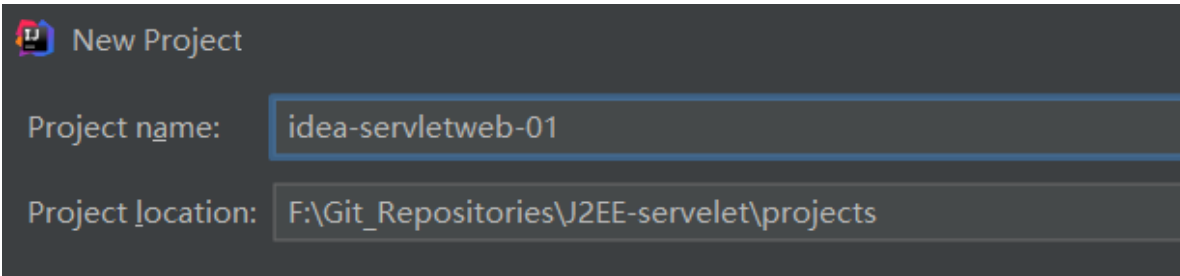
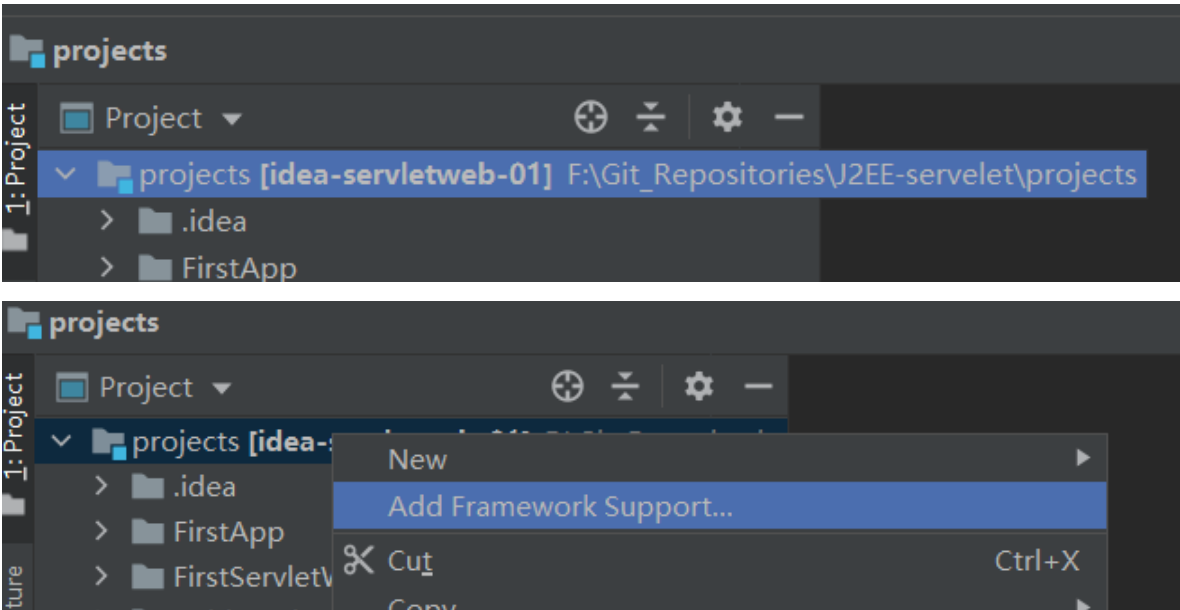


第一步，正常新建一个java项目，该项目在idea中的名字叫idea-servletweb-01，编写的代码存放在这个路径下F:\Git_Repositories\J2EE-servlet\projects，该路径下有着之前没用集成开发工具写的三个项目



