

保姆級tomcat 快速入門

簡說Java 前天

以下文章來源於Java後端技術全棧，作者田維常



Java後端技術全棧

專注Java技術乾貨：JVM、Spring、Spring Boot、Cloud、IDEA、Dubbo、Redis...



點擊上方“簡說Java”，關注並“星標”

帶你從零學Java

本文目录

背景介绍

安装tomcat ⊕

启动tomcat ⊕

Servlet项目部署到tomcat中

IDEA中项目部署到tomcat中 ⊕

总结

作者：田維常

來源：Java後端技術全棧

這是簡說Java推薦的第13篇文章



背景介紹

Apache Tomcat 是Java Servlet、JavaServer Pages（JSP）、Java表達式語言和Java的WebSocket技術的一個開源實現,通常我們將Tomcat稱為Web容器或者Servlet容器。

tomcat各版本和相應規範的映射關係：

Servlet規格	JSP規範	EL規格	WebSocket規範	認證 (JASIC) 規範	Apache Tomcat版本	最新發行版本	支持的Java版本
5.0	3.0	4.0	2.0	2.0	10.0.x	10.0.0 (測試版)	8及更高版本
4.0	2.3	3.0	1.1	1.1	9.0.x	9.0.41	8及更高版本
3.1	2.3	3.0	1.1	1.1	8.5.x	8.5.61	7及更高版本
3.1	2.3	3.0	1.1	不適用	8.0.x (已取代)	8.0.53 (已取代)	7及更高版本
3.0	2.2	2.2	1.1	不適用	7.0.x	7.0.107	6和更高版本 (WebSocket為7和更高版本)
2.5	2.1	2.1	不適用	不適用	6.0.x (已歸檔)	6.0.53 (已歸檔)	5及更高版本
2.4	2.0	不適用	不適用	不適用	5.5.x (已存檔)	5.5.36 (存檔)	1.4及更高版本
2.3	1.2	不適用	不適用	不適用	4.1.x (已歸檔)	4.1.40 (已歸檔)	1.3及更高版本
2.2	1.1	不適用	不適用	不適用	3.3.x (已存檔)	3.3.2 (已存檔)	1.1及更高版本

下載地址

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

下載到本地並解壓：



進入主目錄：

名称	修改日期	类型	大小
bin	2020/12/3 11:46	文件夹	
conf	2020/12/3 11:45	文件夹	
lib	2020/12/3 11:45	文件夹	
logs	2020/12/3 11:43	文件夹	
temp	2020/12/3 11:45	文件夹	
webapps	2020/12/3 11:45	文件夹	
work	2020/12/3 11:43	文件夹	
BUILDING.txt	2020/12/3 11:45	文本文档	20 KB
CONTRIBUTING.md	2020/12/3 11:45	Markdown File	6 KB
LICENSE	2020/12/3 11:45	文件	57 KB
NOTICE	2020/12/3 11:45	文件	3 KB
README.md	2020/12/3 11:45	Markdown File	4 KB
RELEASE-NOTES	2020/12/3 11:45	文件	7 KB
RUNNING.txt	2020/12/3 11:45	文本文档	17 KB

tomcat目錄介紹

<> 上午

啟動，關閉和其他腳本。這些.sh文件（對於Unix系統）是這些.bat文件的功能副本（對於Windows系統）。由於Win32命令行缺少某些功能，因此此處包含一些其他文件。

比如說：windows下啟動tomcat用的是startup.bat，另外Linux環境中使用的是startup.sh。對應還有相應的shutdown關閉腳本。

</> conf

tomcat的配置文件和相關的DTD。這裡最重要的文件是server.xml。它是容器的主要配置文件。

`catalina.policy`：tomcat：安全策略文件，控制JVM相關權限，具體可以參考java.security.Permission。

`catalina.properties`：tomcat Catalina 行為控制配置文件，比如：Common ClassLoader。

`logging.properties`：tomcat日誌配置文件。裡面的日誌採用的是JDK Logging。

`server.xml`：tomcat server配置文件(對於我開發人員來說是非常重要的)。

`context.xml`：全局context配置文件，監視並加載資源文件，當監視的文件發生髮生變化時，自動加載。

`tomcat-user.xml`：tomcat角色配置文件。

`web.xml`：Servlet標準的web.xml部署文件，tomcat默認實現部分配置入內：

- org.apache.catalina.servlets.DefaultServlet。
- org.apache.jasper.servlet.JspServlet

</> 日誌

日誌文件默認位於此處。

`localhost` 有用，當你們的tomcat啟動不了的時候，多看這個文件。比如：

- `NoClassDefFoundError`
- `ClassNotFoundException`

`access` 最沒用。

`catalina.{date}` 主要是控制台輸出，全部日誌都在這裡面。



這是您的webapp所在的位置。其實這裡面這幾個都是一個項目。

簡化web部署的方式。在線上環境中我們的應用是不會放在這裡的。最好的辦法就是外置。

</> LIB

tomcat存放共用的類庫。比如：

- `ecj-4.17.jar`: eclipse Java編譯器
- `jasper.jar` : JSP編譯器。

</> 工作

存放tomcat運行時編譯後的文件，比如JSP編譯後的文件。

</> 溫度

存放運行時產生的臨時文件。

啟動tomcat

啟動tomcat

我們在windows下的就直接啟動bin目錄下的startup.bat，對應Linux環境中我們使用的是startup.sh。


雙擊就能啟動了。控制台會輸8080端口，然後我們訪問：

<http://localhost:8080/>

頁面展示：

[Home](#) [Documentation](#) [Configuration](#) [Examples](#) [Wiki](#) [Mailing Lists](#) [Find Help](#)

Apache Tomcat/9.0.41



If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:

- [Security Considerations How-To](#)
- [Manager Application How-To](#)
- [Clustering/Session Replication How-To](#)

[Server Status](#)

[Manager App](#)

[Host Manager](#)

Developer Quick Start

[Tomcat Setup](#)
[First Web Application](#)

[Realms & AAA](#)
[JDBC Data Sources](#)

[Examples](#)

[Servlet Specifications](#)
[Tomcat Versions](#)

Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 9.0 access to the manager application is split between different users.
[Read more...](#)

[Release Notes](#)

[Changelog](#)

[Migration Guide](#)

[Security Notices](#)

Documentation

[Tomcat 9.0 Documentation](#)

[Tomcat 9.0 Configuration](#)

[Tomcat Wiki](#)

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

- [Tomcat 9.0 Bug Database](#)
- [Tomcat 9.0 JavaDocs](#)
- [Tomcat 9.0 Git Repository at GitHub](#)

Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

[tomcat-announce](#)
Important announcements, releases, security vulnerability notifications. (Low volume).

