

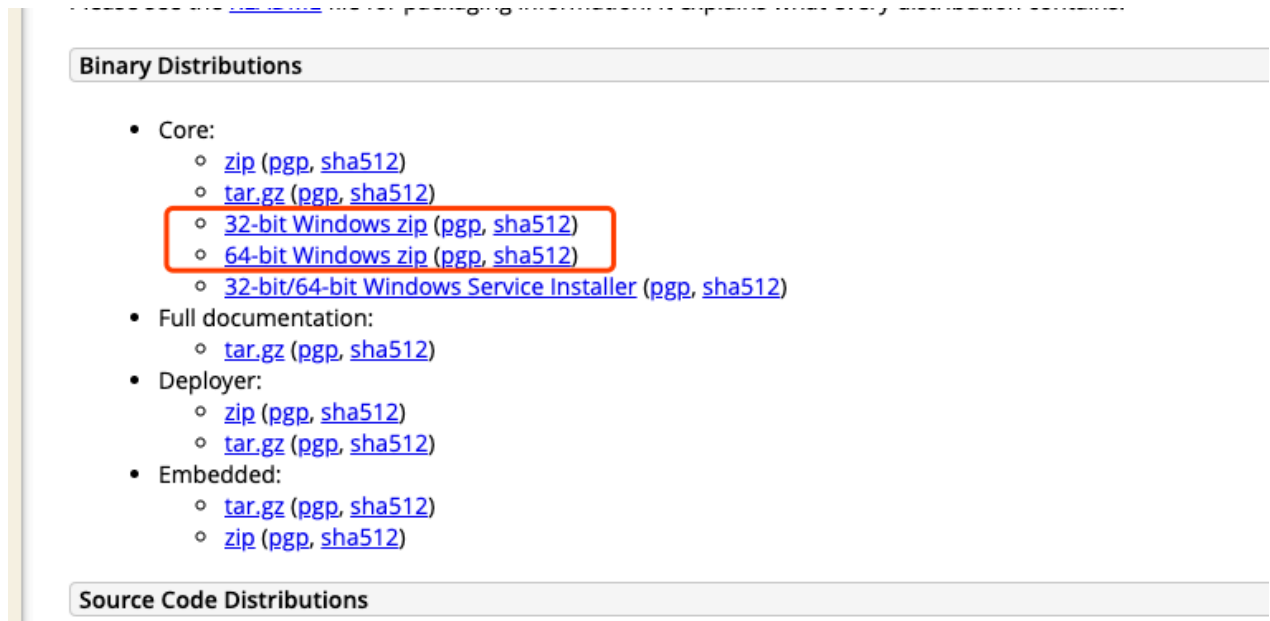
Tomcat

Web 应用服务器：Tomcat、Jboss、Weblogic、Jetty

- 安装 Tomcat

1、官网下载压缩文件。

<https://tomcat.apache.org/download-90.cgi>