第二步，在工程上点击鼠标右键，选择添加框架支持Add Framework Support，

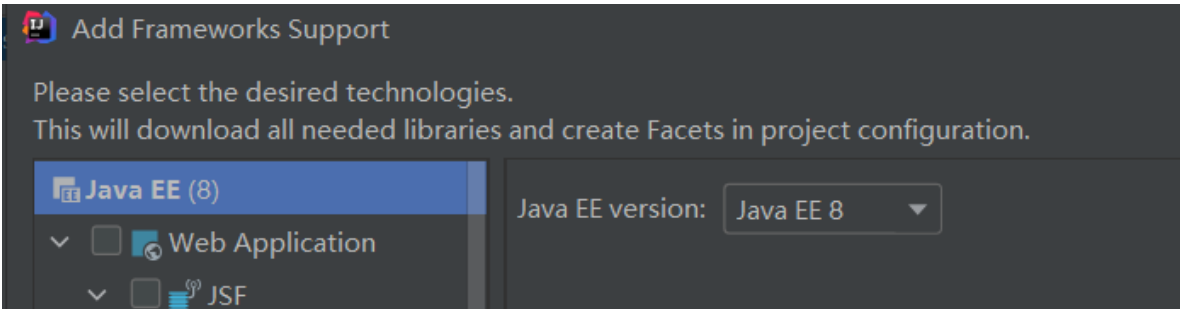


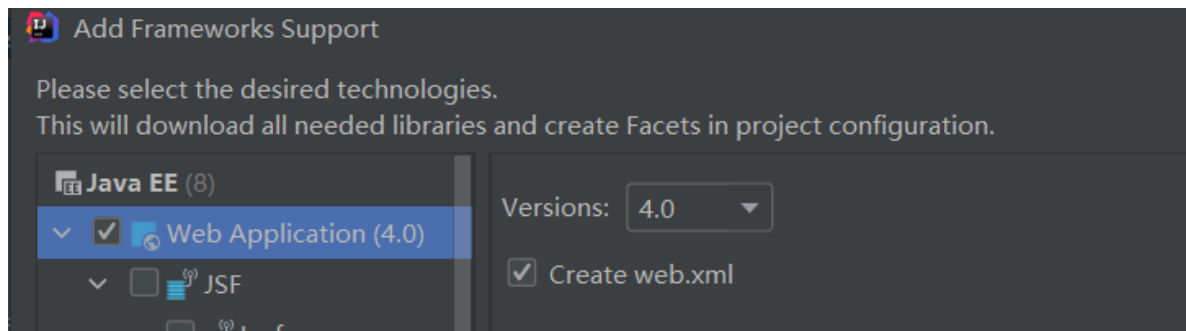
然后选择JavaEE的版本，具体选什么更具自己安装的Tomcat的版本选择，具体选择如下图

Tomcat版本

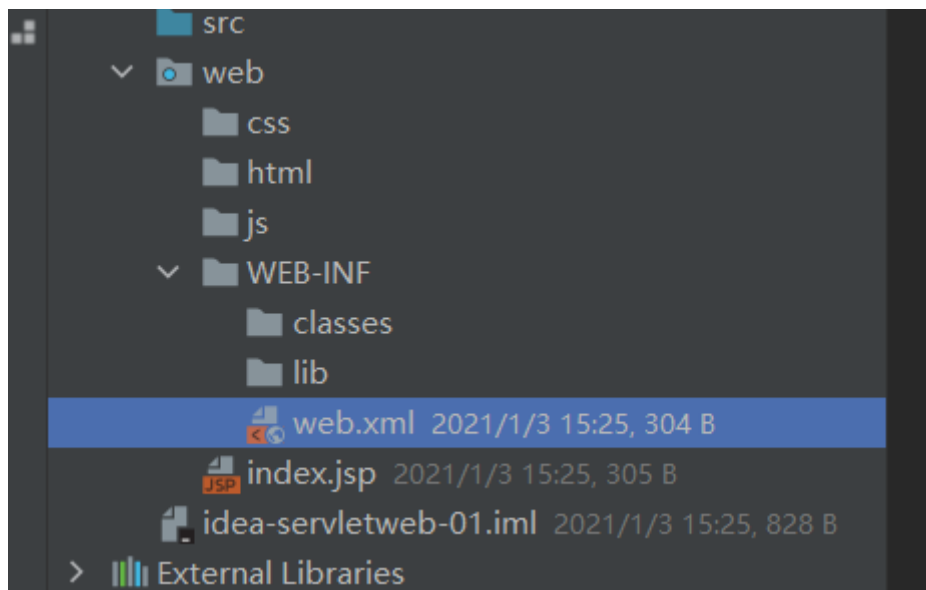
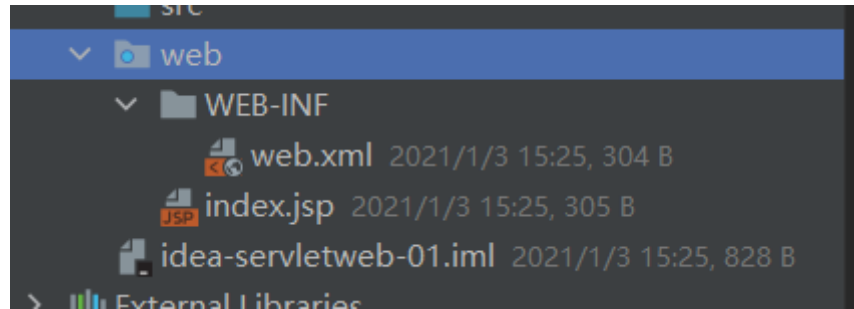
Tomcat	JDK	JavaEE	Servlet	JSP
9.x	8及更高的版本	JavaEE8	4.0	2.4?
8.x	7及更高的版本	JavaEE7	3.1	2.3
7.x	6及更高的版本	JavaEE6	3.0	2.2
6.x	5及更高的版本	JavaEE5	2.5	2.1
5.x	1.4及更高的版本	JavaEE1.4	2.4	2.0

javaEE8的子规范是Servlet4.0，我们所安装的Tomcat9实现了Servlet4.0，JDK是1.8，所以选择javaEE8，然后勾选WebApplication，默认创建web.xml。点击OK。