[tomcat-users](#)
User support and discussion

[taglibs-user](#)
User support and discussion for [Apache Taglibs](#)

[tomcat-dev](#)
Development mailing list, including commit messages

Other Downloads

- [Tomcat Connectors](#)
- [Tomcat Native](#)
- [Taglibs](#)
- [Deployer](#)

Other Documentation

- [Tomcat Connectors](#)
- [mod_jk Documentation](#)
- [Tomcat Native](#)
- [Deployer](#)

Get Involved

- [Overview](#)
- [Source Repositories](#)
- [Mailing Lists](#)
- [Wiki](#)

Miscellaneous

- [Contact](#)
- [Legal](#)
- [Sponsorship](#)
- [Thanks](#)

Apache Software Foundation

- [Who We Are](#)
- [Heritage](#)
- [Apache Home](#)
- [Resources](#)

這就代表著我們的tomcat啟動成功了。

此時， <http://localhost:8080/> 請求到的是ROOT目錄。

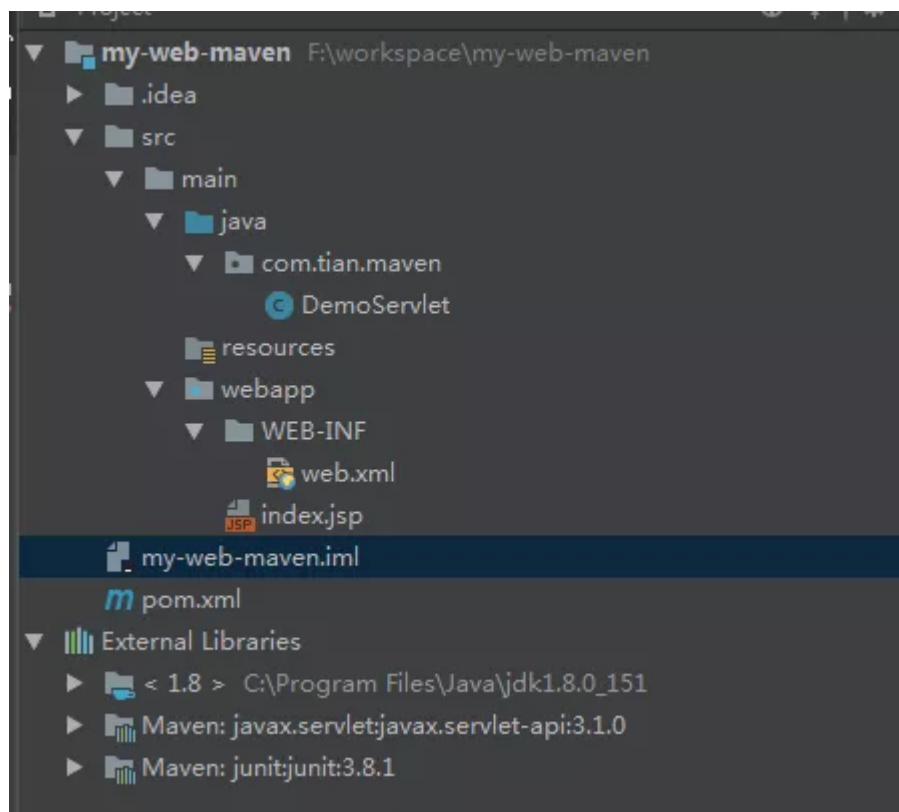
比如：我們還可以 <http://localhost:8080/manager>

Servlet項目部署到tomcat中

<> 創建web項目

使用maven創建web項目，既然說tomcat是Servlet容器，那麼在項目中創建一個Servlet類，然後打成war包，拷貝到tomcat中部署。

項目結構如下：



添加依賴


```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.tian.maven</groupId>
  <artifactId>my-web-maven</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>my-web-maven Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>3.1.0</version>
    </dependency>
  </dependencies>
  <build>
    <finalName>my-web-maven</finalName>
  </build>
</project>
```

創建DemoServlet

```
package com.tian.maven;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

@WebServlet(urlPatterns = "/demo")
public class DemoServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException {

        String msg = req.getParameter("message");
        String contextPath = req.getServletContext().getContextPath();
        System.out.println("contextPath=" + contextPath);
        resp.getWriter().println(msg);
    }
}
```

web.xml什麼都沒有，僅僅就是為了打包。

```
<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd" >
```

```
<web-app>
  <display-name>Archetype Created Web Application</display-name>
</web-app>
```

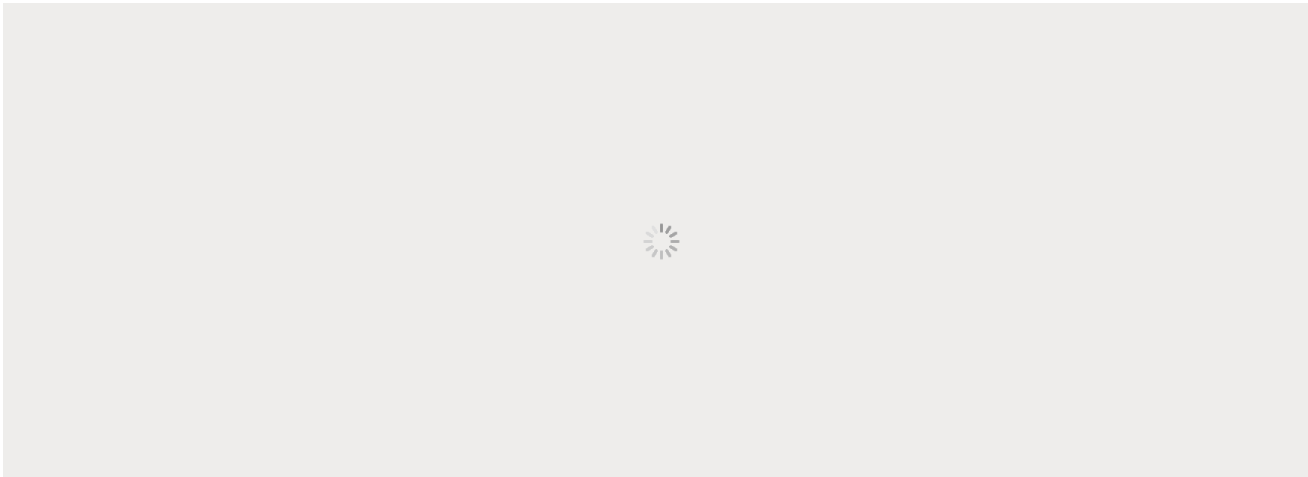
index.jsp找那個也沒什麼內容：

```
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
```

