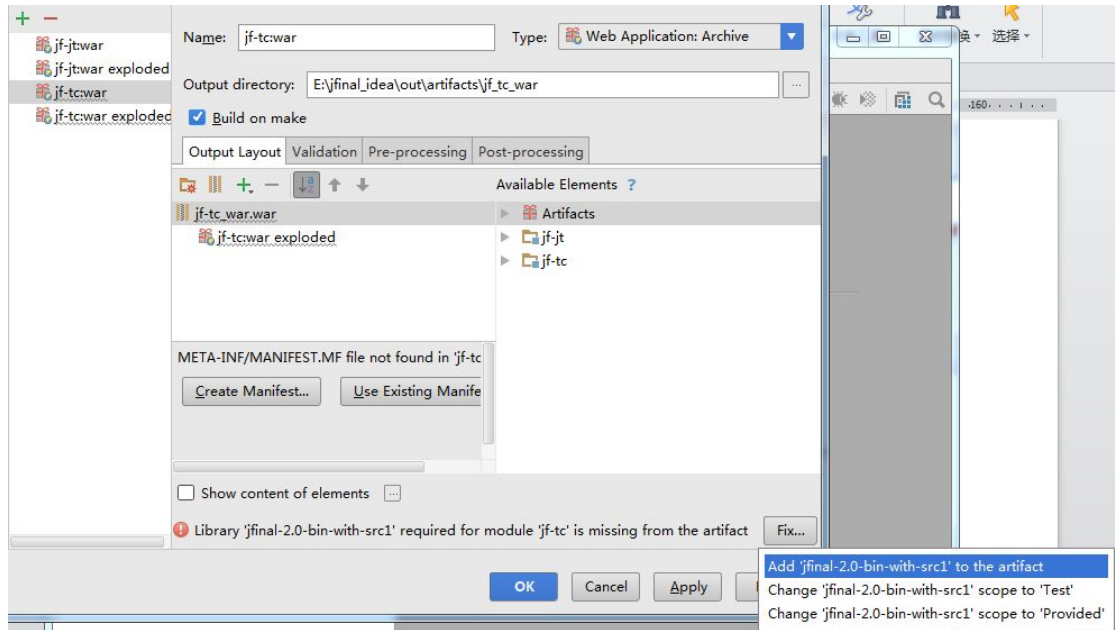


添加一个 Artifact



记住不要选错模块。

添加成功后，记住 **Fix** 掉警告信息(新加的两个 **Artifact** 都要 **fix**)。然后都选上 “**Build on make**” 选项。