2、解压缩。

bin：存放各个平台下启动和停止 Tomcat 服务的脚本文件。

conf：存放各种 Tomcat 服务器的配置文件。

lib：存放 Tomcat 服务器所需要的 jar。

logs：存放 Tomcat 服务运行的日志。

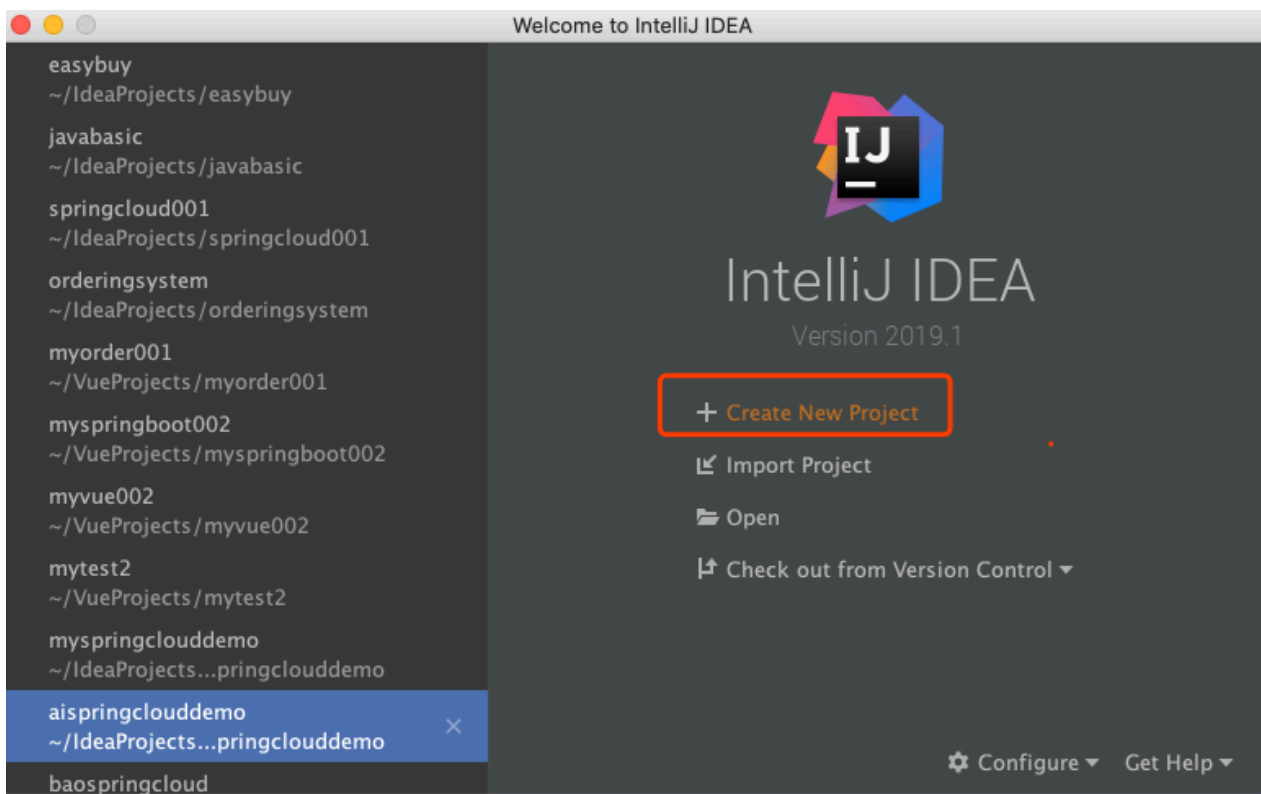
