

Servlet综合练习

本节目标

-

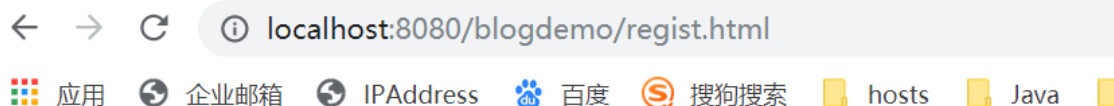
介绍

功能简介

实现一个简易的博客功能，包括用户登录、注册，发表新文章，显示文章详情，显示文章列表，显示文章列表接口访问量。

业务流程

- 用户注册页面：需要注册后才可以登录并操作



用户注册

用户名: 密码:

- 用户登录页面：登录后才可以发表新文章



用户登录

用户名: 密码:
[注册账号](#)

- 发表文章：如果没有登录，就退出到登录页面，只有登录的用户才可以发表新文章

← → ↻

localhost:8080/blogdemo/add.html

应用

企业邮箱

IPAddress

百度

搜狗搜索

hosts

Java

工具

项目

spring

数据库

添加博客文章

标题: 欢喜冤家

内容: 方法法师打发士大夫时代大厦

图片: 选择文件 2 个文件

提交

- 显示文章详情：发表文章成功后跳转到文章详情。也可以单独访问文章详情

← → ↻

localhost:8080/blogdemo/detail?id=8

应用

企业邮箱

IPAddress

百度

搜狗搜索

hosts

Java

工具

项目

spring

数据库

前端

比特

linux

文章详情

标题: 欢喜冤家

内容: 方法法师打发士大夫时代大厦



比特科技
BIT-TECHNOLOGY



- 显示文章列表：url输入地址后访问，显示所有文章信息

← → ↻

localhost:8080/blogdemo/list

应用

企业邮箱

IPAddress

百度

搜狗搜索

hosts

Java

工具

项目

spring

数据库

前端

比特

linux

系统

[[{"id":2,"title":"范德萨发顺丰","content":"ds佛挡杀佛水电费水电费水电费","user":{"id":1,"name":"9527","password":null},"imgs":["http://localhost:8080/upload/img/java内存模型.jpg","http://localhost:8080/upload/img/多线程 (2).png"]}, {"id":4,"title":"手动阀","content":"第三方发顺丰的撒旦法太丰富的","user":{"id":1,"name":"9527","password":null},"imgs":["http://localhost:8080/upload/img/java课程安排 - 副本.jpg"]}, {"id":5,"title":"1","content":"撒旦撒放的地方沙发垫","user":{"id":2,"name":"stu","password":null},"imgs":["http://localhost:8080/upload/img/课调.png"]}, {"id":6,"title":"欢喜冤家","content":"方法法师打发士大夫时代大厦","user":{"id":2,"name":"stu","password":null},"imgs":["http://localhost:8080/upload/img/类的加载.png"]}, {"id":7,"title":"欢喜冤家","content":"方法法师打发士大夫时代大厦","user":{"id":2,"name":"stu","password":null},"imgs":["http://localhost:8080/upload/img/jvm main.png"]}, {"id":8,"title":"欢喜冤家","content":"方法法师打发士大夫时代大厦","user":{"id":2,"name":"stu","password":null},"imgs":["http://localhost:8080/upload/img/bit.png","http://localhost:8080/upload/img/bit-logo.jpg"]}]