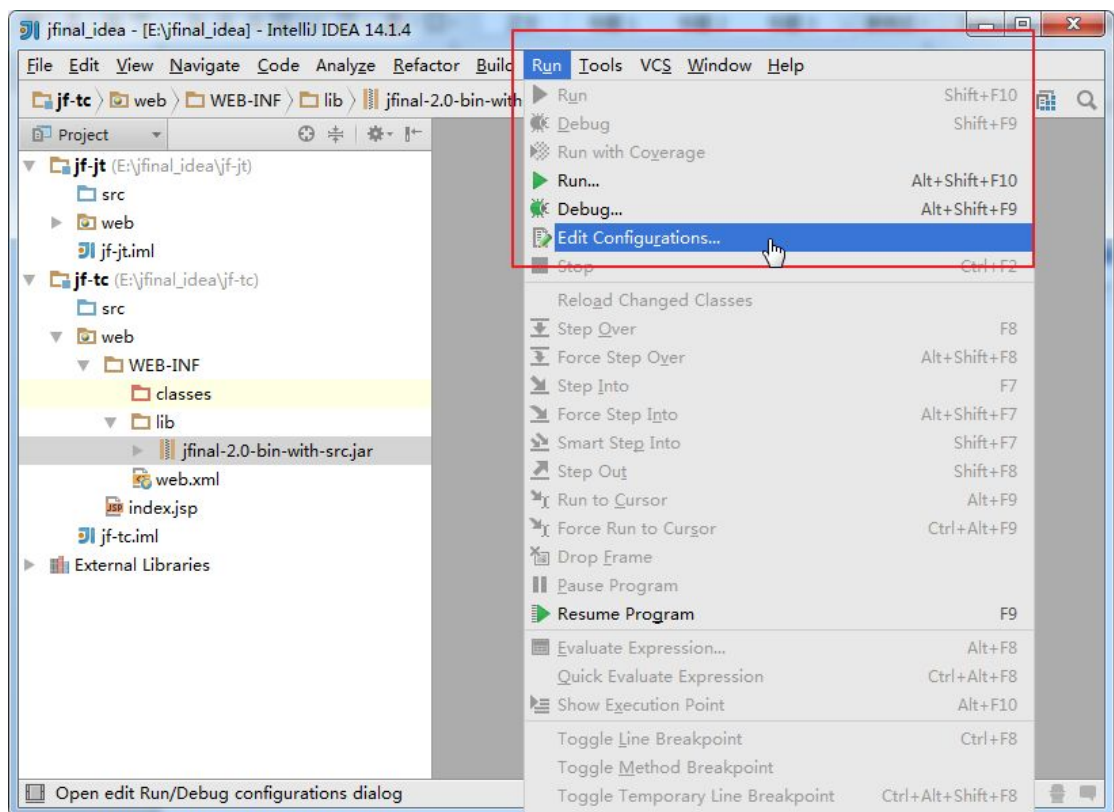


至此，两个模块都建好，可以开始真正的编写代码之旅了。

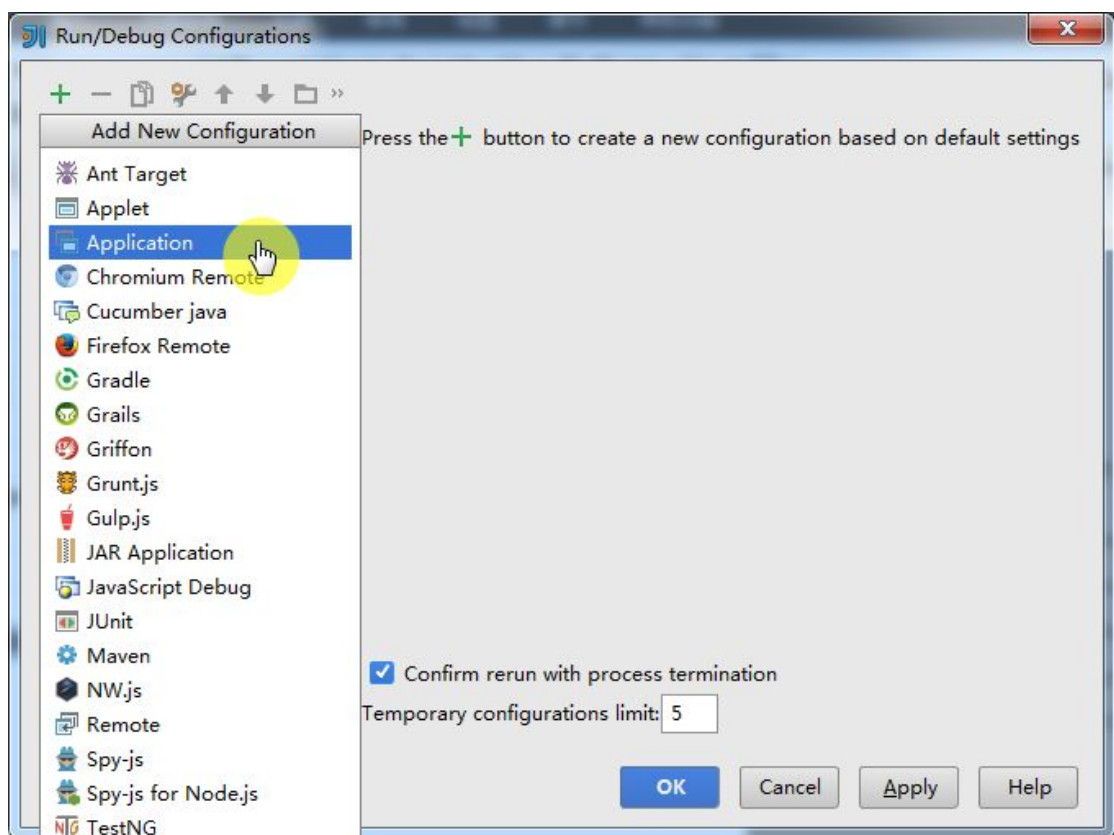
三、修改运行配置

1. 创建 Jetty 运行配置

点击【run】-【Edit Configurations】菜单：



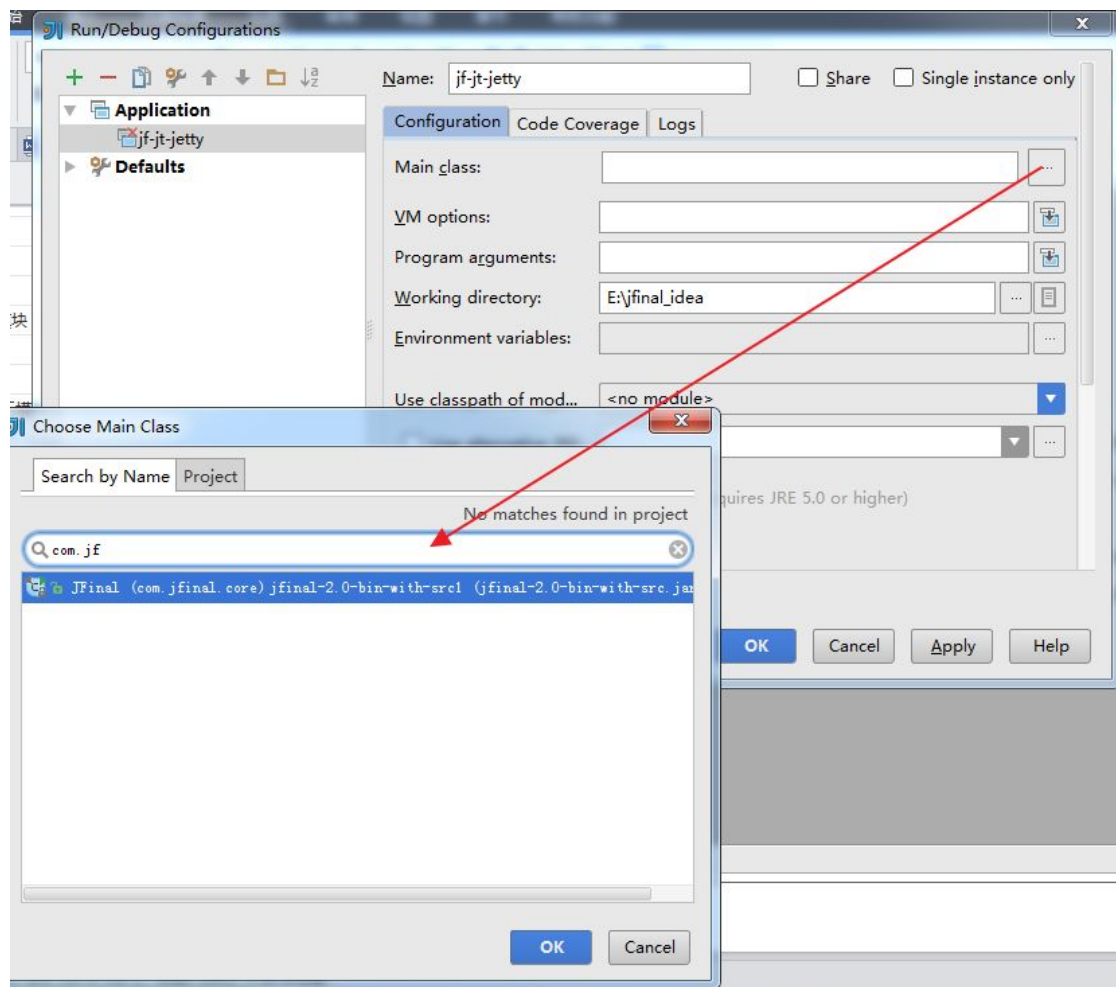
出现下面的界面，并点击左上角的“+”号，选“Application”选项：