temp：Tomcat 运行时的临时文件。

webapps：存放允许客户端访问的资源（Java 程序）。

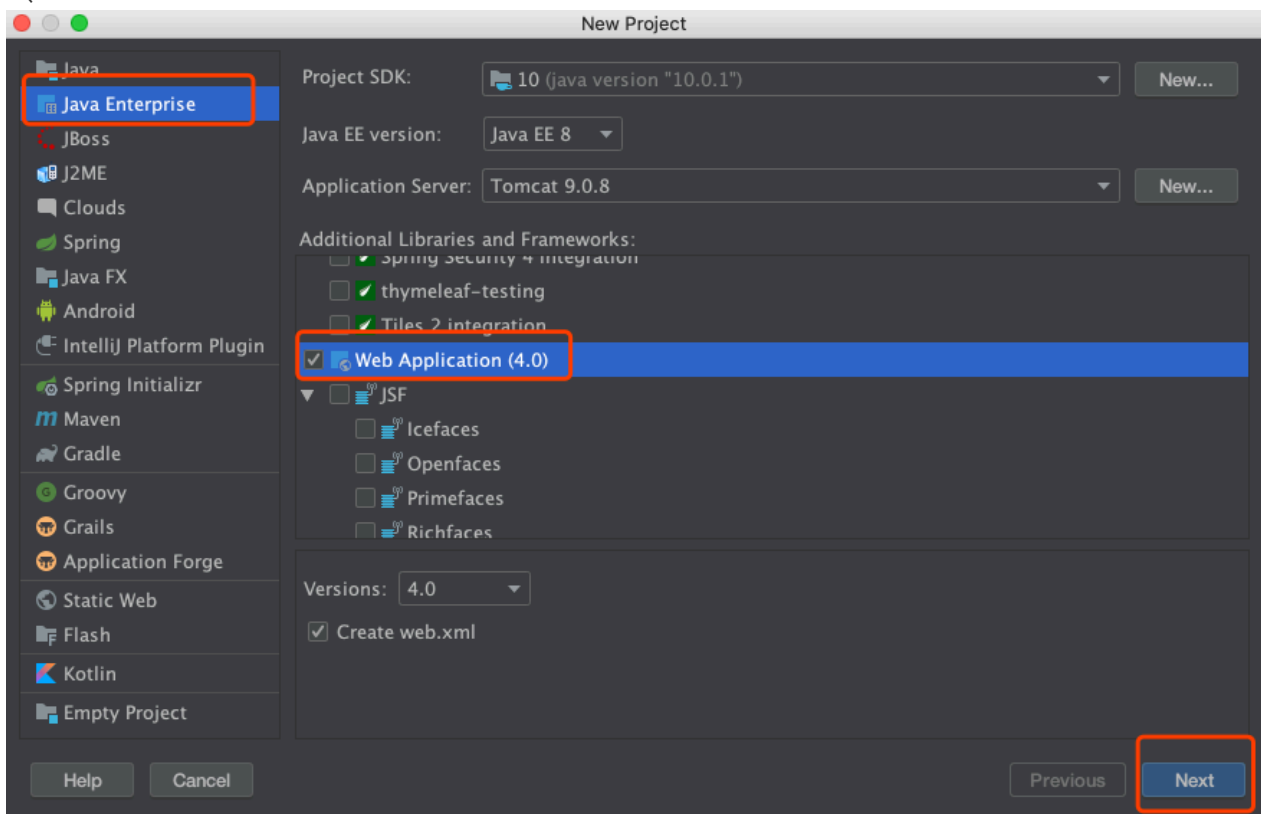
work：存放 Tomcat 将 JSP 转换之后的 Servlet 文件。

IDEA 集成 Tomcat