- 显示文章列表访问量：统计服务器启动后，新增文章接口的访问次数

list接口的访问次数为：1

使用的技术

maven：使用maven来管理依赖，打包项目

mysql：使用MySQL作为业务数据的存储

html：使用HTML来编写前端页面

tomcat：使用Tomcat作为Web项目部署的服务器

servlet：每个页面调用后台接口都需要使用Servlet来完成业务。

session：在登录后才可以访问新增文章接口，否则直接返回到登录页面

jackson：jackson是java对象与JSON字符串数据进行序列化、反序列化的工具。

多线程：synchronized、volatile关键字，在多线程下访问Servlet共享变量，需要保证线程安全

项目准备

需要的资源

Maven、IDEA、MySQL、Chrome浏览器、Fiddler4抓包工具

创建Web项目

1. 在IDEA中，点击 File -> New -> Module，选择Maven -> Next。
2. 在GroupId和ArtifactId中分别输入组织名和项目名，注意上面的两项Add as module to和Parent都设置为none。完成以后Next。
3. 设置项目名和本地保存路径，完成后点击Finish
4. 在弹出的项目pom.xml文件中配置：

```
<?xml version="1.0" encoding="UTF-8"?>
<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/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.bit</groupId>
    <artifactId>blogdemo</artifactId>
    <version>1.0-SNAPSHOT</version>
    <!-- web项目需要配置为war，表示打包为war文件 -->
    <packaging>war</packaging>
```

```

<dependencies>
  <!-- MySQL数据库JDBC驱动包 -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.38</version>
  </dependency>
  <!-- jackson: 提供Java对象与JSON数据格式进行序列化
        及反序列化的支持 -->
  <dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.8.9</version>
  </dependency>

  <!-- 使用官方的servlet依赖，并配置为scope=provided，表示在打包时不打入war包
        中 -->
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.0.1</version>
    <scope>provided</scope>
  </dependency>

  <!-- 日志记录框架: logback -->
  <dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-classic</artifactId>
    <version>1.2.3</version>
  </dependency>
</dependencies>

<build>
  <!-- 最后使用mvn package命令打包的文件名称 -->
  <finalName>${project.artifactId}</finalName>
  <plugins>
    <plugin>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.1</version>
      <configuration>
        <source>1.8</source>
        <target>1.8</target>
        <encoding>UTF-8</encoding>
      </configuration>
    </plugin>
  </plugins>
</build>
</project>

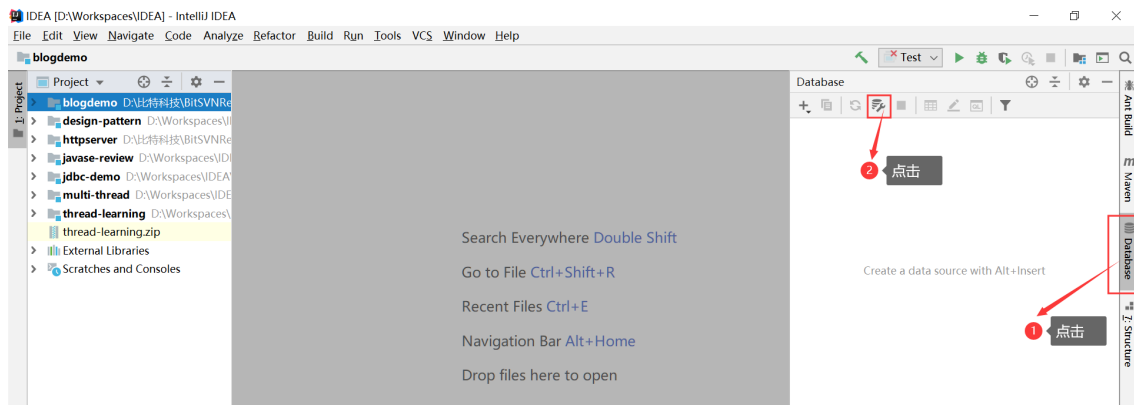
```