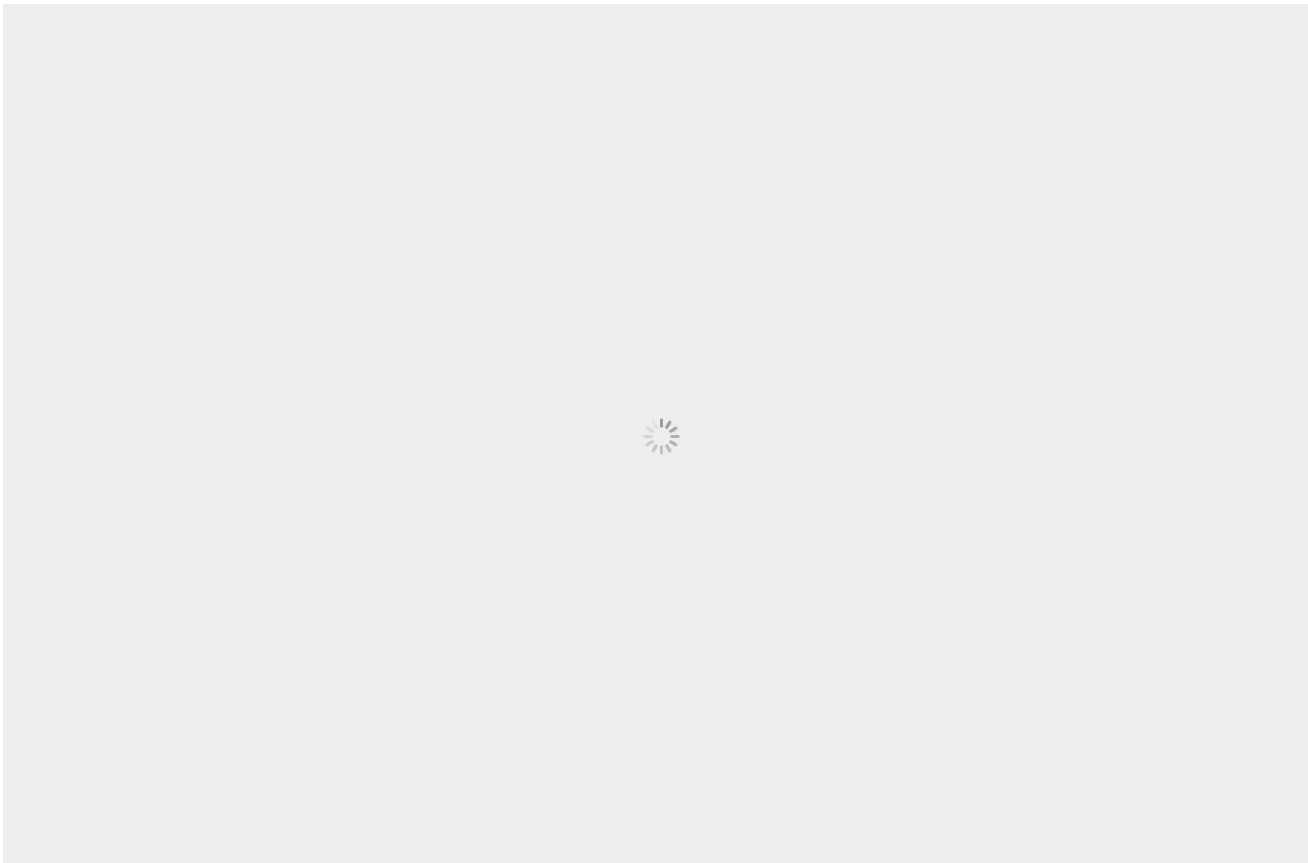
使用mvn命令打成war包。



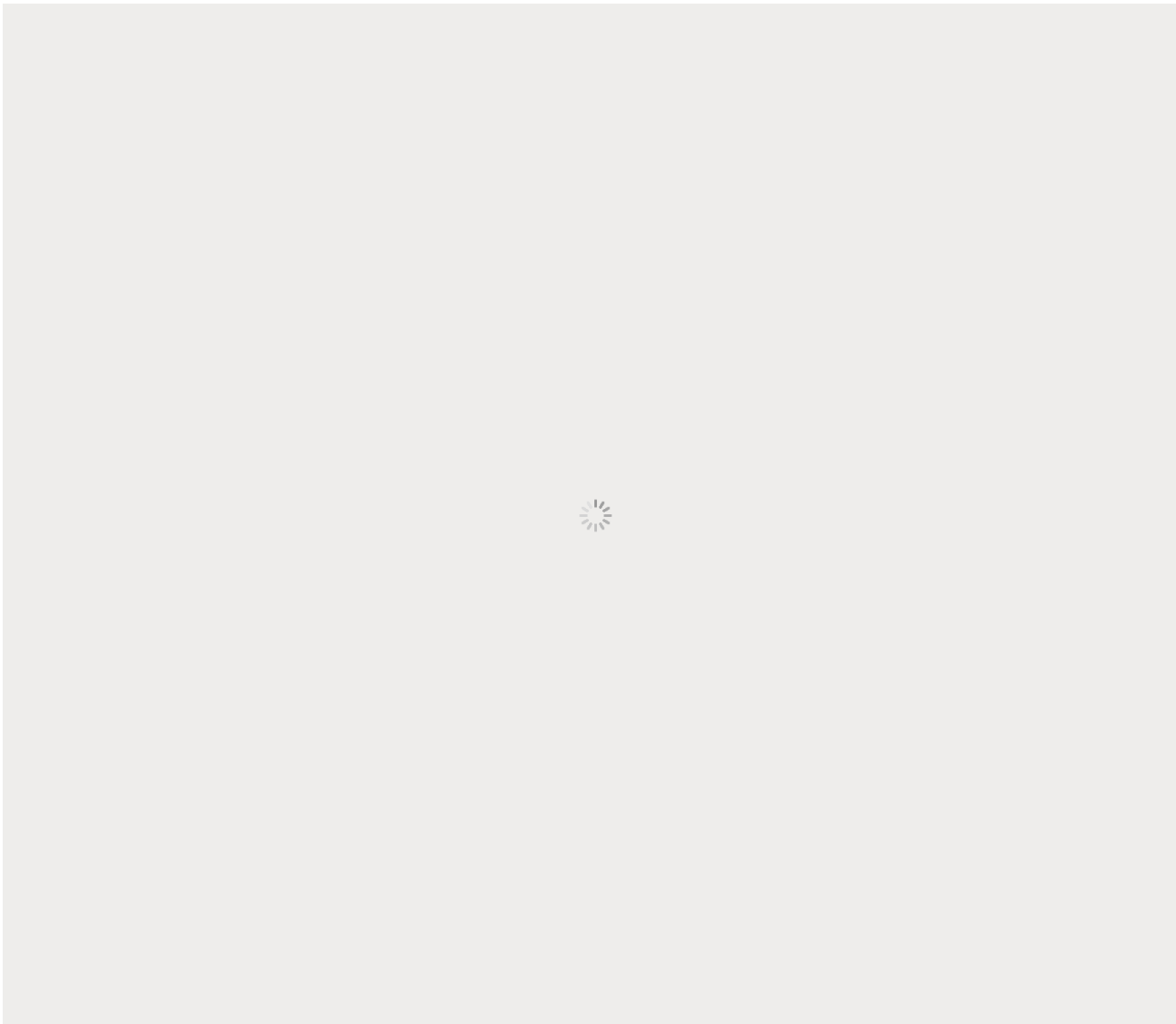
把打成的war包拷貝（其實拷貝my-web-maven文件夾也是一樣的）到tomcat中webapps目錄下：