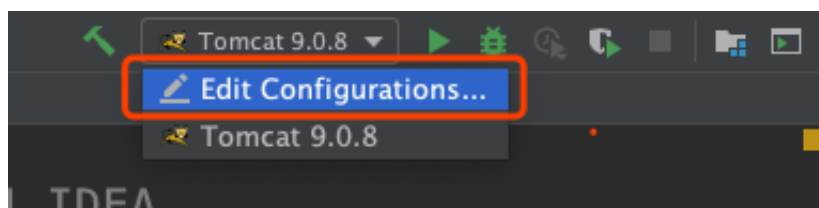
1、创建 Java Web 工程。

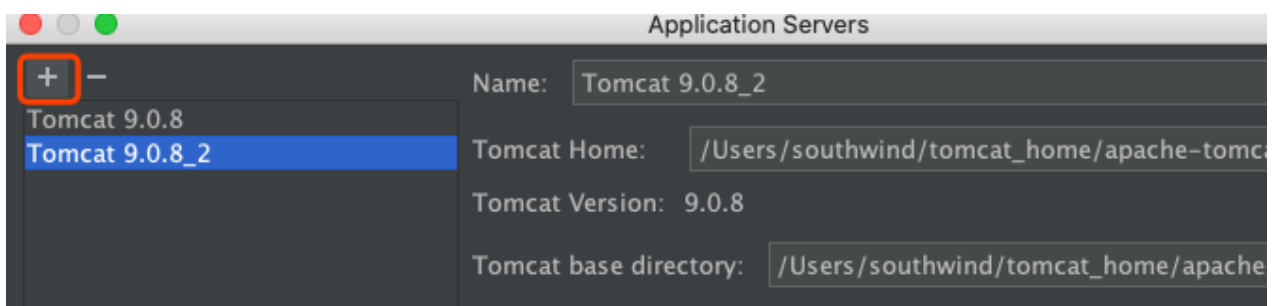
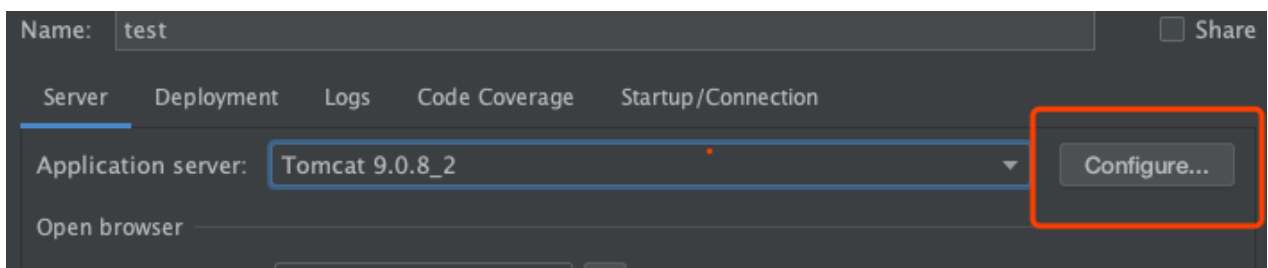
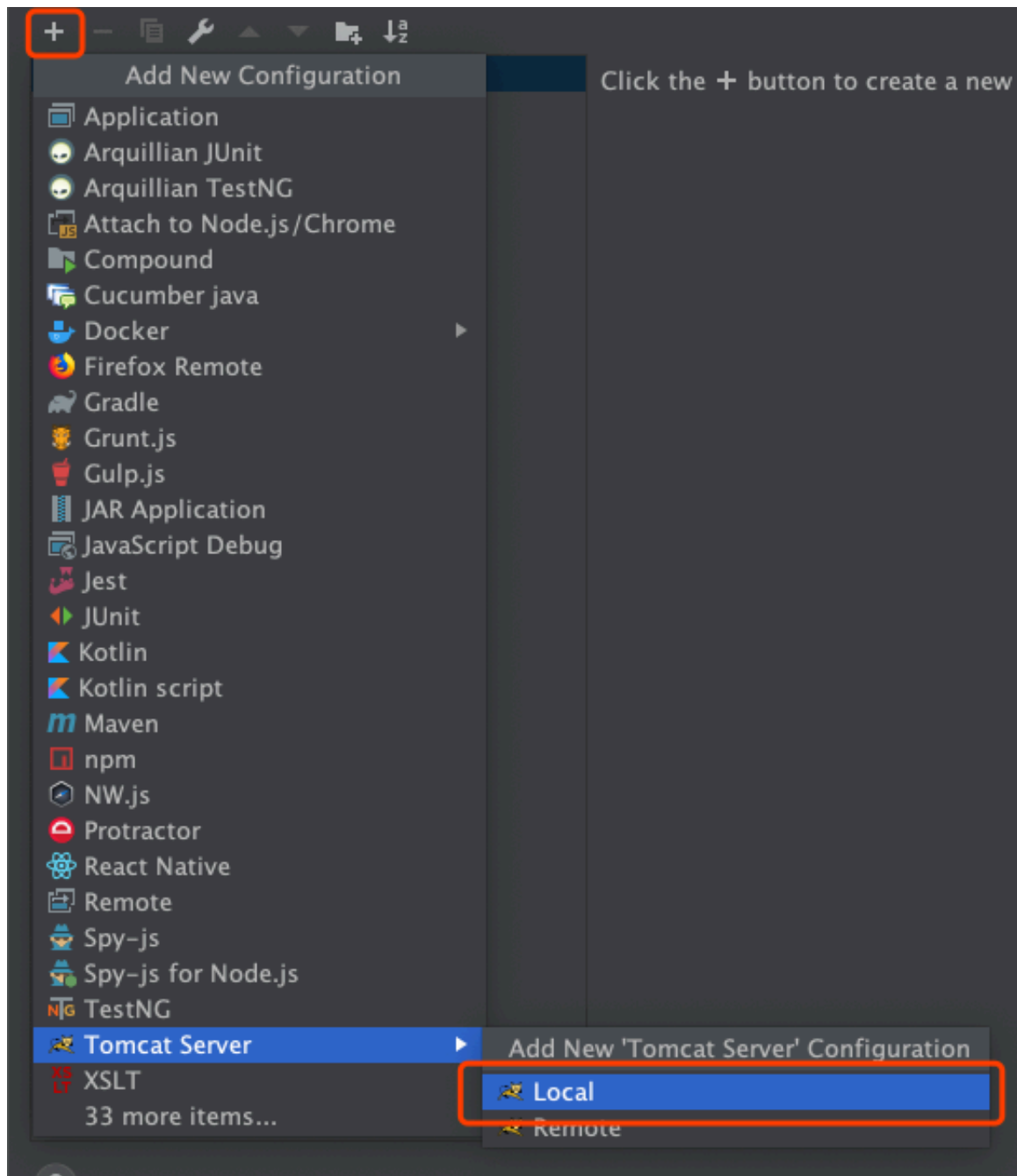