5. 在

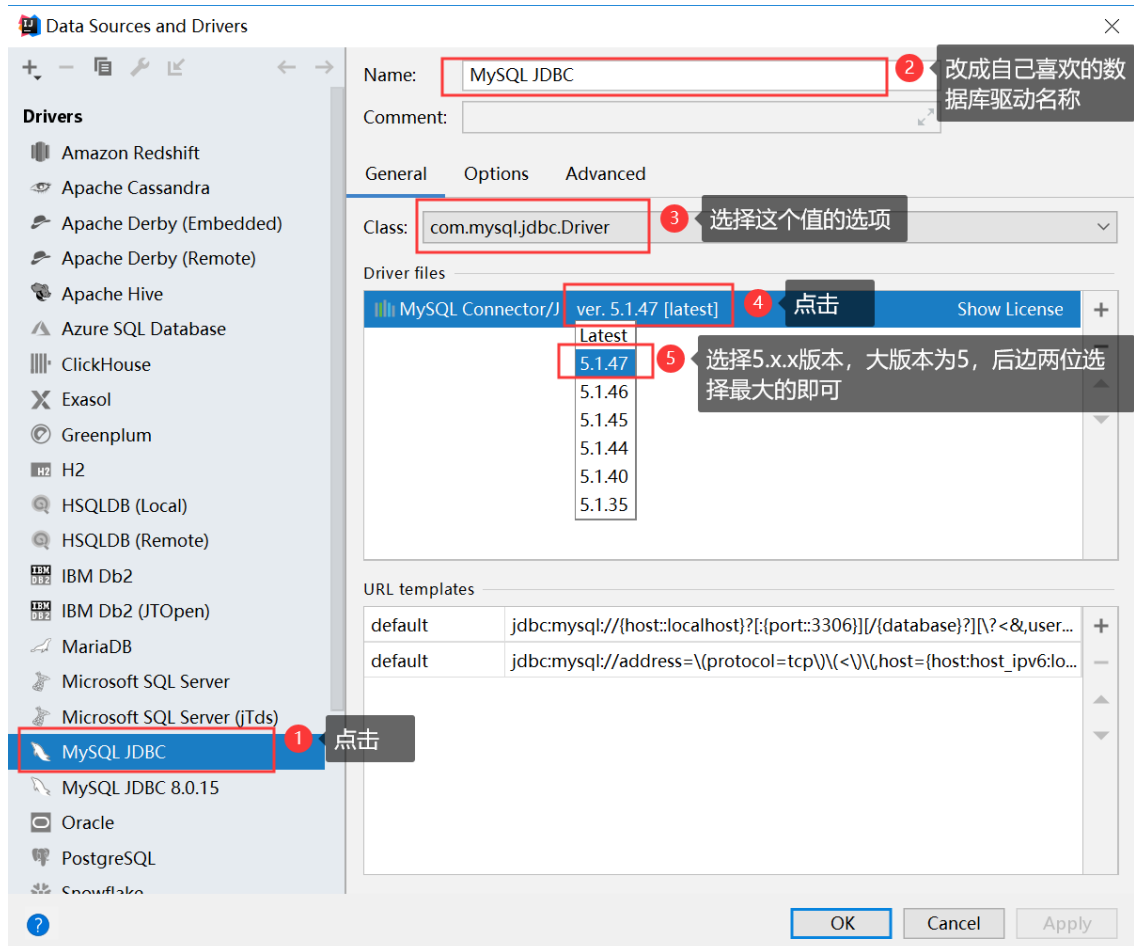
项目及IDEA配置

- idea中的数据库连接配置

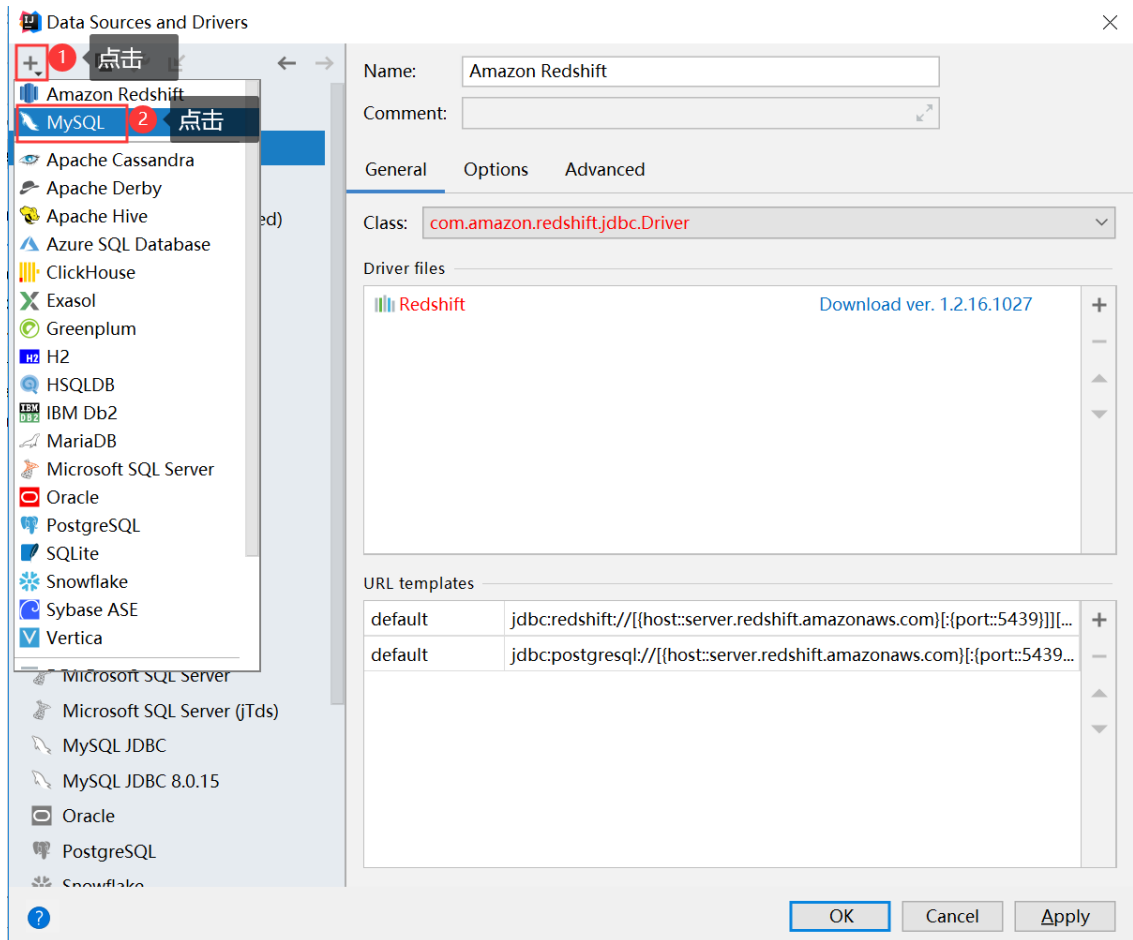
(1) 从idea的database面板打开数据库驱动配置