然後到bin目錄下，雙擊



項目運行起來後，訪問<http://localhost:8080/>



證明我們的項目已經啟動成功了。

接下來我們訪問我們寫的Servlet：

`http://localhost:8080/demo?message=hello`



報錯，HTTP狀態404 - 未找到；

注意：這裡訪問的時候，我們需要把項目名稱當做contextPath，即訪問方式應該是：

`http://localhost:8080/my-web-maven/demo?message=hello`

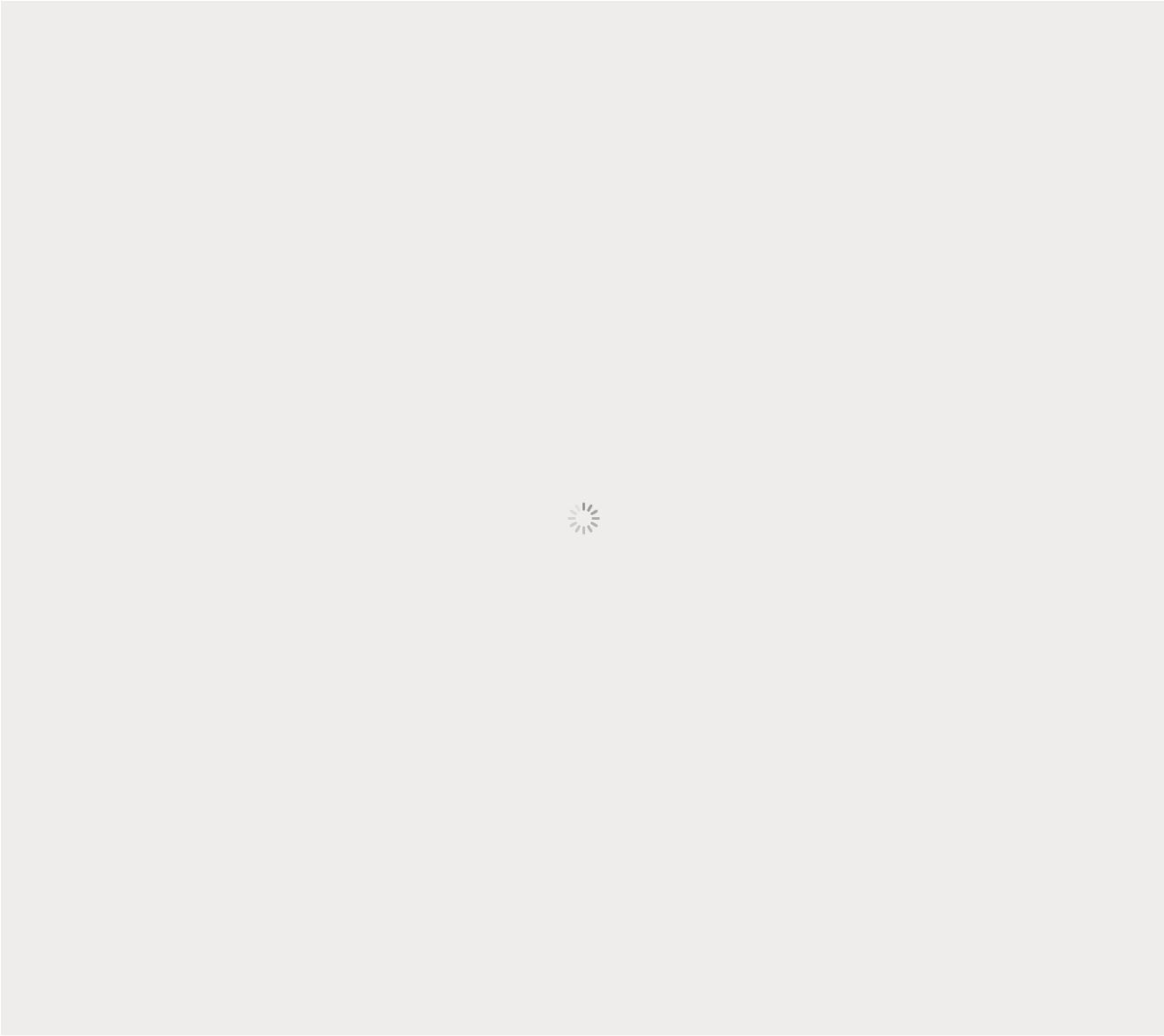
頁面上輸出