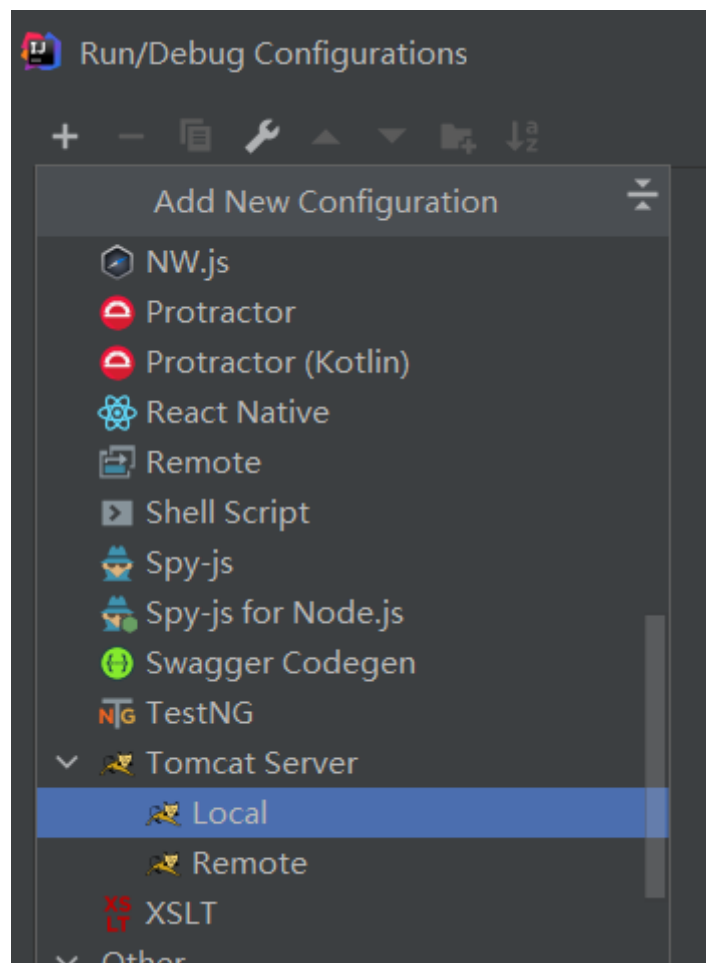
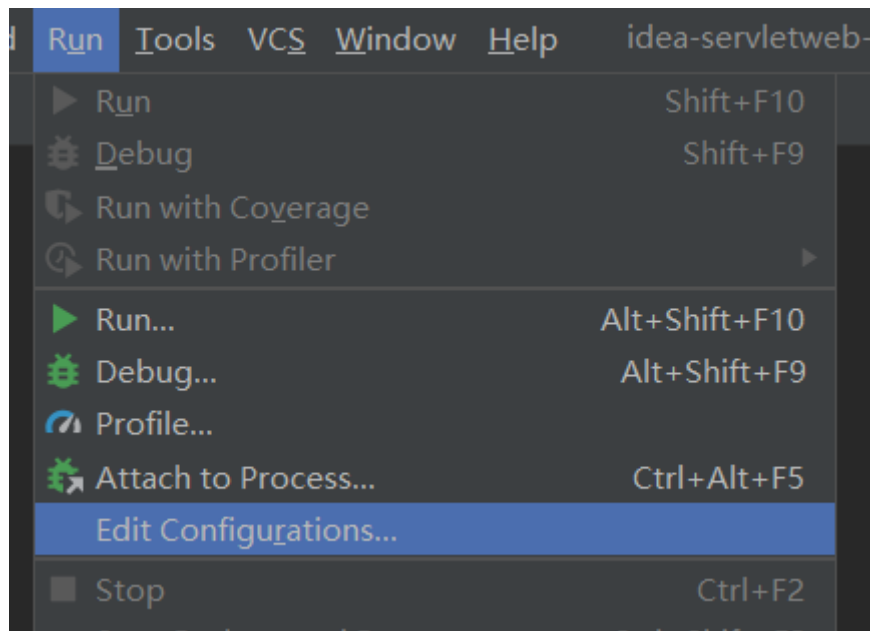




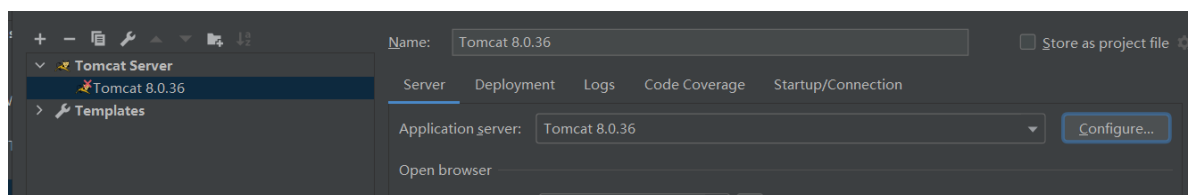
这样该项目就是一个web项目了。可以看到右边新生成了一个web，打开里面可以看到里面帮我们建好了WEB-INF文件夹，以及该目录下的一个web.xml文件。我们自己可以在web目录下新建目录如css、html和js，但这不是必须的。但是在WEB-INF目录下必须要建classes和lib目录，具体要不要手动创建，还是以后idea会自动生成还不确定，这里手动新建。