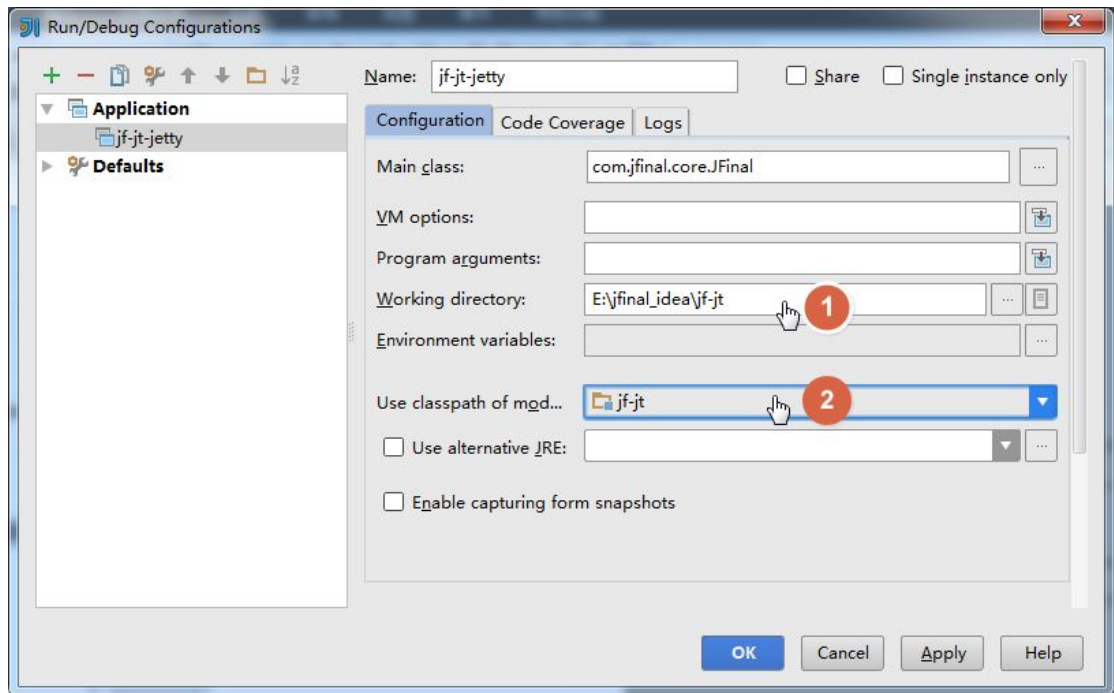
设置一个名称，在这里命名为“jfinal-jetty”。

设置 Main Class(在出来的“Choose Main Class”窗口里直接输入

`com.jfinal`……; 下面会直接把 `class` 列出来, 不用搜索和查找).



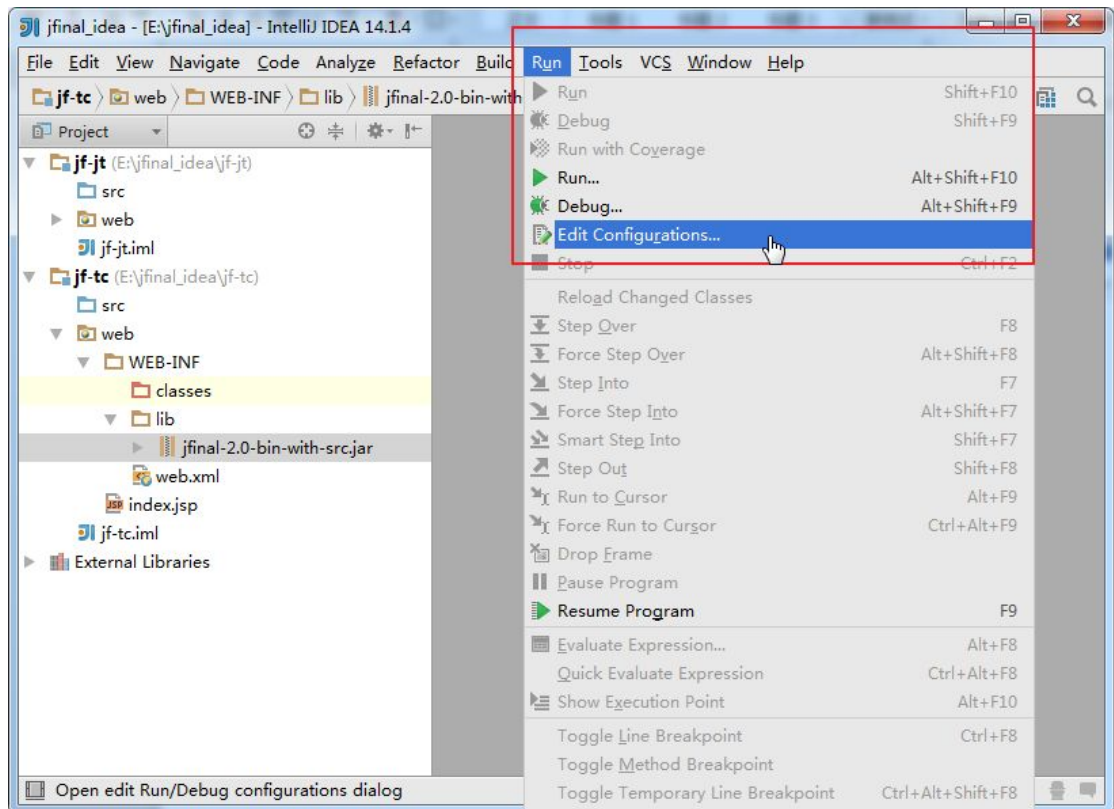
设好 Main Class 以后, 设置 “working directory” 和 “Use classpath of module” 两项, 如下图所示:



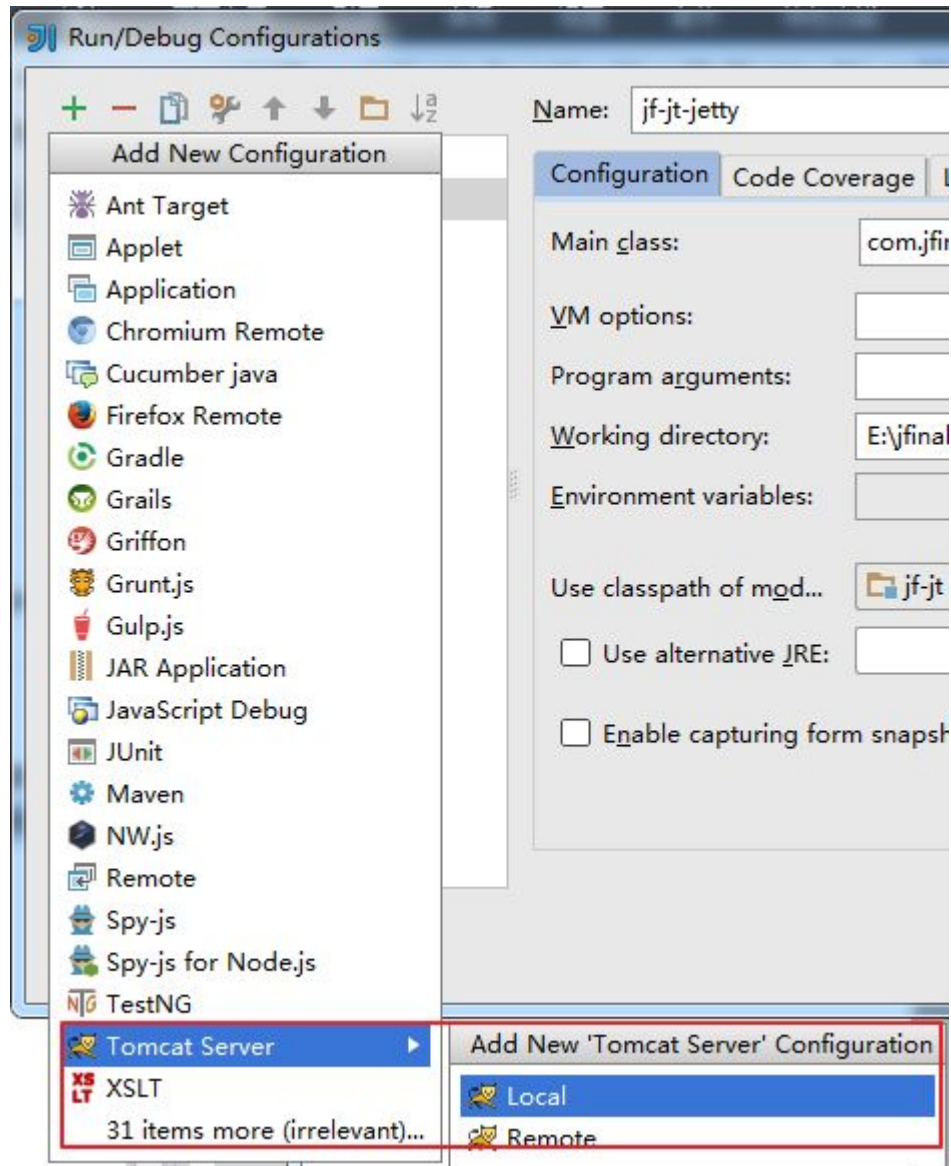
点击“OK”，设置完成。

2. 创建 tomcat 运行配置

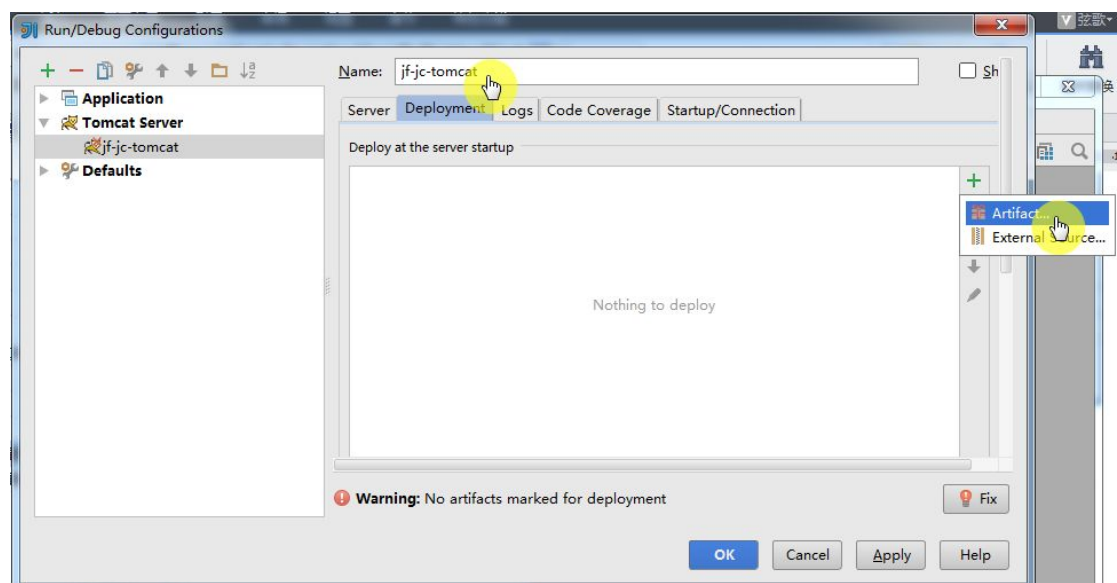
点击【run】-【Edit Configurations】菜单：



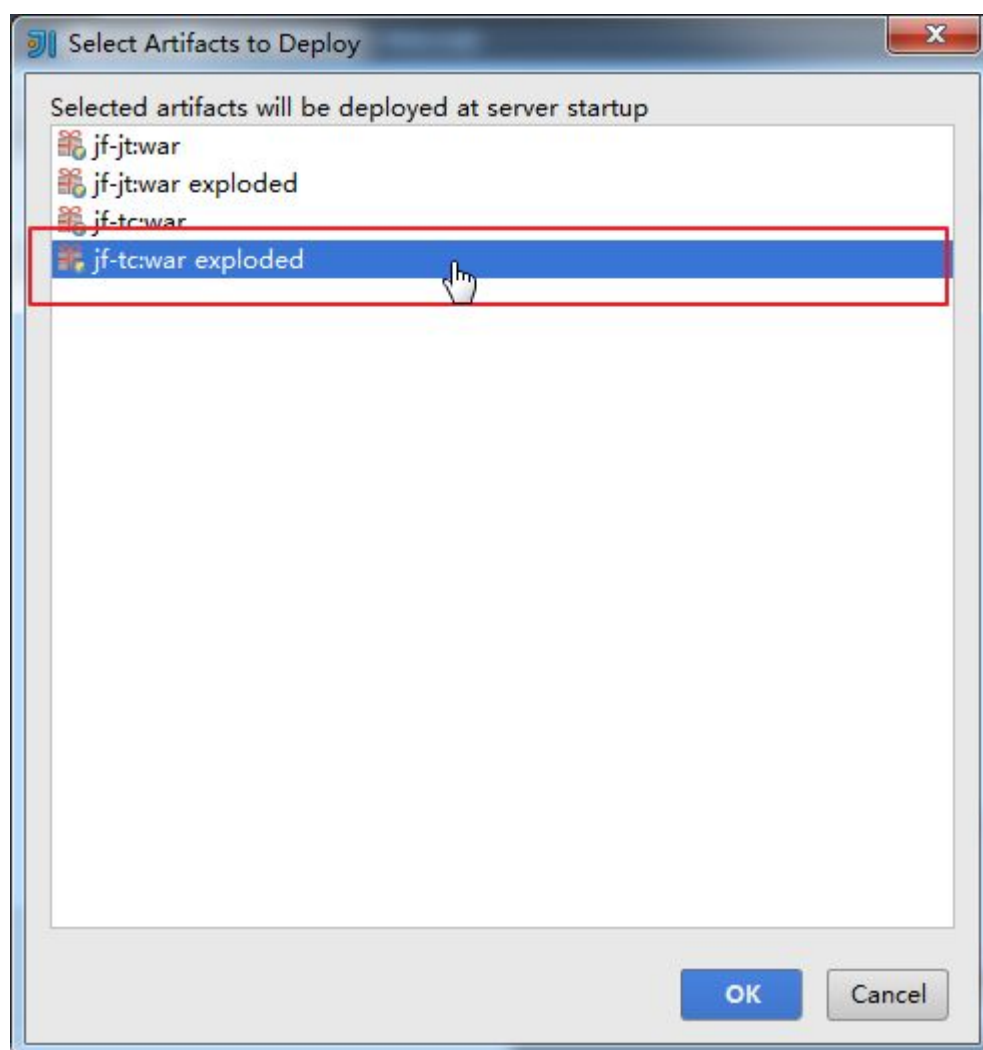
选择【Tomcat Server】-【local】菜单：



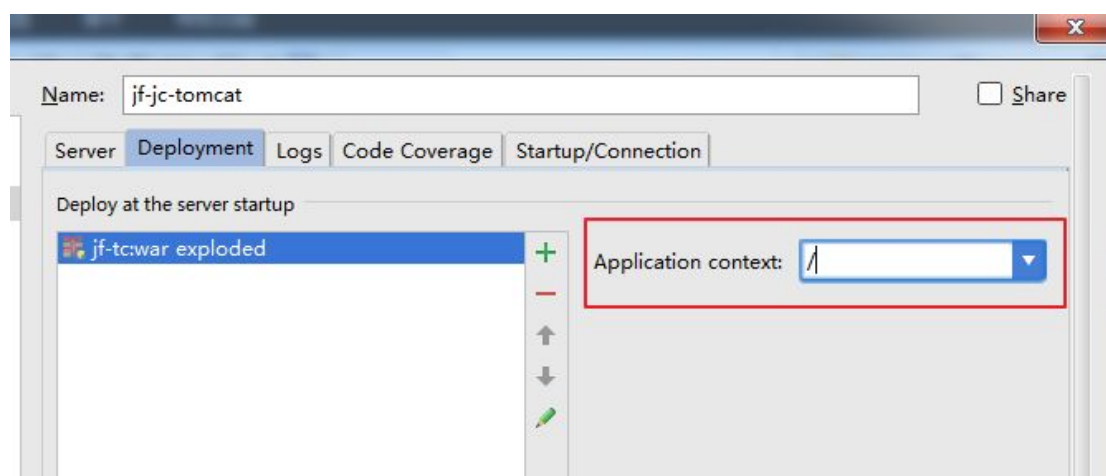