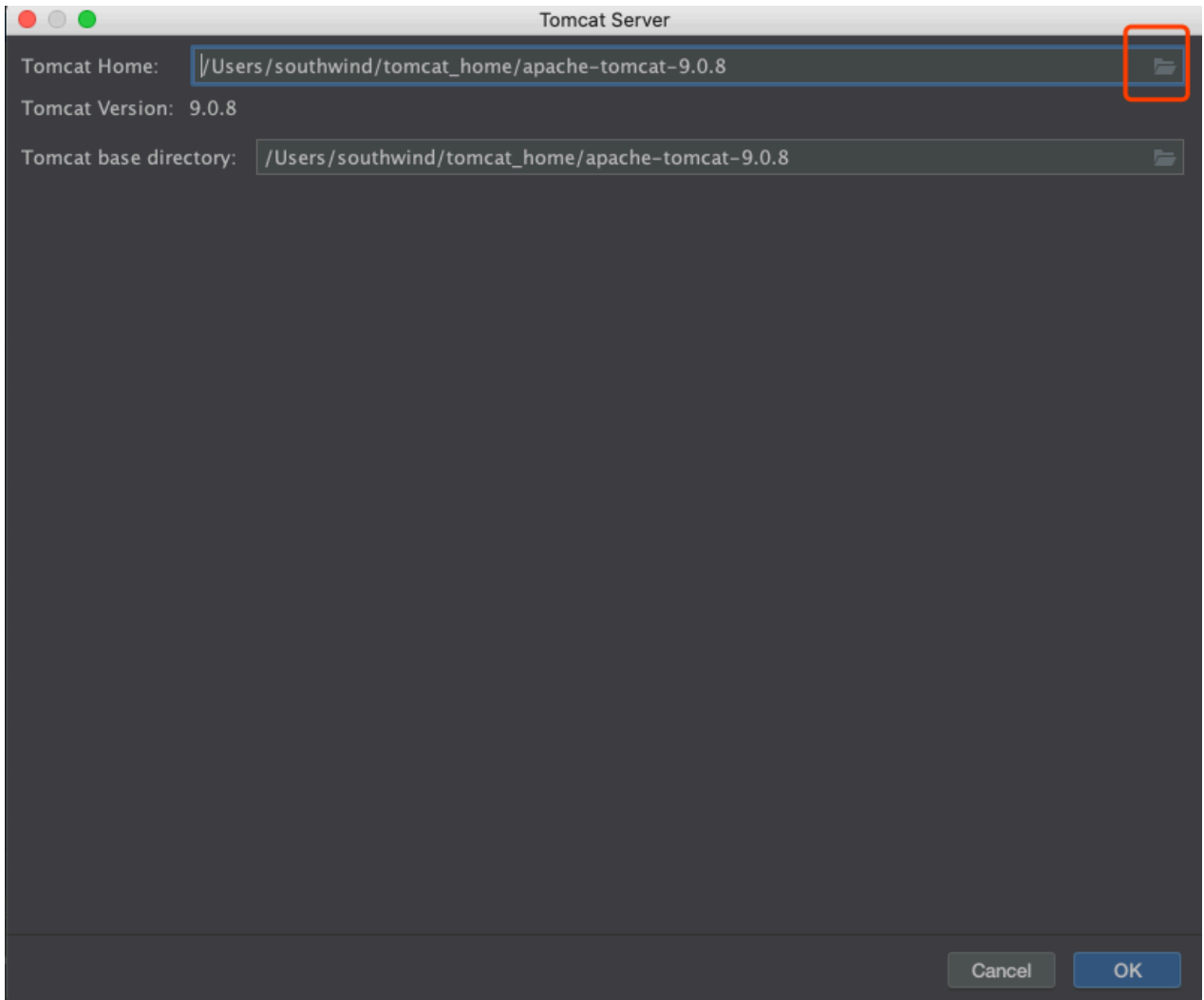
2、

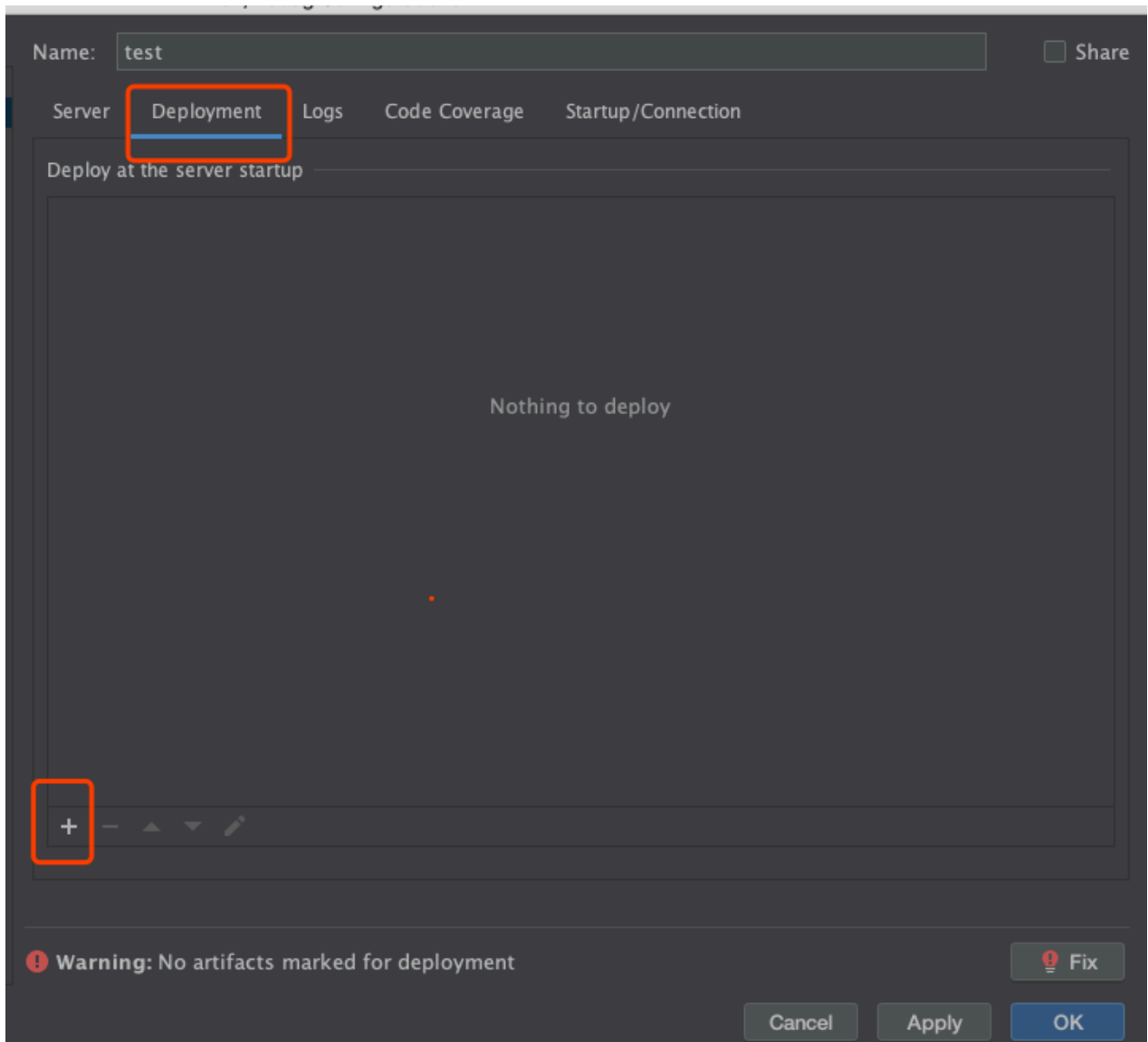


3、IDEA 中配置 Tomcat。









Servlet

- 什么是 Servlet?

Servlet 是 Java Web 开发的基石，与平台无关的服务器组件，它是运行在 Servlet 容器/Web 应用服务器/Tomcat，负责与客户端进行通信。

Servlet 的功能：

- 1、创建并返回基于客户请求的动态 HTML 页面。
- 2、与数据库进行通信。

- 如何使用 Servlet?

Servlet 本身是一组接口，自定义一个类，并且实现 Servlet 接口，这个类就具备了接受客户端请求以及做出响应的功能。

```
package com.southwind.servlet;
```

```

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

public class MyServlet 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 {
        String id = servletRequest.getParameter("id");
        System.out.println("我是Servlet, 我已经接收到了客户端发来的请求, 参数是"+id);
        servletResponse.setContentType("text/html;charset=UTF-8");