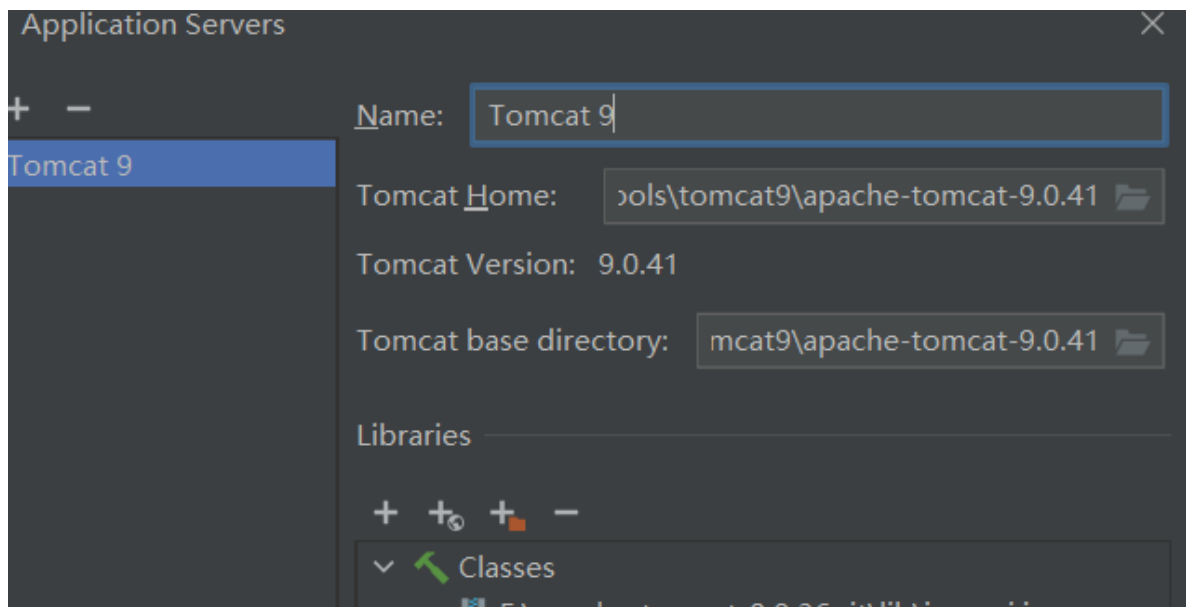


第三步，配置项目的Tomcat服务器。选择Run->Edit Configuration->左上角加号->Tomcat Server->Local

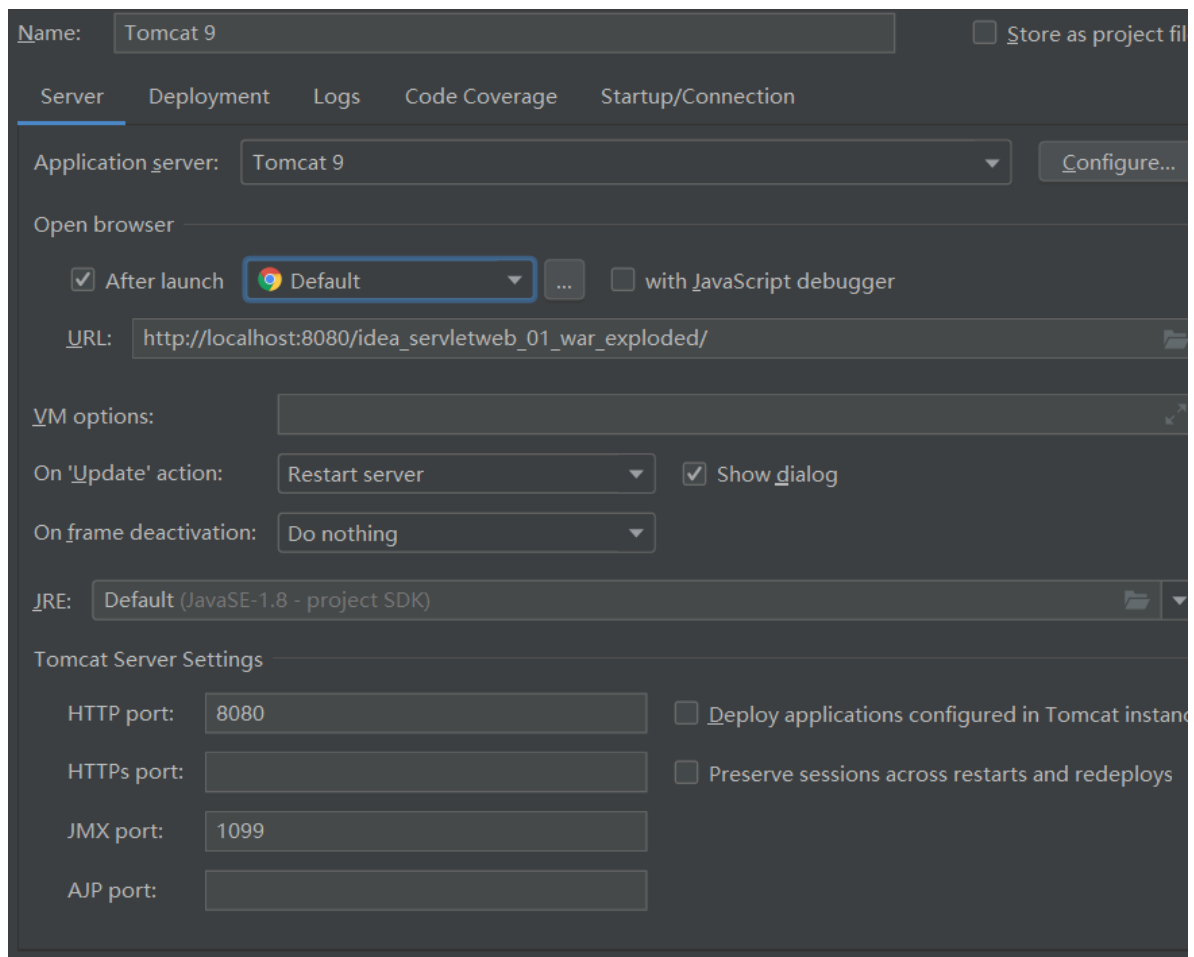


点开，找到右边的Configure，选择自己Tomcat的路径



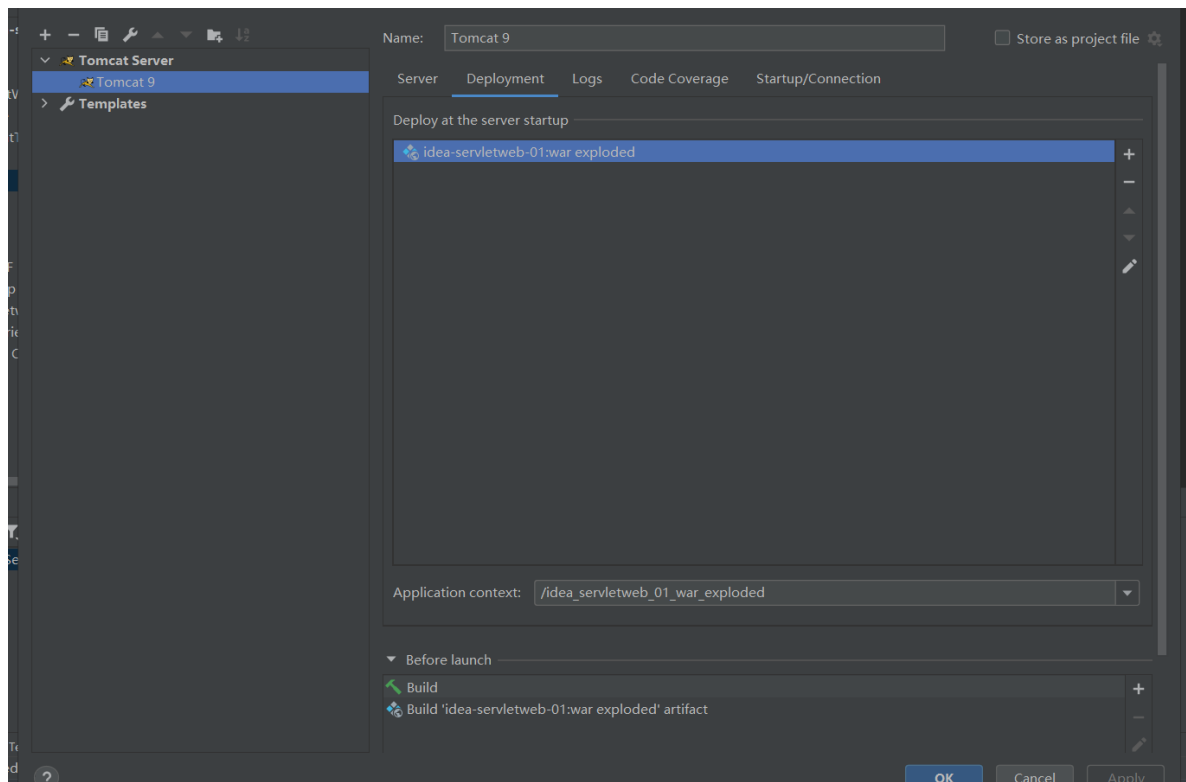


可以看到上面Tomcat Home跟我们之前配CATALINA_HOME一样，路径是找到Tomcat中的bin的上级目录就是Tomcat的家，Name那里可以自己命名，Tomcat 9。选好号点击OK。