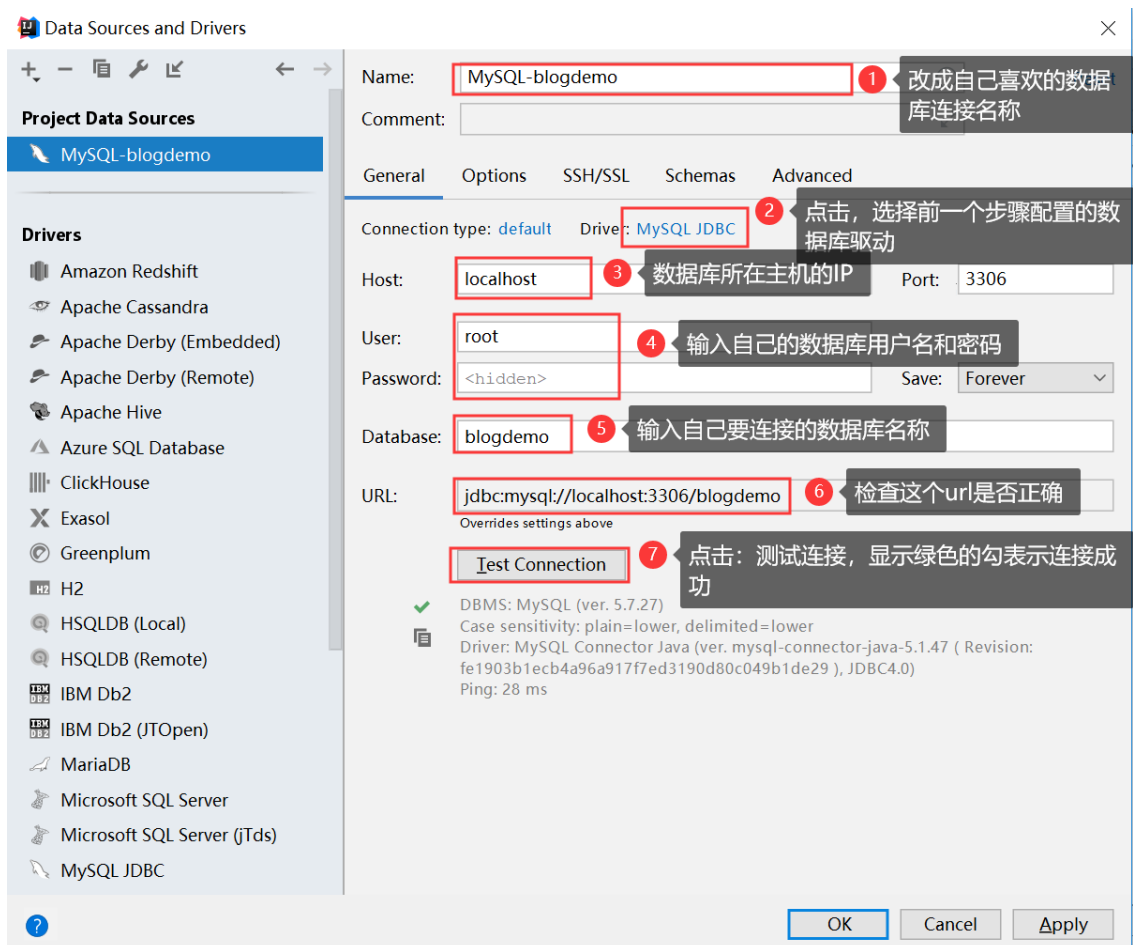
(2) 配置数据库驱动



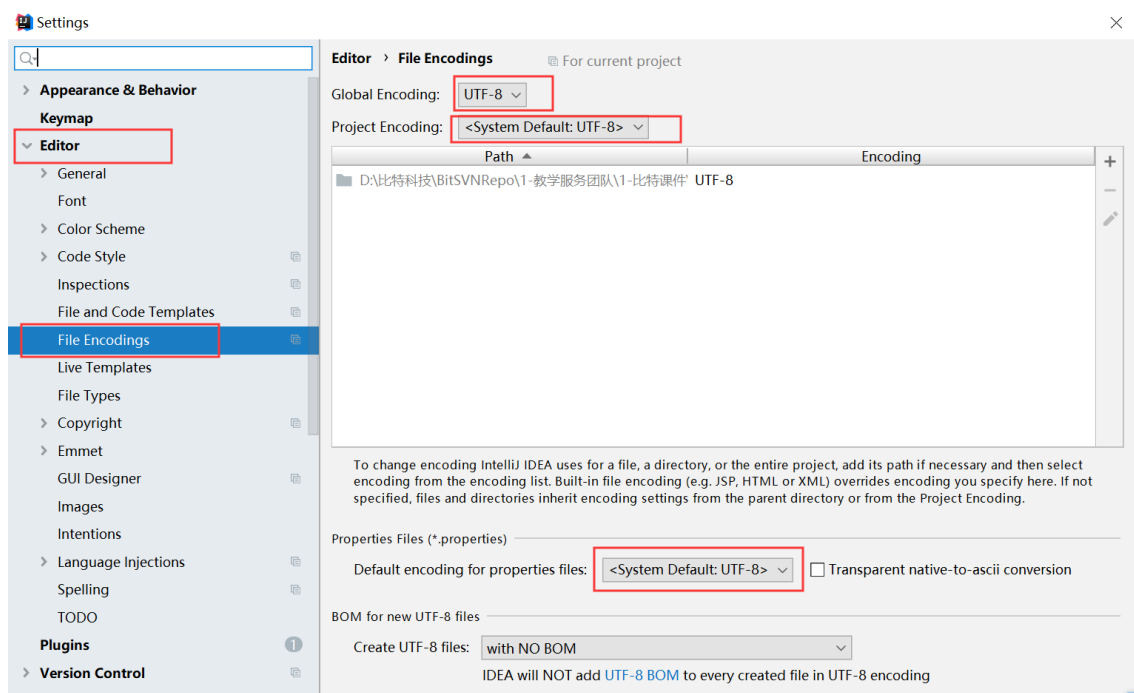
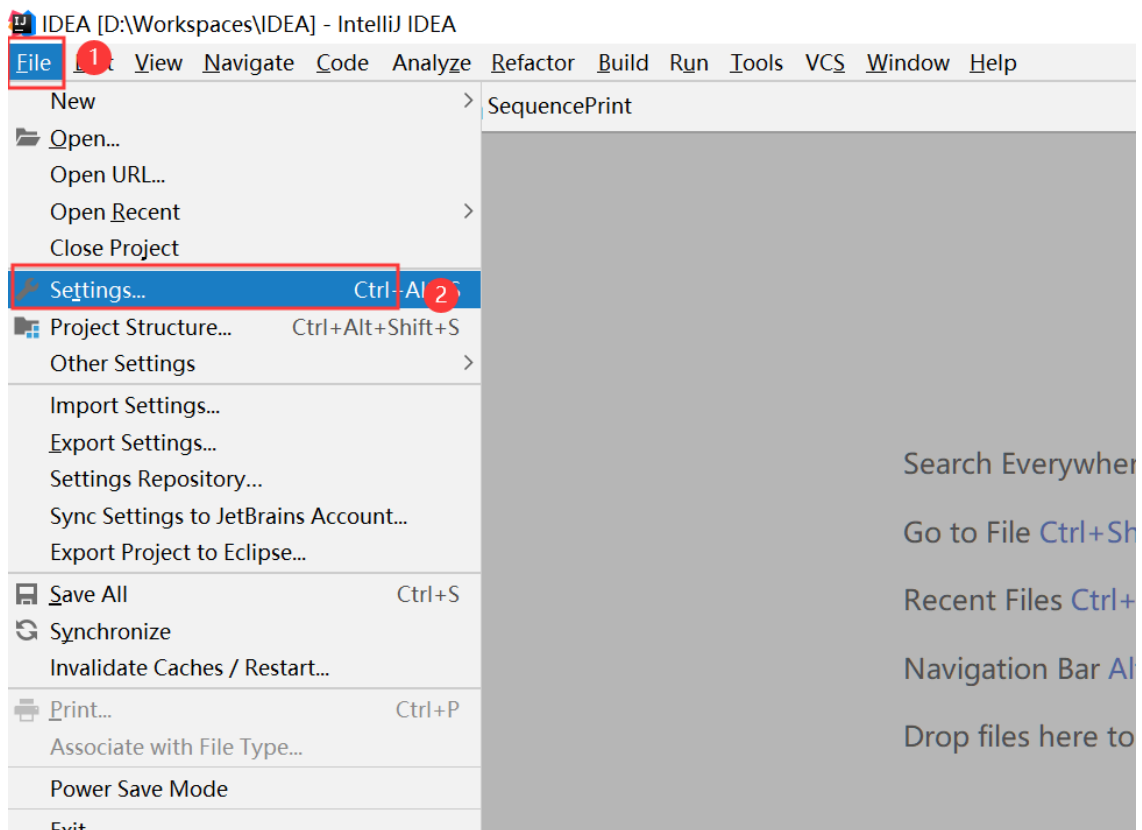
(3) 打开数据库连接配置