配置 Tomcat 参数,起一个名字,然后点击最右边的“+”号,增加一个 Artifact:



在出来的窗口中，选“jf-tc:war exploded”即 exploded 的那个 war。



点击“OK”后，就配置完成了。



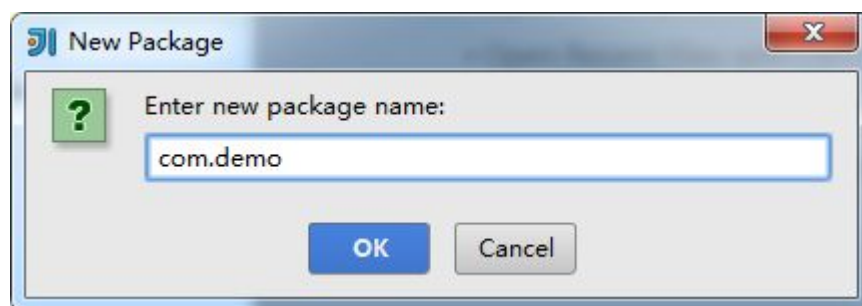
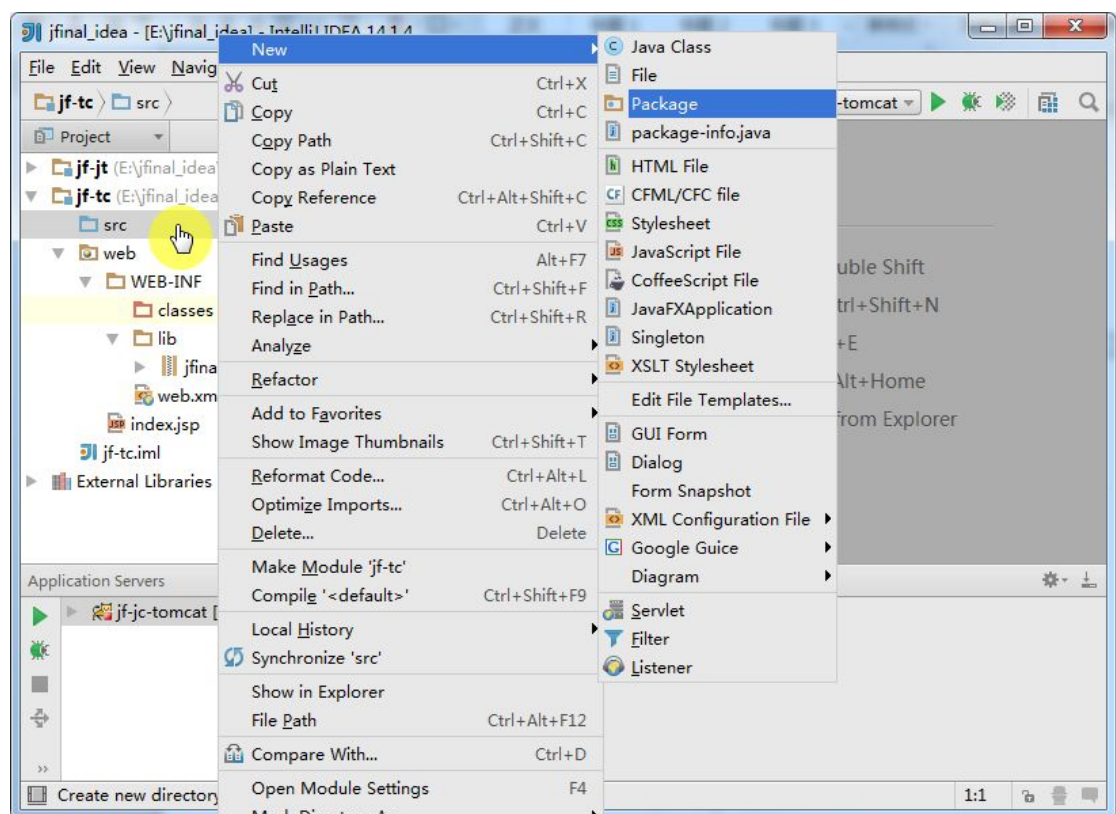
注意，这个 Application context 里填写的路径，是你调试或运行时出现的 url 的后缀，比如，如果你在此设置“/”，则最后是通过“http://localhost:8080/”