该界面中可以设置项目调试的浏览器，不用管，以及端口号的设置。

第四步，回到Edit Configurations，选择第二个选项Deployment（部署），选中项目



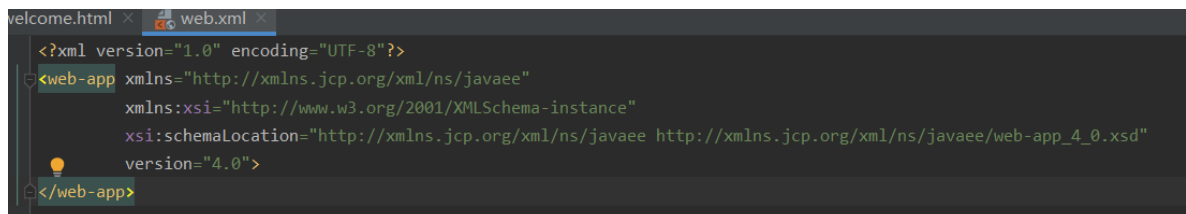
我们可以修改部署到Tomcat服务器中该项目的名字，可以看到Application context中默认是/idea_servletweb_01_war_exploded，我们可以修改为taobao，这就是该项目部署到Tomcat里webapps中的项目的名字，到时候浏览器中请求路径中的项目名就是taobao，而不是我们之前创建项目时的Project Name: idea-servletweb-01，这个名字只是在本项目在idea中的项目名。

示例：

我们先在html目录下新建一个welcome.html，在里面写一个超链接，连接到该项目的的一个Servlet实现类，

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>welcome page</title>
</head>
<body>
  <a href="/taobao/hello">hello</a>
</body>
</html>
```

然后点开web.xml文件，可以看到头部标签已经帮我们写好了，我们编写资源与请求的绑定代码



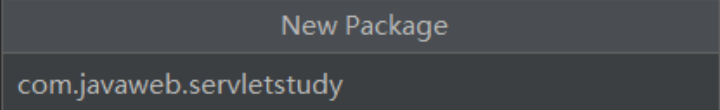
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
  version="4.0">
```

```
version="4.0">

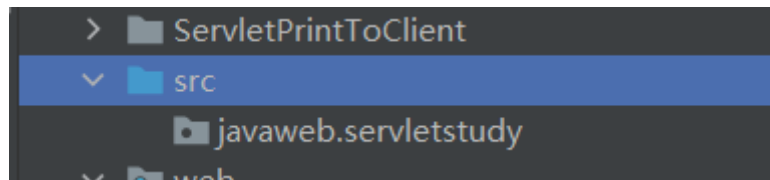
<servlet>
  <servlet-name>HelloServlet</servlet-name>
  <servlet-class>com.javaweb.servletstudy.HelloServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>HelloServlet</servlet-name>
  <url-pattern>/hello</url-pattern>
</servlet-mapping>
</web-app>
```