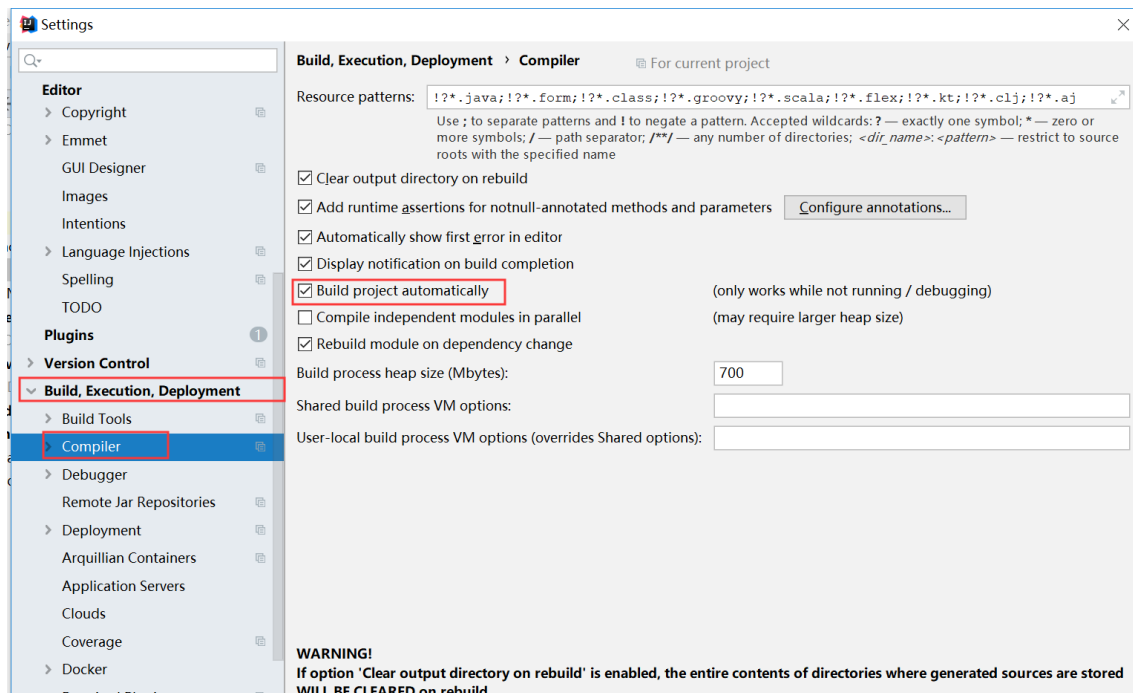
(4) 配置数据库连接



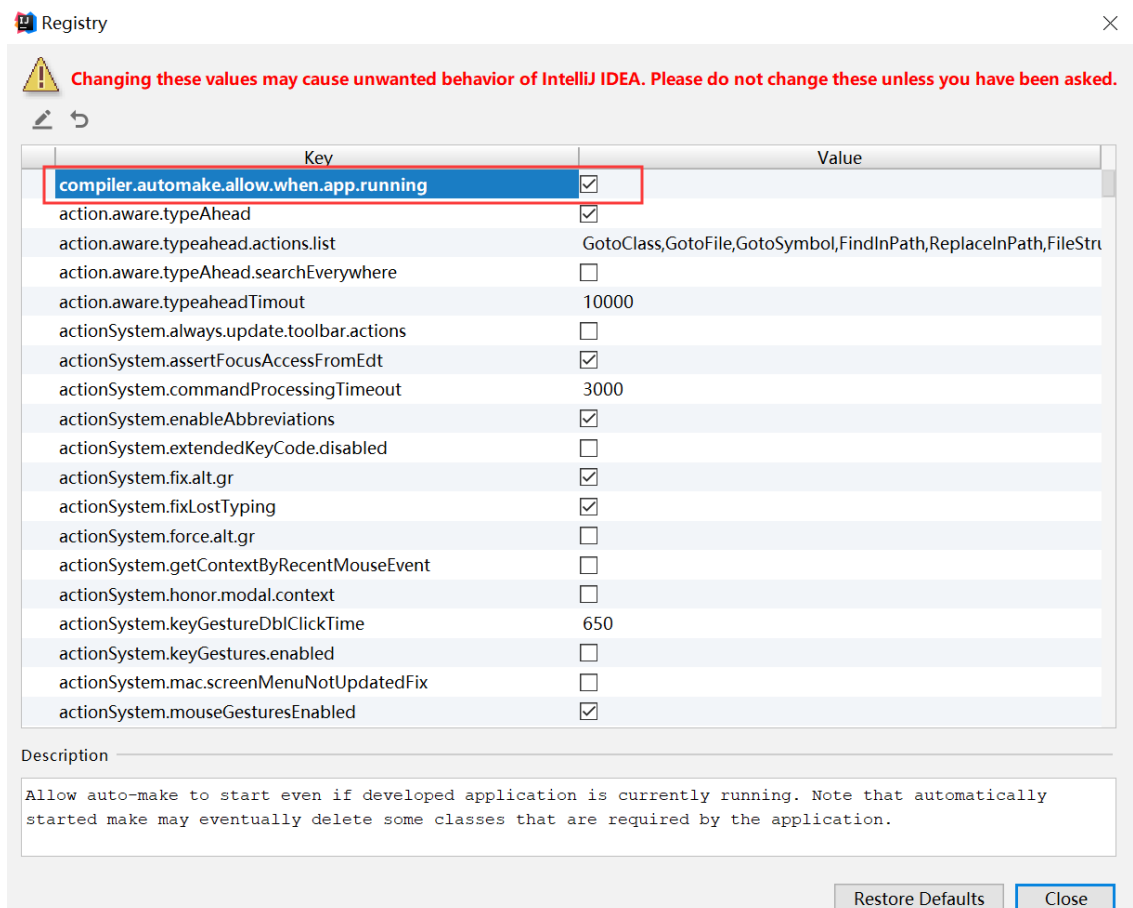
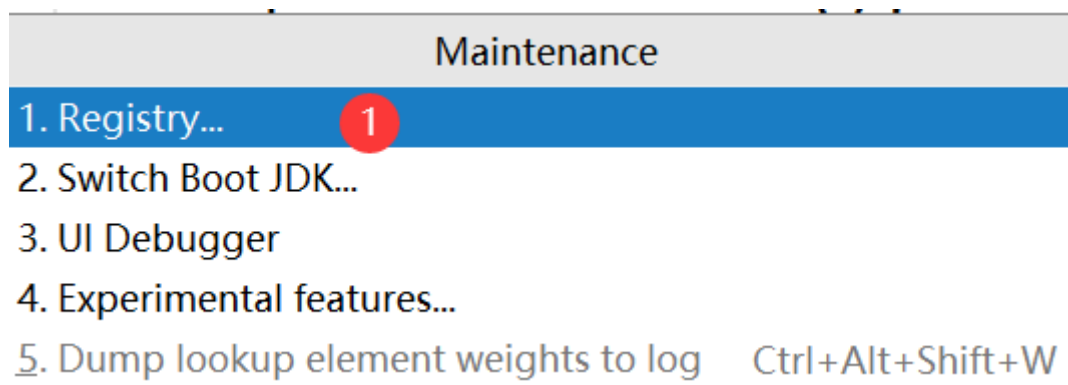
- IDEA编码设置