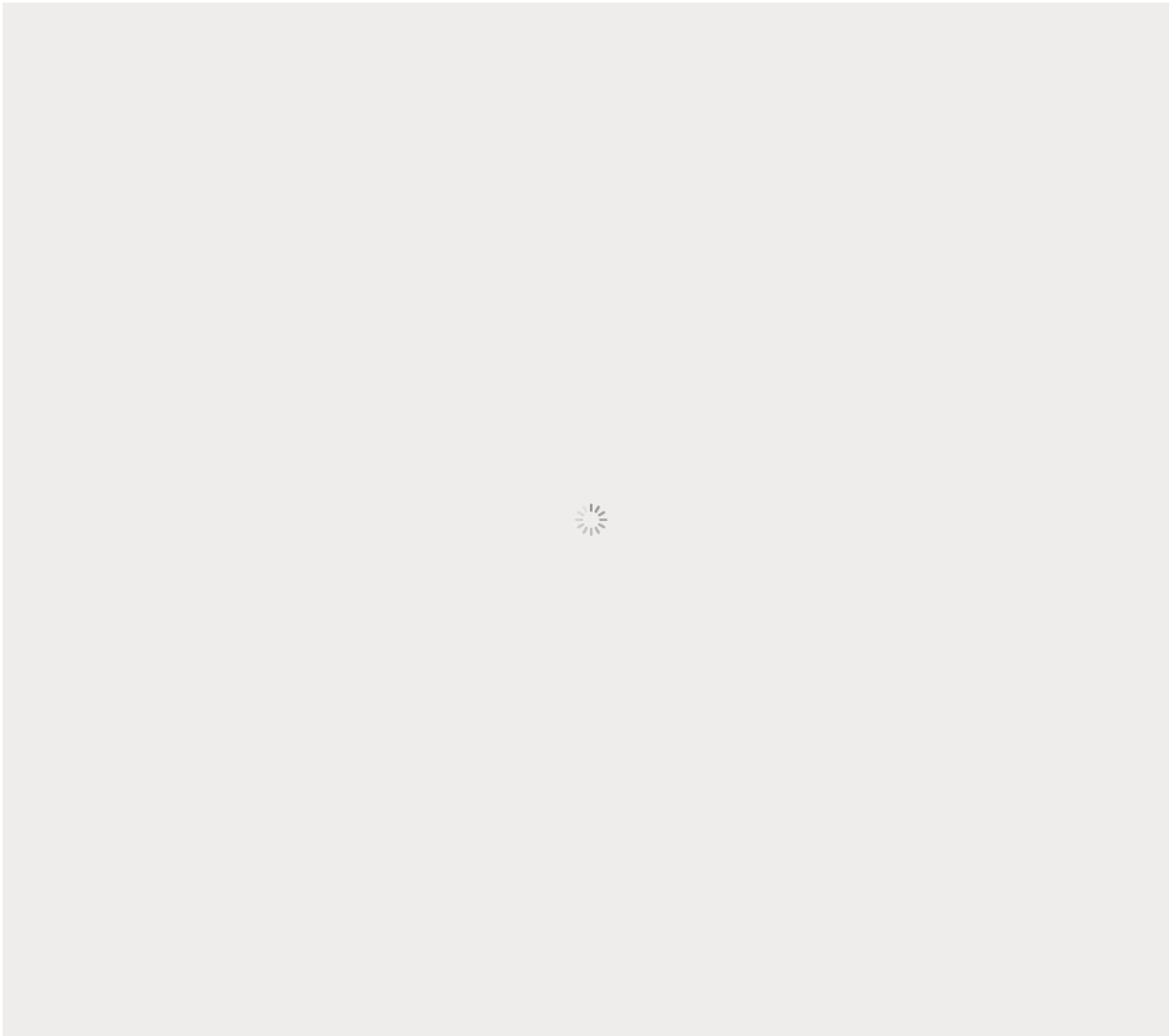
你好

輕鬆搞定，這樣我們的項目就成功的部署到tomcat中。

🔗 IDEA中的項目部署到tomcat中

創建一個servlet項目，項目名稱my-servlet。



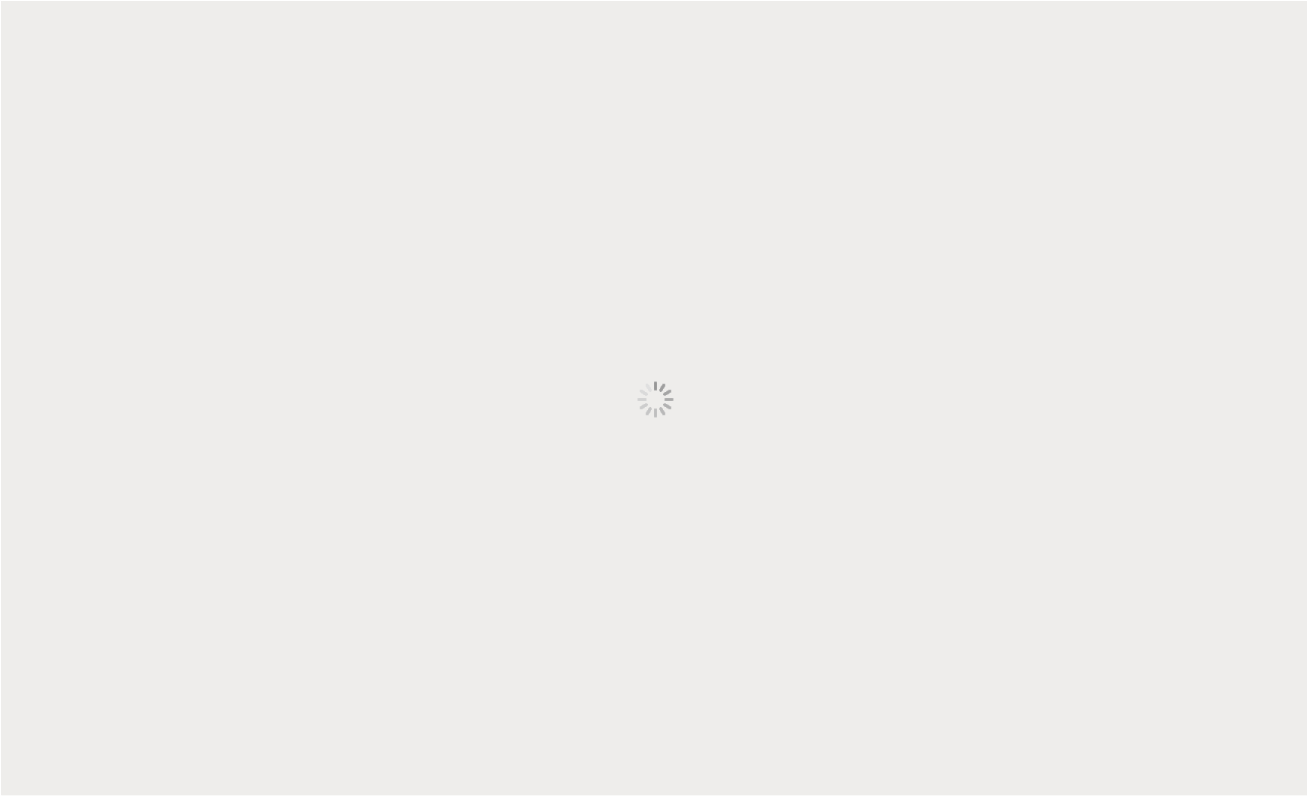


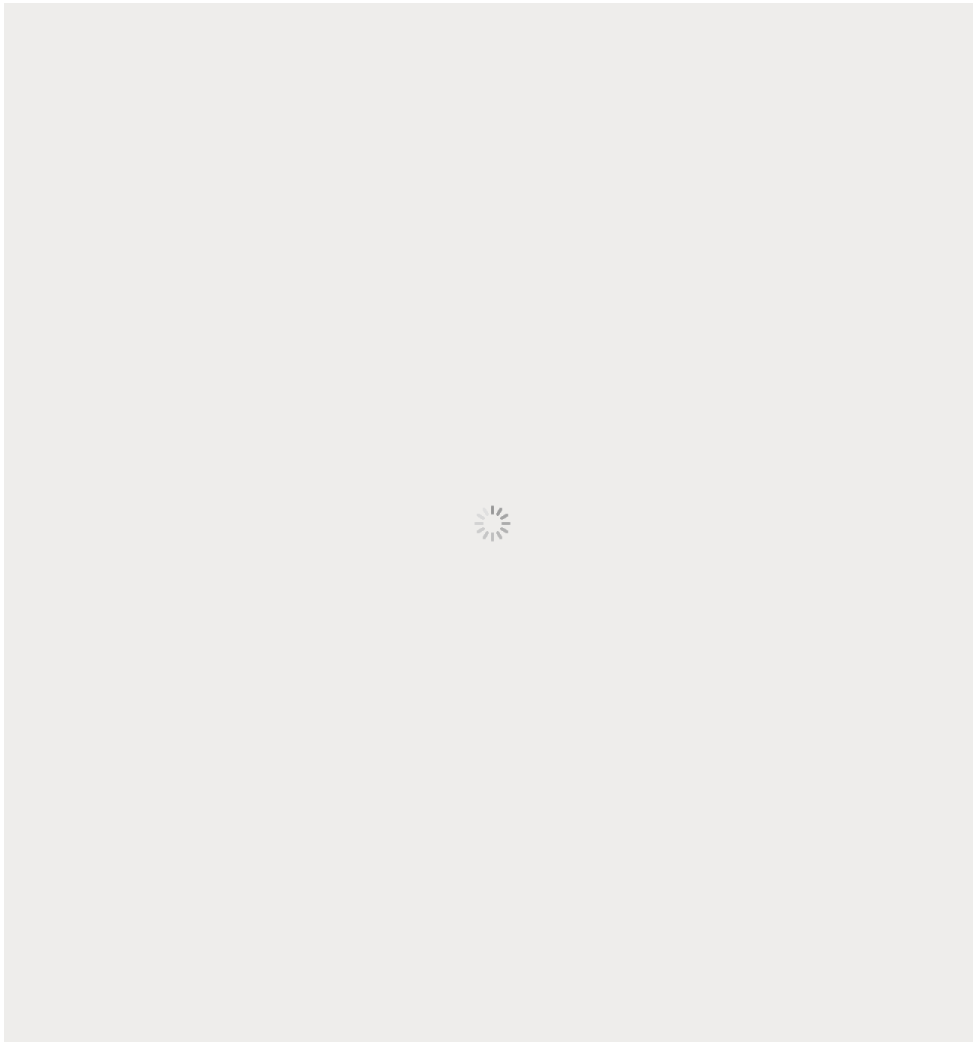
新建一個類MyServlet



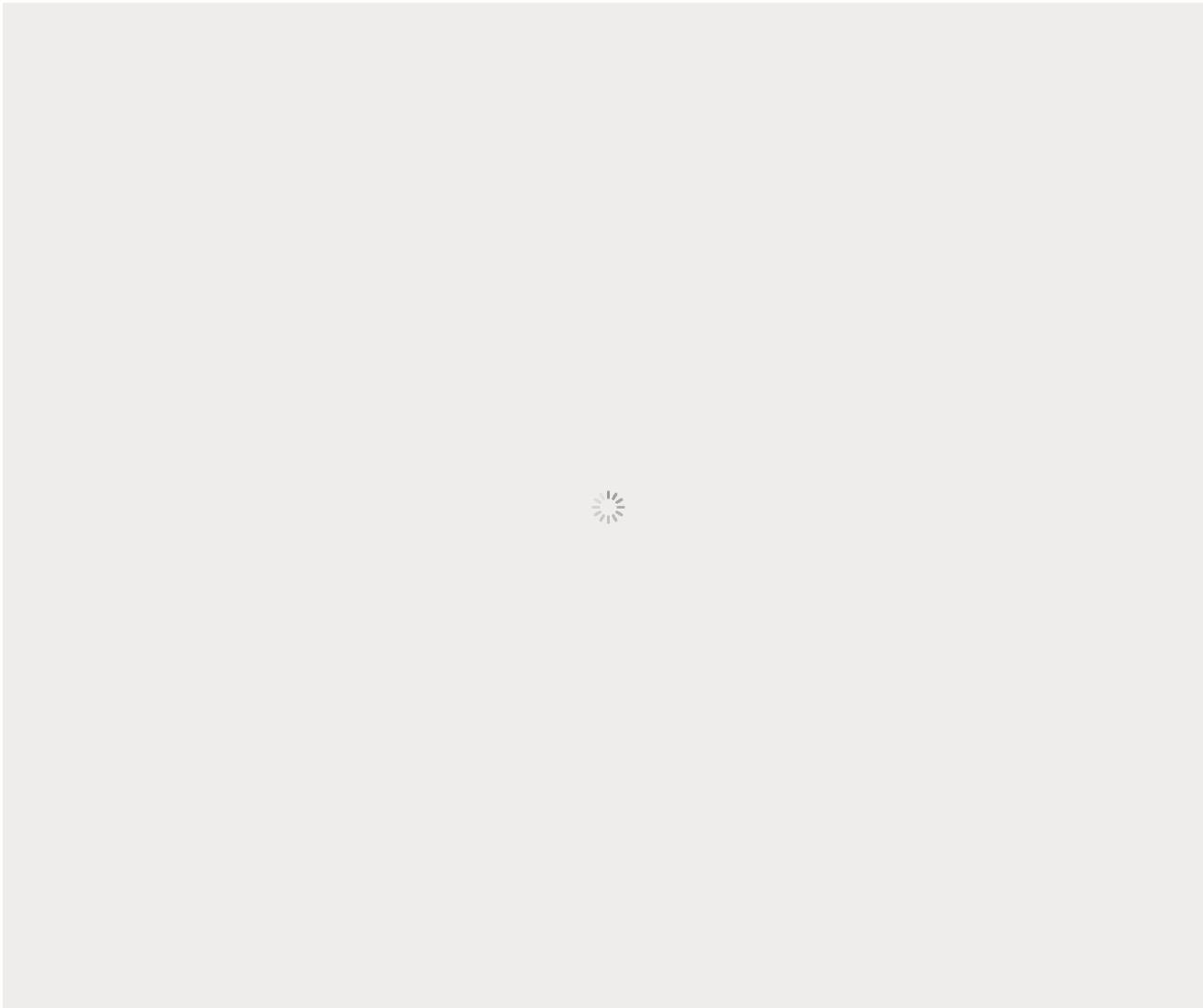
進入我們剛剛安裝的tomcat目錄，進入到lib目錄中，選中servlet-api.jar。