复制其中的包名，在src下创建包

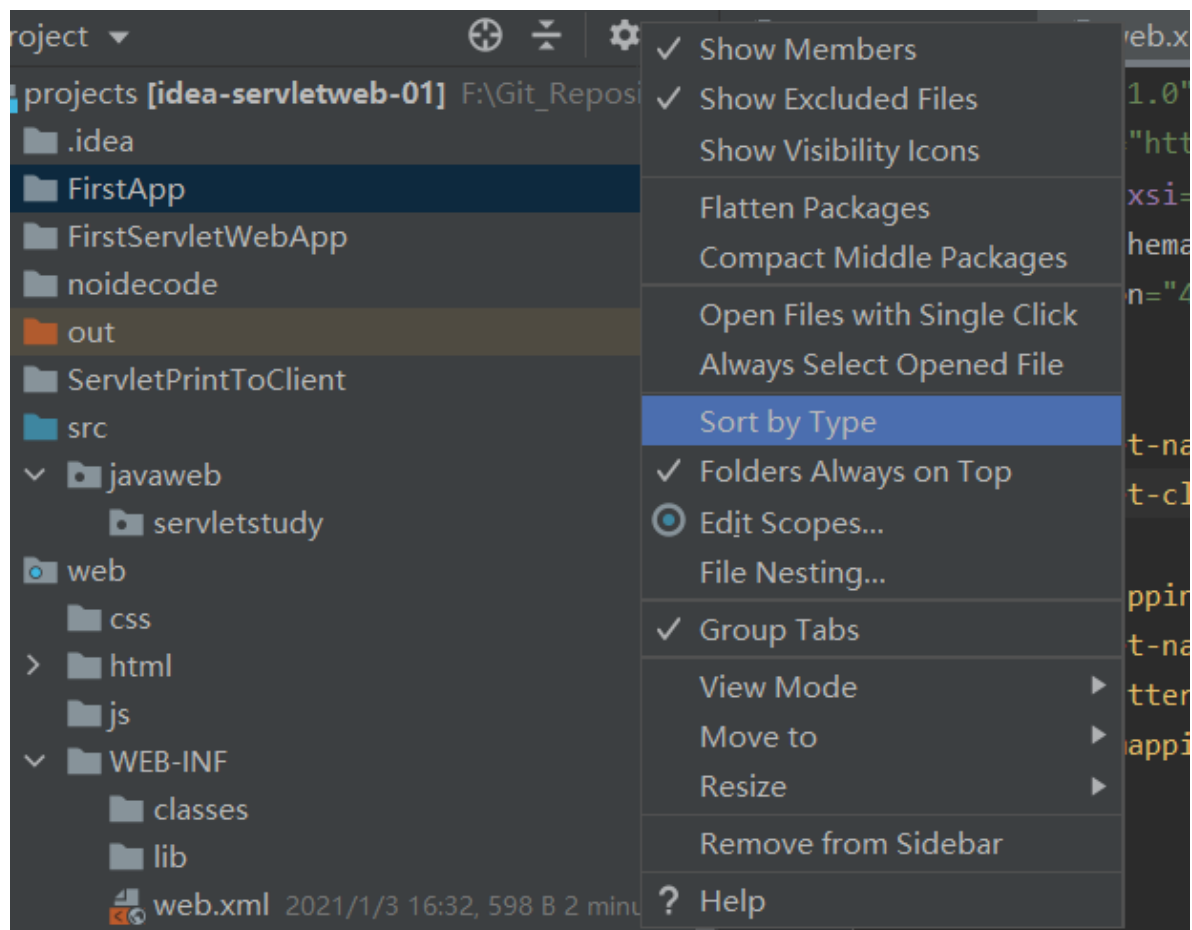
```
<servlet>
  <servlet-name>HelloServlet</servlet-name>
  <servlet-class>com.javaweb.servletstudy.HelloServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>HelloServlet</servlet-name>
  <url-pattern>/hello</url-pattern>
</servlet-mapping>
web-app>
```



发现创建的包是如此显示



点击上面的小齿轮，将Flatten Packages和Compact Middle Packages的勾去掉



复制类名，新建一个Servlet实现类HelloServlet

```
package javaweb.servletstudy;

import javax.servlet.*;
import java.io.IOException;
import java.io.PrintWriter;

public class HelloServlet implements Servlet {
    @Override
    public void init(ServletConfig servletConfig) throws ServletException {

    }

    @Override
    public ServletConfig getServletConfig() {
        return null;
    }

    @Override