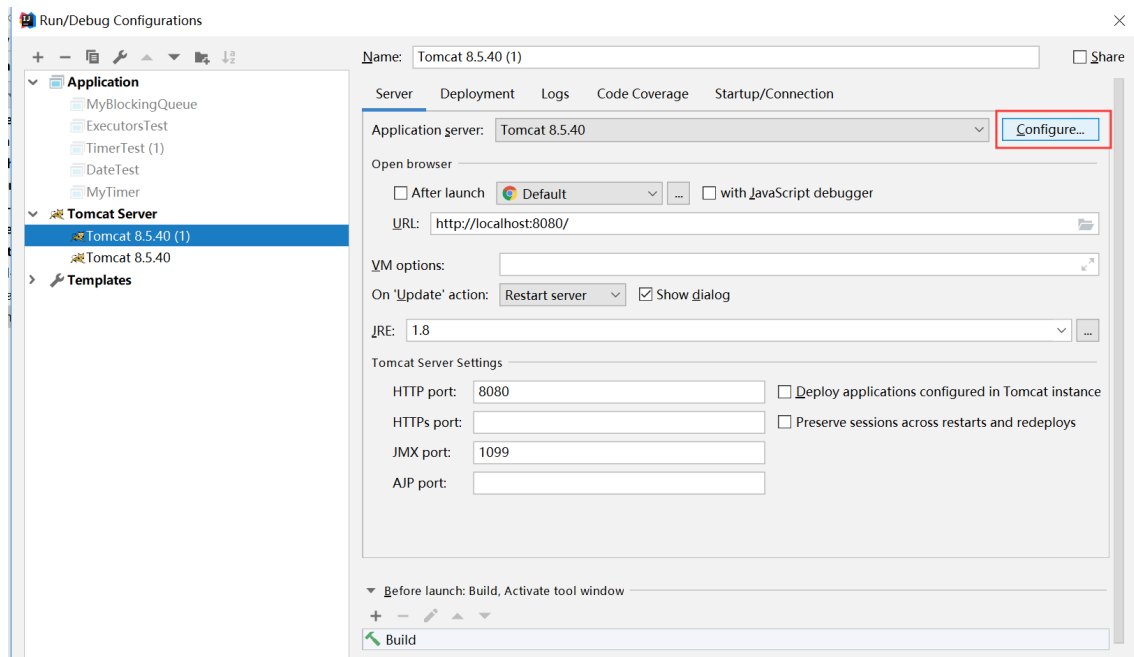
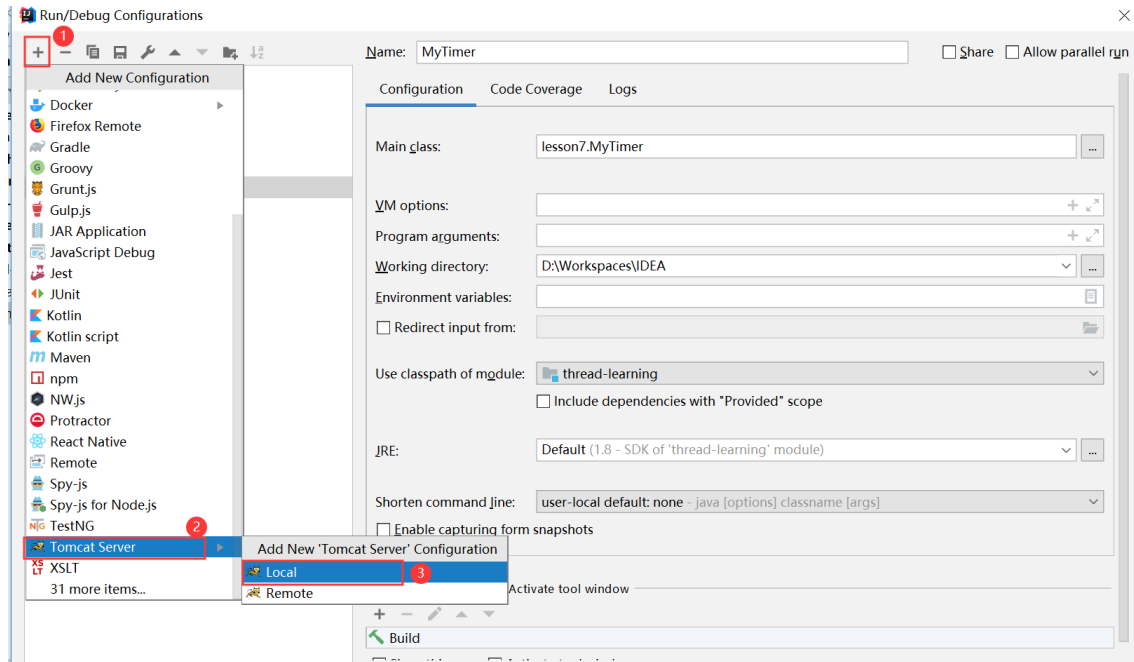
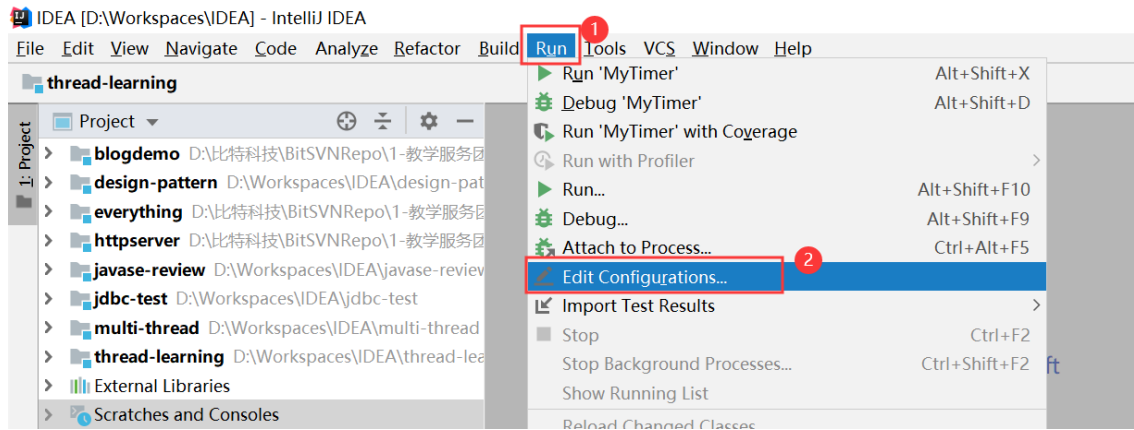
- 自动编译设置