运行和调试；假如你在此设置为“/test”，则最后是通过“http://localhost:8080/test”进行访问和调试。

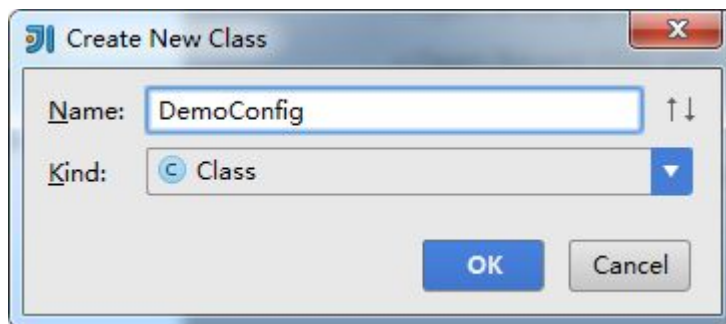
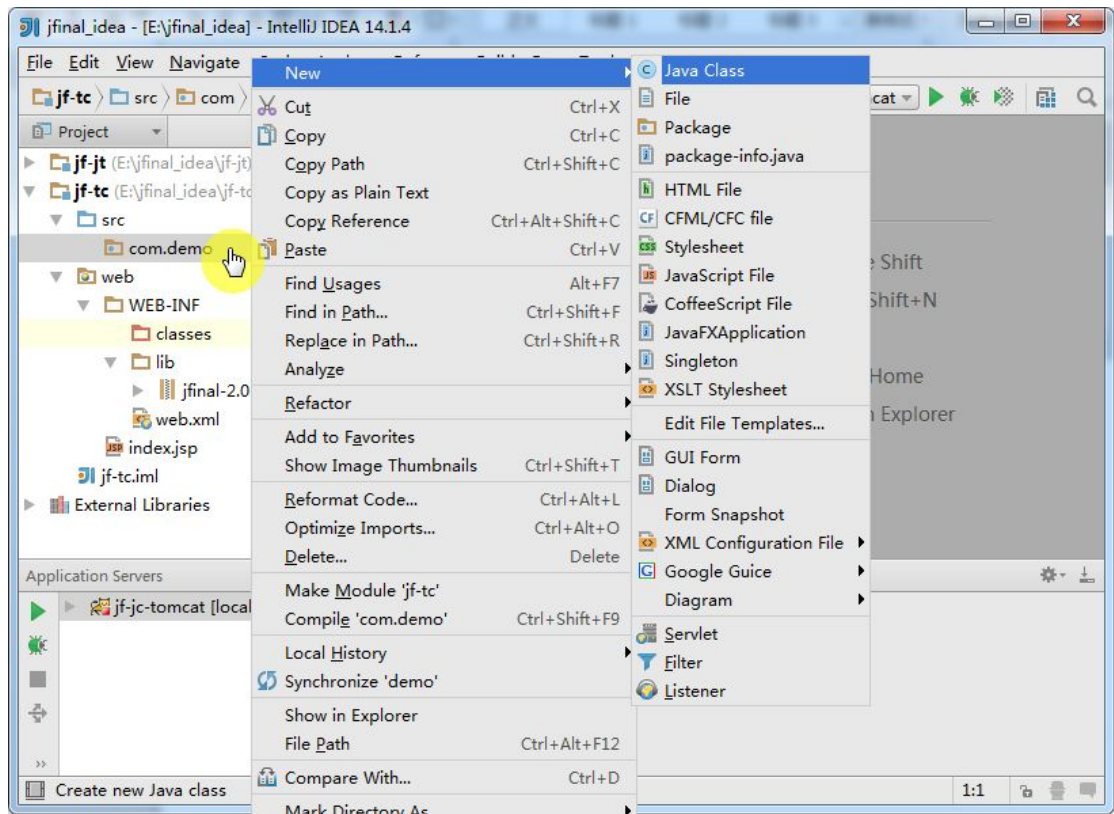
四、添加源文件

现在可以添加源文件了。我们可以分别在两个项目下建立源文件(建源文件的过程不管上面 jetty 和 tomcat 项目都是一样的，我们以 tomcat 项目来举例)：

我们在 src 路径下，添加一个 package 名称为 com.demo。



再在这个 package 下建三个类：



内容如下：

```
DemoConfig.java x
package com.demo;

import com.jfinal.config.*;
import com.jfinal.ext.handler.ContextPathHandler;
import com.jfinal.render.ViewType;

/**
 * Create by 弦歌
 */
public class DemoConfig extends JFinalConfig {
    public void configConstant(Constants me) { me.setDevMode(true); }
    public void configRoute(Routes me) {
        me.add("/hello", HelloController.class);
        me.add("/", IndexController.class);
    }
    public void configPlugin(Plugins me) {}
    public void configInterceptor(Interceptors me) {}
    public void configHandler(Handlers me) {}
}
```

另外再建两个类 HelloController.java 和 Indexcontroller.java,内容分别如下:

```
HelloController.java x IndexController.java x
package com.demo;
import com.jfinal.core.Controller;

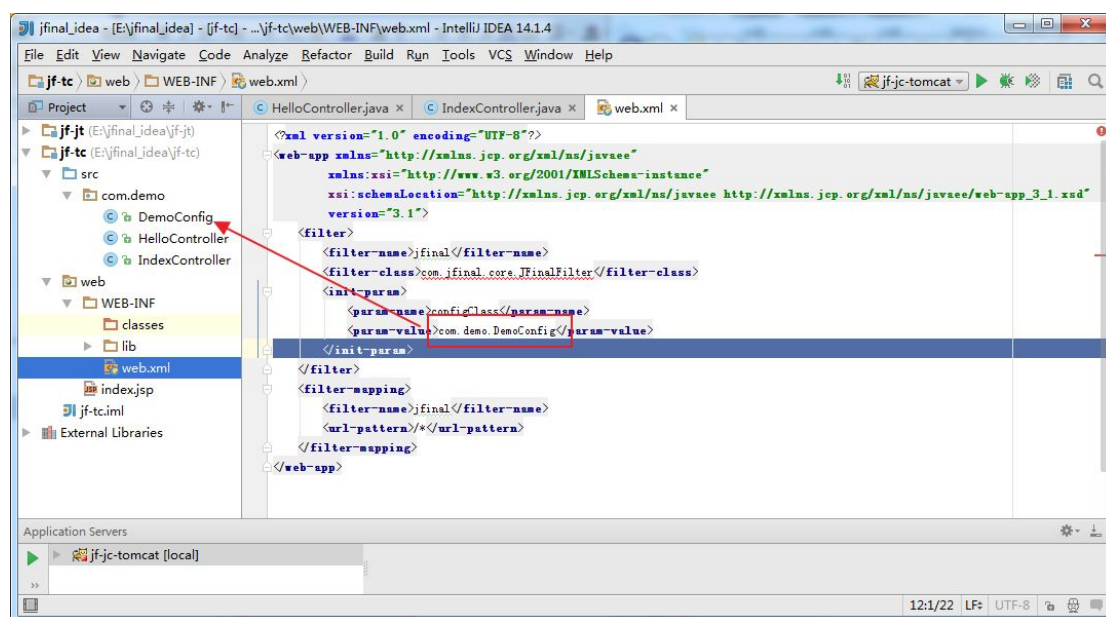
/**
 * Created by 弦歌
 */
public class HelloController extends Controller{
    public void index() { renderText("欢迎来到JFinal的世界!"); }
}
```

```
HelloController.java x IndexController.java x
package com.demo;

import com.jfinal.core.Controller;

/**
 * Created by 勇 on 2014/11/10.
 */
public class IndexController extends Controller{
    public void index() { renderText("这是Tomcat下的index页面!"); }
}
```