    public void service(ServletRequest servletRequest, ServletResponse
servletResponse) throws ServletException, IOException {

        servletResponse.setContentType("text/html;charset=UTF-8");
        PrintWriter out = servletResponse.getWriter();
        out.print("<html>");
        out.print("<head>");
        out.print("<title>hello servlet</title>");
        out.print("<body>");
        out.print("我正在学习中.....");
        out.print("</body>");
    }
}
```

```

        out.print("</head>");
        out.print("</html>");

    }

    @Override
    public String getServletInfo() {
        return null;
    }

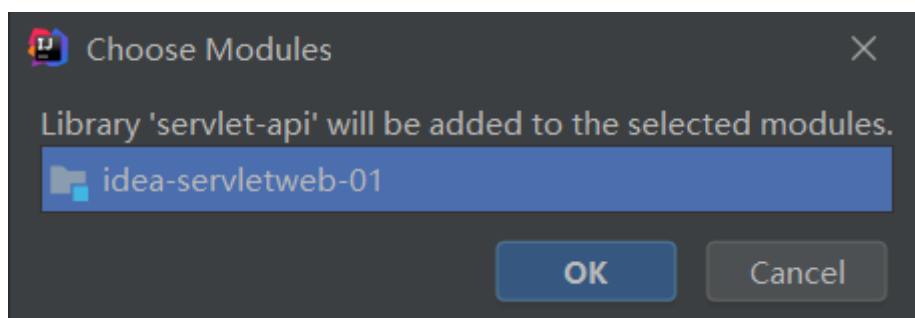
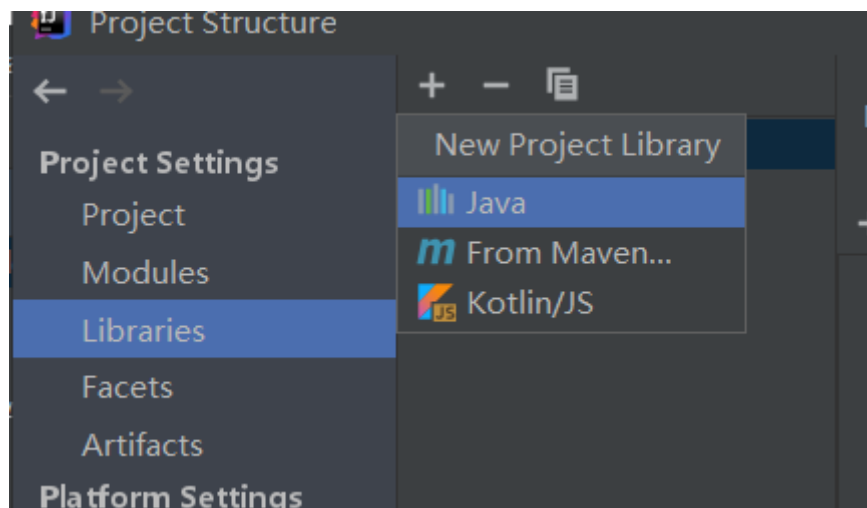
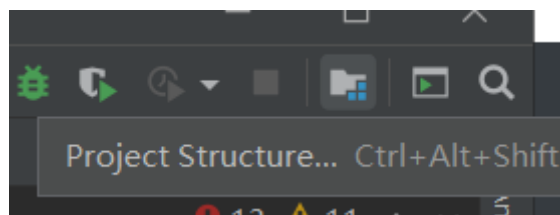
    @Override
    public void destroy() {

    }

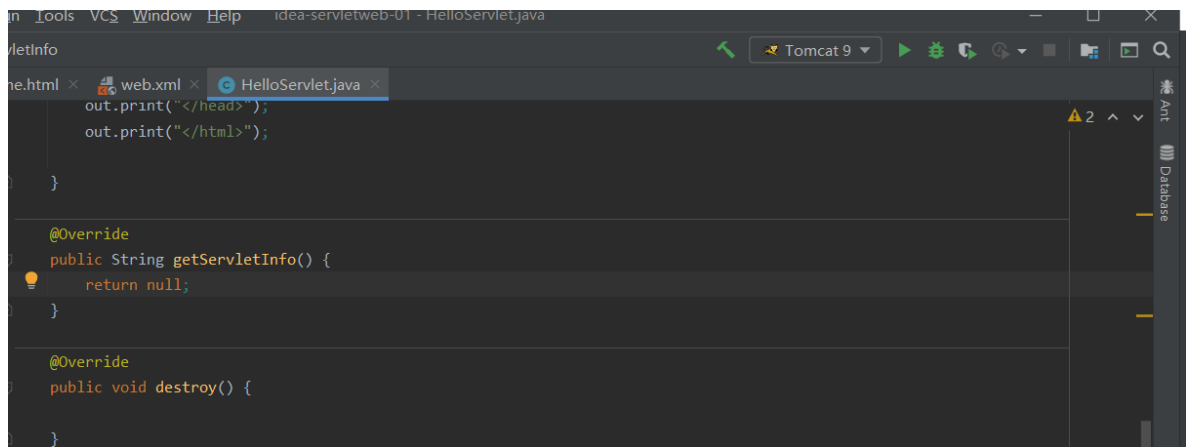
}