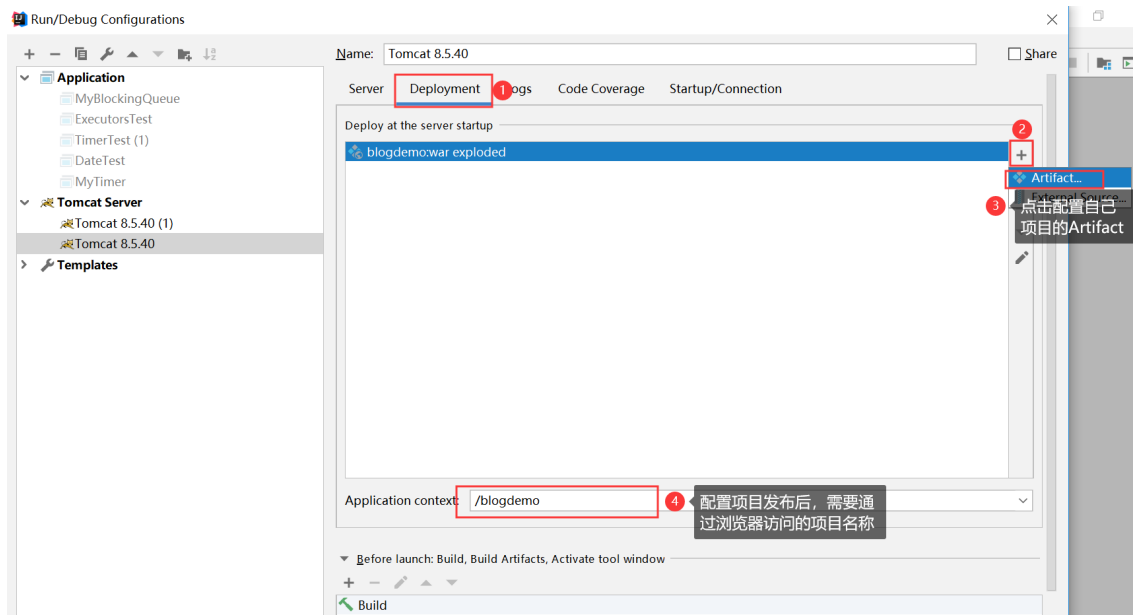
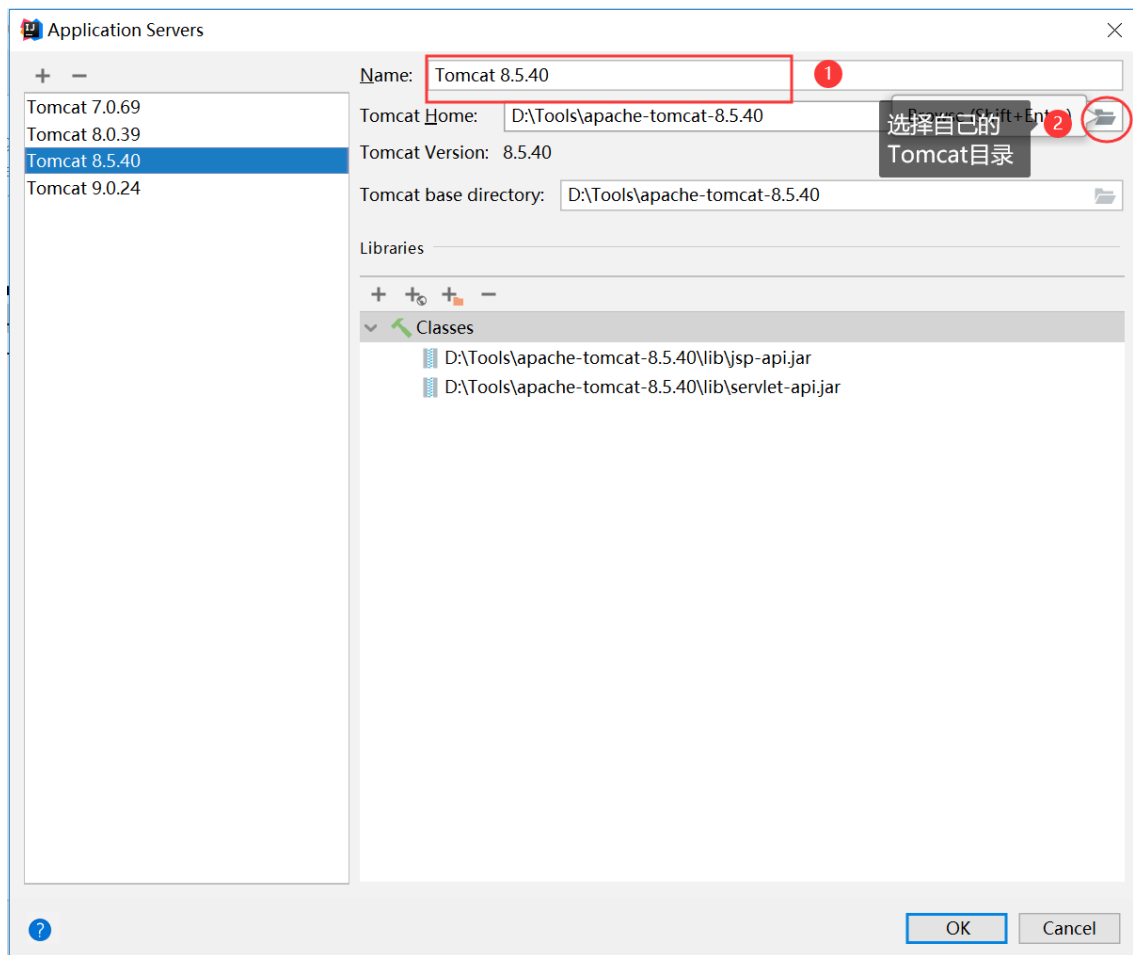