點擊ok。



點擊Apply，然後點擊OK。

修改MyServlet內容：

```
package com.tian.servlet;
```

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

//实现接口Servlet · 必须重写下面的这些方法
public class MyServlet implements Servlet {

    private transient ServletConfig servletConfig;

    @Override
    public void init(ServletConfig servletConfig) throws ServletException {
        this.servletConfig = servletConfig;
    }

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

    @Override
    public void service(ServletRequest request, ServletResponse response) throws ServletException {
        String servletName = servletConfig.getServletName();
        //网页响应类型 · 浏览器将其渲染为HTML格式
        response.setContentType("text/html");
        PrintWriter writer = response.getWriter();
        writer.println("<html><head></head>" + "<body> Hello this is  " + servletName +
    }

    @Override
    public String getServletInfo() {
        return "my first Servlet";
    }
}
```

```
@Override  
public void destroy() {  
}  
}
```

修改web.xml文件內容：

```
<?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  
    version="3.1">  
    <servlet>  
        <servlet-name>myServlet</servlet-name>  
        <servlet-class>com.tian.servlet.MyServlet</servlet-class>  
    </servlet>  
    <servlet-mapping>  
        <servlet-name>myServlet</servlet-name>  
        <url-pattern>/demo</url-pattern>  
    </servlet-mapping>  
</web-app>
```

另外我們把index.jsp內容修改一下，主要是為了更好的演示：

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

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

<html>

<body>

<h1> hello world </h1>

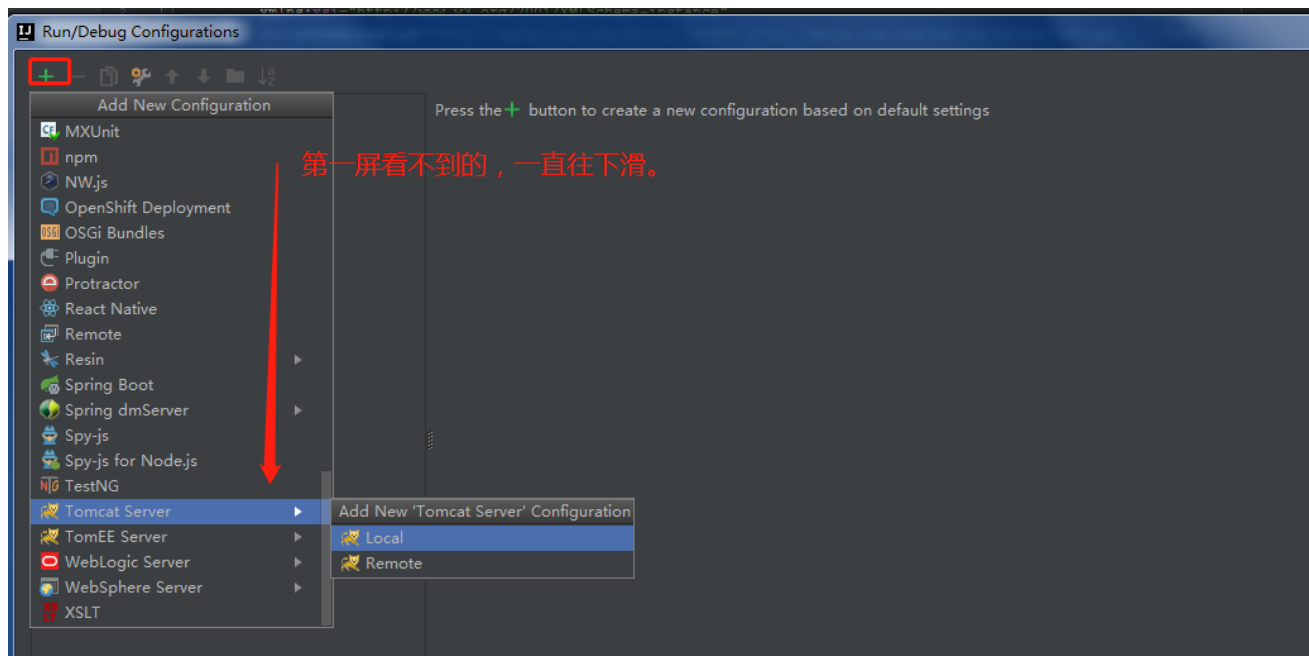
</body>

</html>
```