编辑模块下，web\WEB-INF 路径下的 web.xml 文件，内容如下：

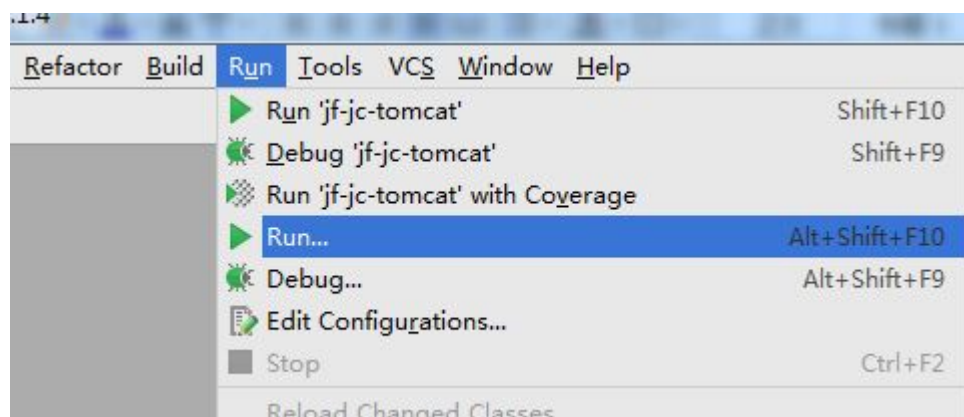


至此，tomcat 部分已经完成，按同样的方式配置 jetty 部分(也可以直接把 package 和 web.xml 文件直接拷贝过去)。

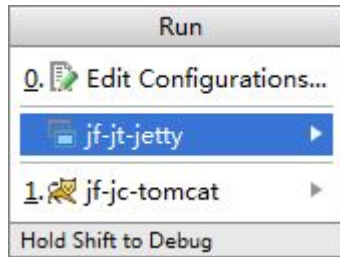
五、运行项目

1. 在 jetty 下运行

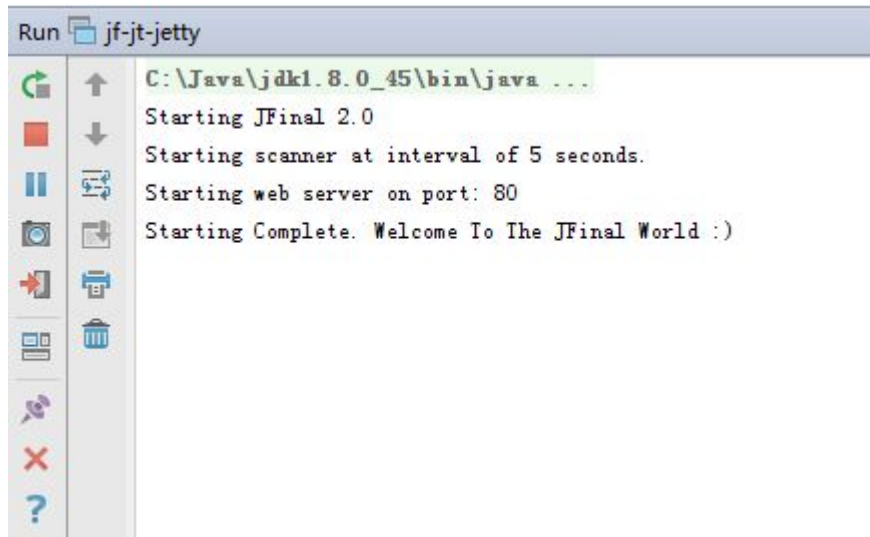
点击【Run】-【Run】菜单：



页面中间会出现让你选择运行哪个模块的选项，我们选择 jetty 的模块。

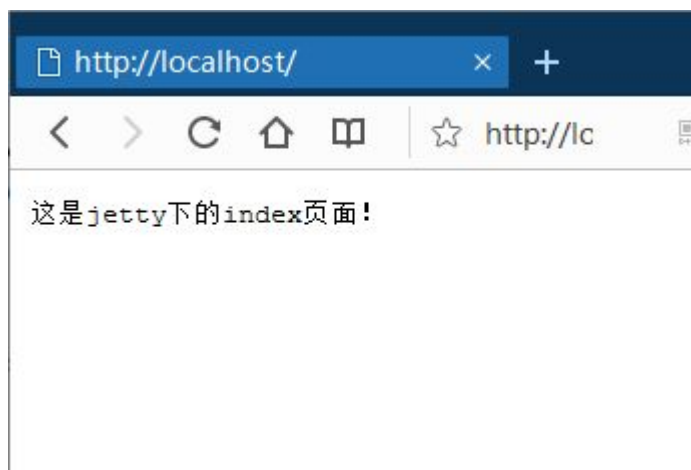


此时 IDEA 界面下半部会显示 jetty 的启动信息，如下：



说明 jetty 已经正常启动了。

此时需要手动打开浏览器页面，输入“http://localhost”，就会出现 Index 控制器对应的页面。(注意，如果你的电脑上装了别的 Web 服务器，注意端口冲突)。