        servletResponse.getWriter().write("客户端你好, 我已接收到你的请求");
    }

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

    @Override
    public void destroy() {

    }
}

```

浏览器不能直接访问 Servlet 文件，只能通过映射的方式来间接访问 Servlet，映射需要开发者手动配置，有两种配置方式。

- 基于 XML 文件的配置方式。

```

<servlet>
    <servlet-name>hello</servlet-name>
    <servlet-class>com.southwind.servlet.HelloServlet</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>hello</servlet-name>
    <url-pattern>/demo2</url-pattern>
</servlet-mapping>

```

- 基于注解的方式。

```
@WebServlet("/demo2")
public class HelloServlet implements Servlet {

}
```

上述两种配置方式结果完全一致，将 demo2 与 HelloServlet 进行映射，即在浏览器地址栏中直接访问 demo 就可以映射到 HelloServlet。

Servlet 的生命周期

- 1、当浏览器访问 Servlet 的时候，Tomcat 会查询当前 Servlet 的实例化对象是否存在，如果不存在，则通过反射机制动态创建对象，如果存在，直接执行第 3 步。
- 2、调用 init 方法完成初始化操作。
- 3、调用 service 方法完成业务逻辑操作。
- 4、关闭 Tomcat 时，会调用 destory 方法，释放当前对象所占用的资源。

Servlet 的生命周期方法：无参构造函数、init、service、destory

- 1、无参构造函数只调用一次，创建对象。
- 2、init 只调用一次，初始化对象。
- 3、service 调用 N 次，执行业务方法。
- 4、destory 只调用一次，卸载对象。

ServletConfig

该接口是用来描述 Servlet 的基本信息的。

getServletName() 返回 Servlet 的名称，全类名(带着包名的类名)

getInitParameter(String key) 获取 init 参数的值 (web.xml)

getInitParameterNames() 返回所有的 initParamter 的 name 值，一般用作遍历初始化参数

getServletContext() 返回 ServletContext 对象，它是 Servlet 的上下文，整个 Servlet 的管理者。

ServletConfig 和 ServletContext 的区别：

ServletConfig 作用于某个 Servlet 实例，每个 Servlet 都有对应的 ServletConfig，ServletContext 作用于整个 Web 应用，一个 Web 应用对应一个 ServletContext，多个 Servlet 实例对应一个 ServletContext。

一个是局部对象，一个是全局对象。