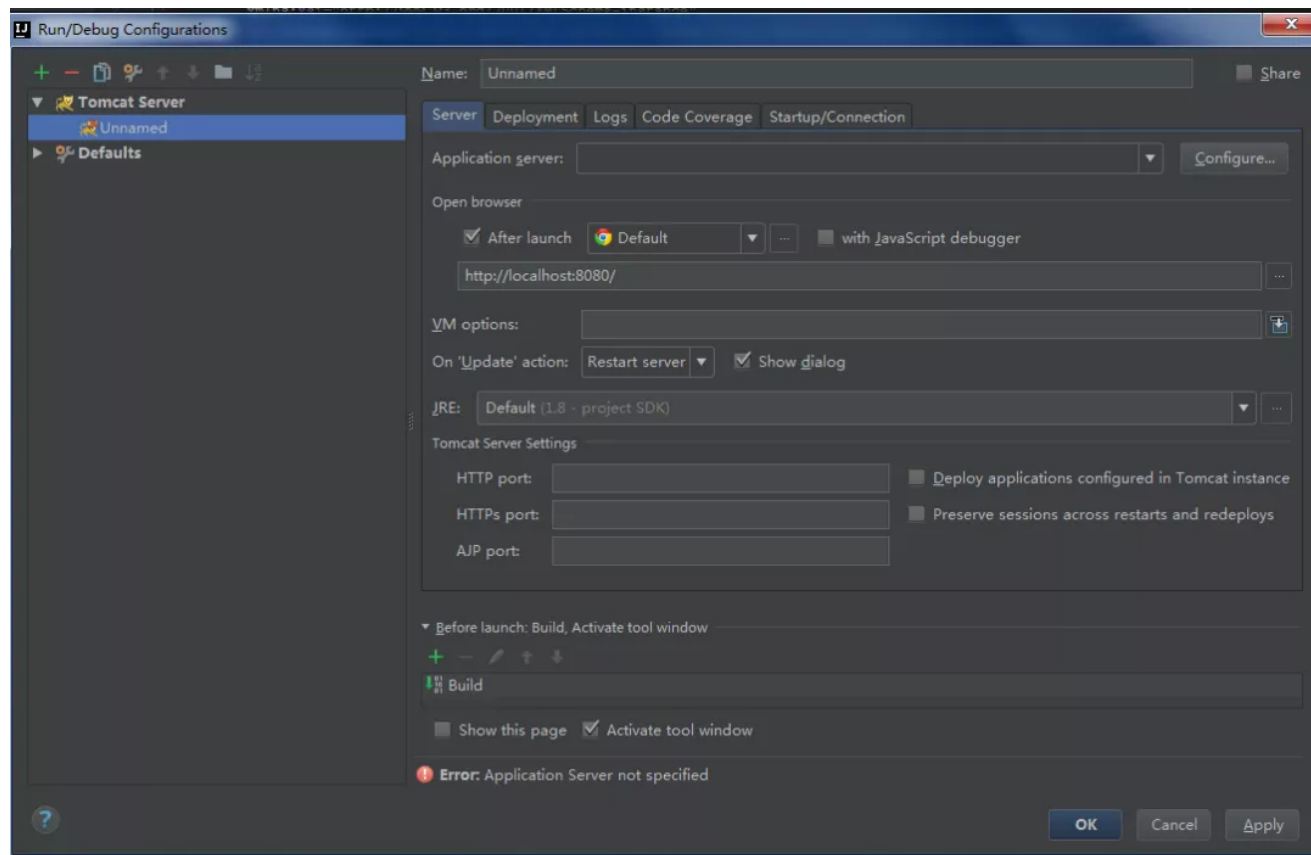
IDEA集成tomcat

在我們的IDEA中添加我們安裝的tomcat:





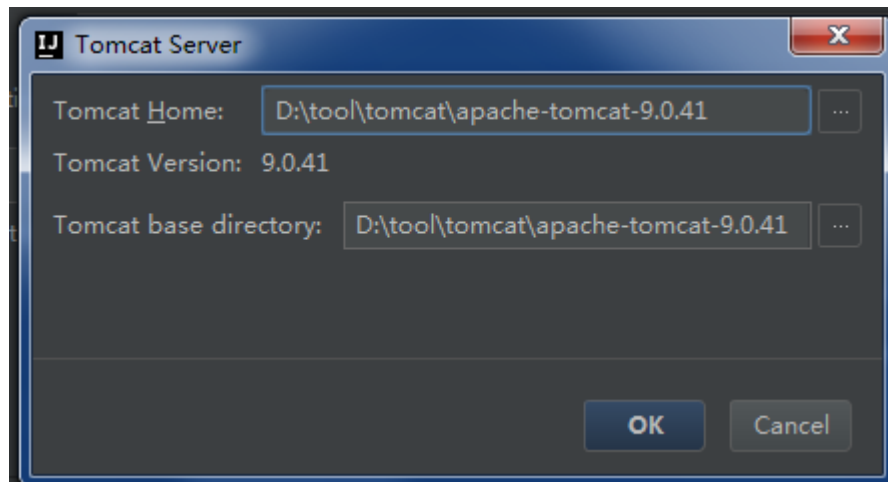
來到tomcat配置的界面：



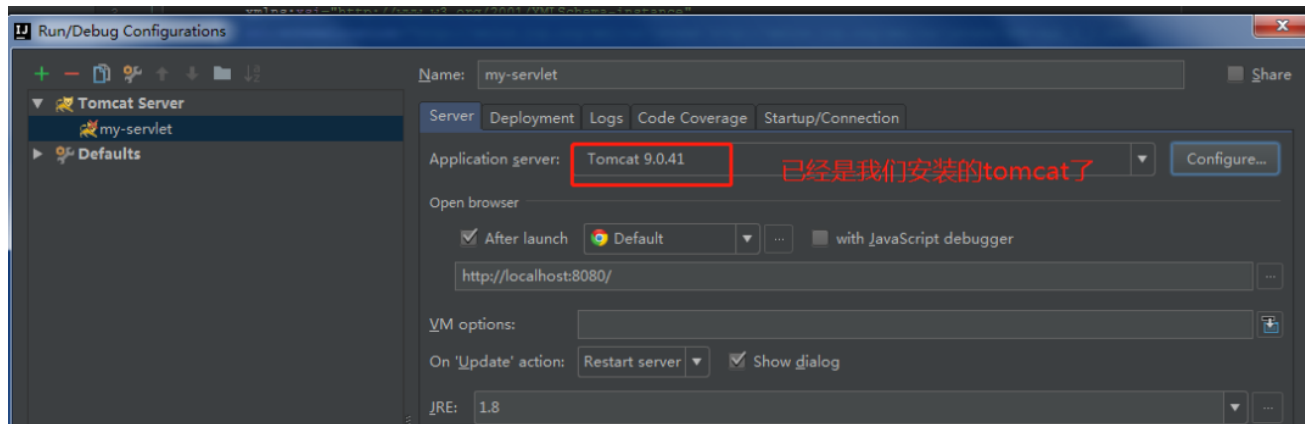
配置tomcat：



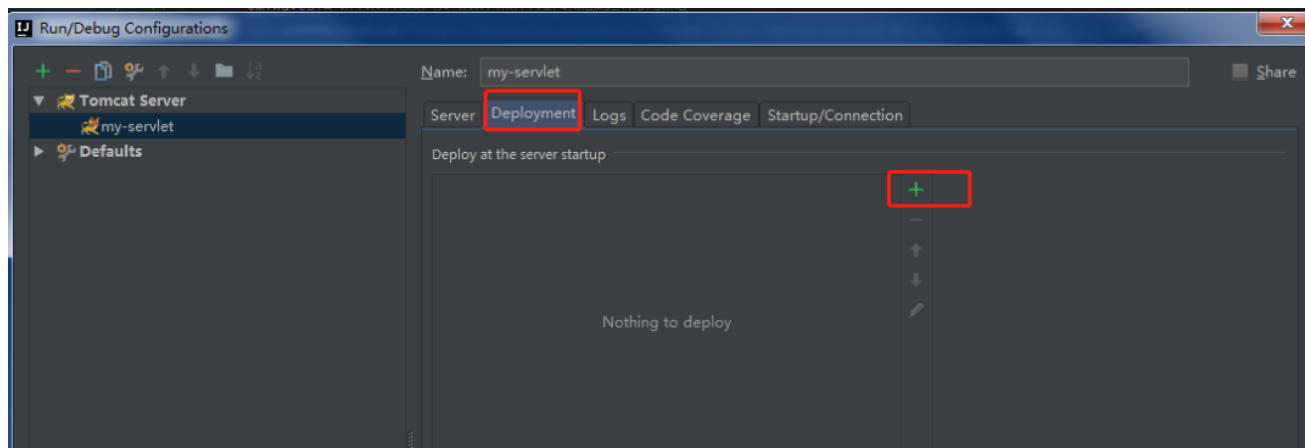
進入安裝目錄：



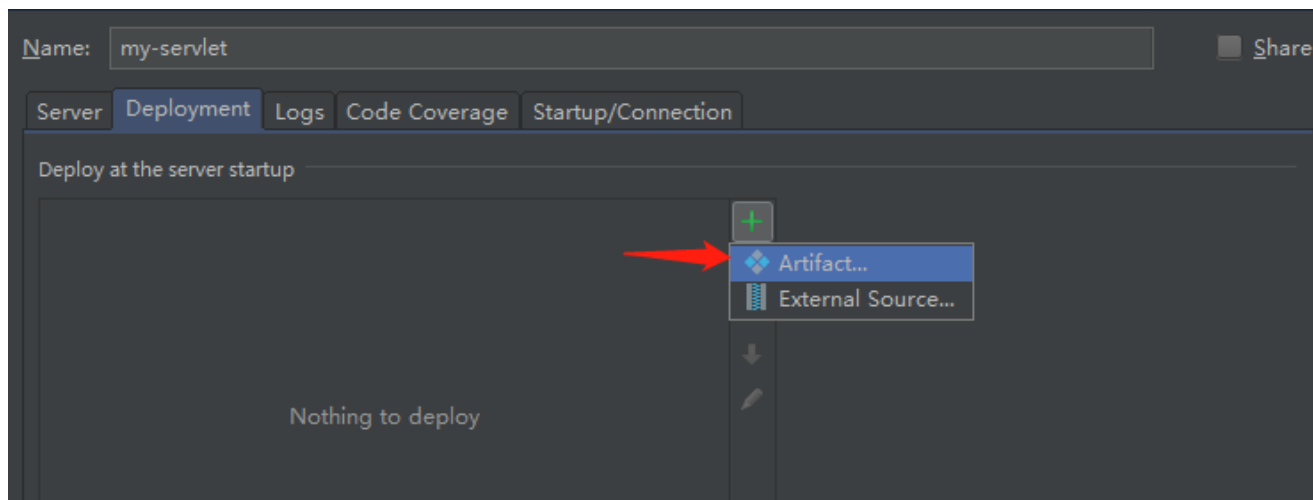
點擊OK，



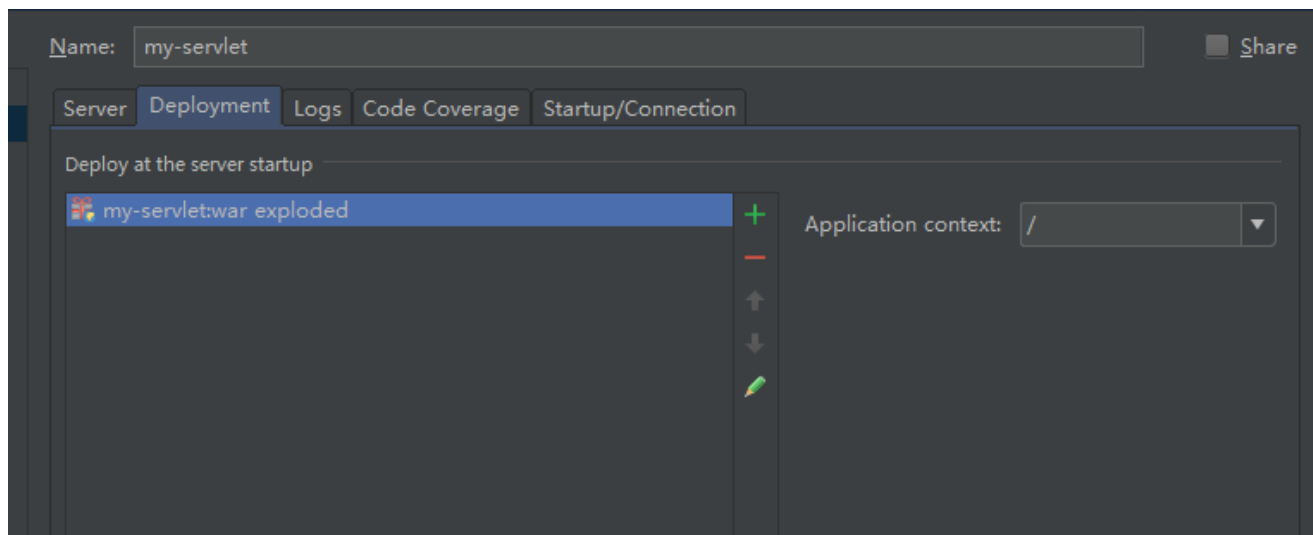
然後進入deployment欄目：



添加我們創建的servlet項目：

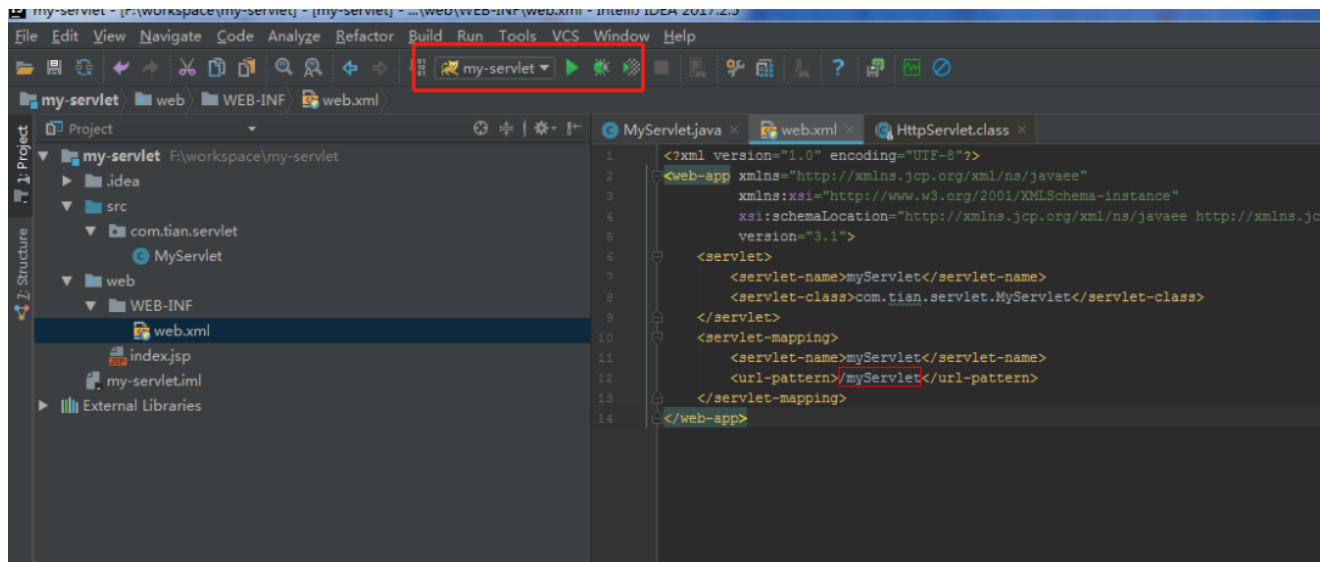


自動就添加了我們的項目：



然後點擊Apply，再點擊OK。

IDEA集成tomcat，以及把我們項目給關聯起來了。下面我們就來運行一下：



啟動tomcat

點擊綠色三角：



證明我們的項目在tomcat中已經啟動成功。

hello world

訪問我們的servlet

此時，我們就可以來訪問我們的servlet了。

訪問：<http://localhost:8080/> 頁面顯示就是我們前面修改的index.jsp的內容。

再訪問我們自己寫的Servlet：

`http://localhost:8080/demo`

Hello this is myServlet

成功的把我們的內容給輸出來了。好了本文內容就這些，簡單式tomcat入門。



tomcat是什麼，tomcat如何安裝，如何啟動tomcat，如何部署我們自己的Servlet項目，我們在IDEA如何集成tomcat以及如何啟動tomcat。

「人只要不失去方向，就不会失去自己」

長按掃碼關注

一起學Java



優秀文章，莫要白嫖，如果覺得有點用的話，請毫不留情地素質三連吧，分享、點贊、在看，我不挑，因為這將是我寫作更多優質文章的最強動力。

喜歡此內容的人還喜歡

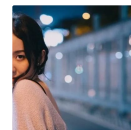
Hadoop周邊組件學習筆記

數據社



Nginx從入門到干活，看這一篇足矣

架構師修行之路



我用Redis實現了一個輕量級的搜索引擎！

碼猿技術專欄