```

还有一点需要注意，idea工具不提供JavaEE Libraries全套规范，我们要实现Servlet接口，需要导入Tomcat的lib目录中的servlet-api包。先点击右上角的文件图标，Project Structure。然后点击Libraries，点击加号，点击java，找到要导入的包，选中要加到哪个项目模块，点击OK，点击Apply应用，点击OK。完成



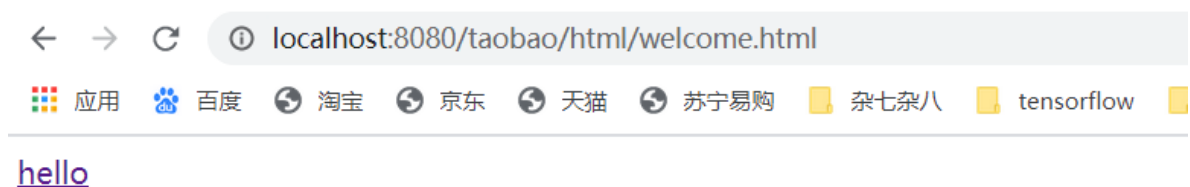
编写号代码后，在右上角选择Tomcat 9，然后运行



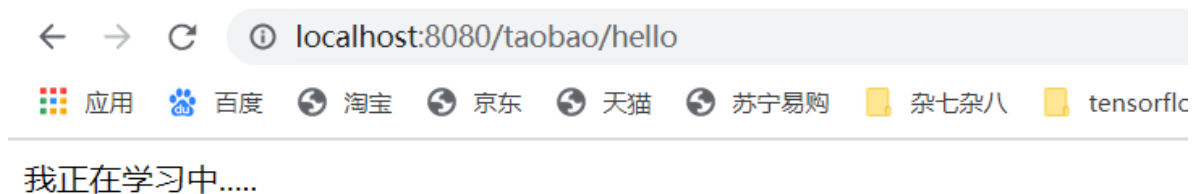
运行后就会调用之前设置的浏览器和默认的请求<http://localhost:8080/taobao/>



然后输入<http://localhost:8080/taobao/html/welcome.html>



点击hello, 调转



我们可以看到Tomcat服务器启动的一些乱码问题, 之后会解决