IDEA 下部窗口会出现相应的调试信息。



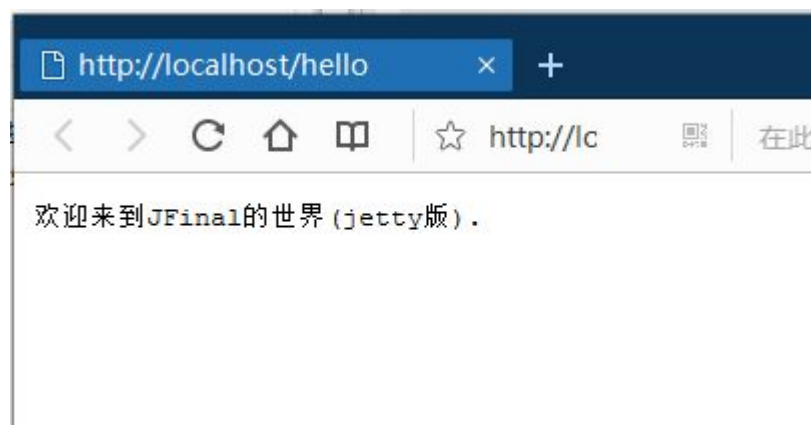
```
Run jf-jt-jetty
C:\Java\jdk1.8.0_45\bin\java ...
Starting JFinal 2.0
Starting scanner at interval of 5 seconds.
Starting web server on port: 80
Starting Complete. Welcome To The JFinal World :)
六月 28, 2015 11:20:19 下午 com.jfinal.core.ActionHandler warn
警告: 404 Action Not Found: /ifr_status/

JFinal action report ----- 2015-06-28 23:21:32 -----
Controller : demo.IndexController. (IndexController.java:1)
Method      : index

JFinal action report ----- 2015-06-28 23:22:29 -----
Controller : demo.HelloController. (HelloController.java:1)
Method      : index

JFinal action report ----- 2015-06-28 23:23:55 -----
Controller : demo.IndexController. (IndexController.java:1)
Method      : index
```

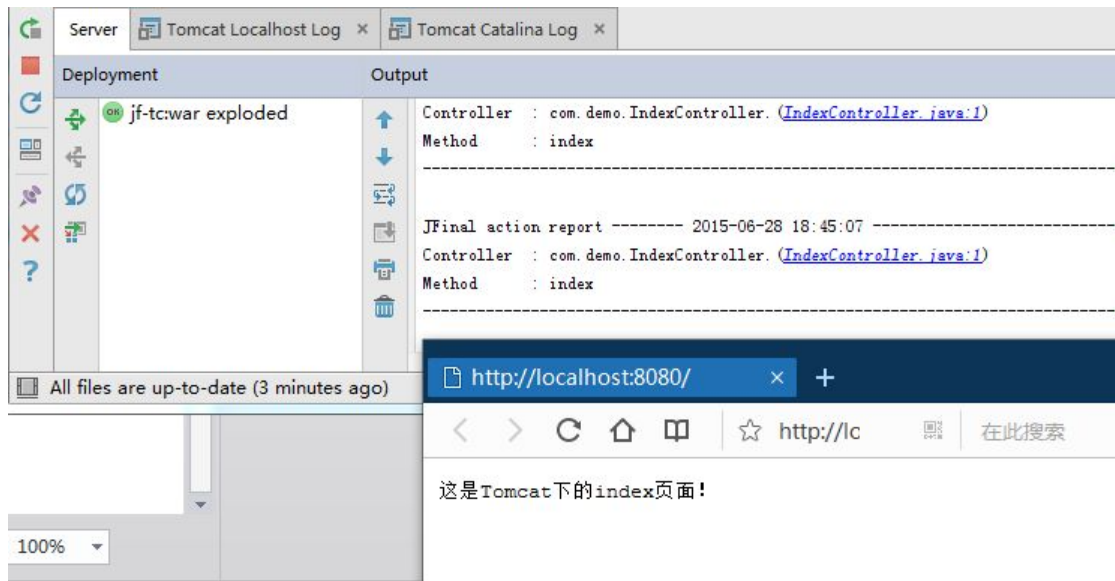
输入 `http://localhost/hello`,会出现 hello 控制器对应的页面。



2. 在 tomcat 下运行

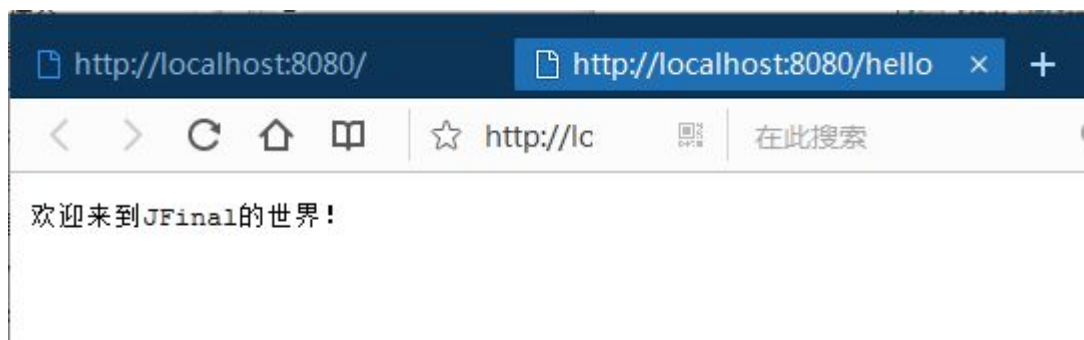
同样点击【Run】-【Run】菜单，在页面中心的弹出菜单选“jf-tc-tomcat”。

IDEA 界面下半部分会显示启动 tomcat 过程中的日志，等待 tomcat 启动后，会自动调用本地浏览器窗口，把 index 页面内容显示出来。



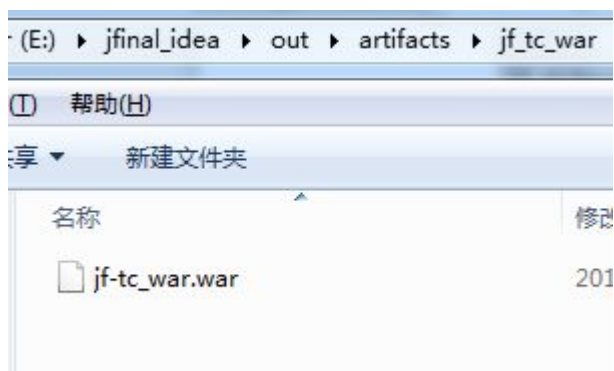
输入 `http://localhost:8080/hello`

出来 hello 控制器中的内容:



六、项目实际部署

IntelliJ IDEA 打包的 war 文件位于项目根目录的 out 目录下。



把此文件拷贝到 tomcat 的 webapps 目录下(可以改名,例如改名为 test.war), 无需做任何修改, 启动 tomcat 后, 直接访问: `http://ip:8080/test/` 即可正常访问页面。

要更改输出目录，请在【File】-【Project Structure】-【Project】标签页修改即可。