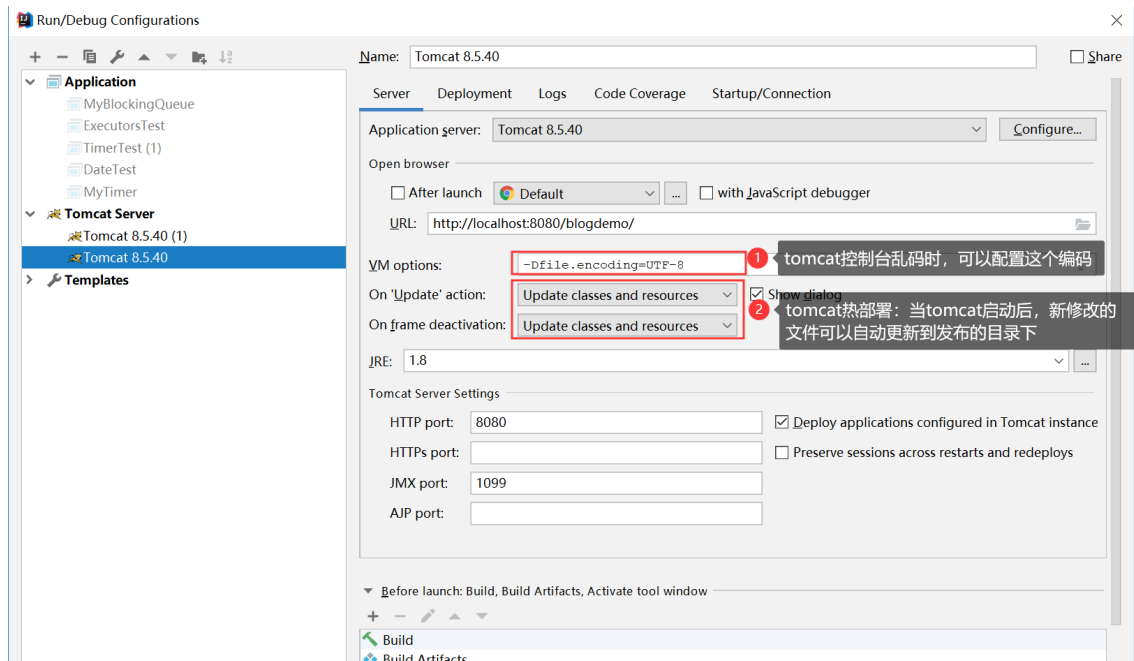
再点击shift+ctrl+alt+/键，弹出窗点击如下



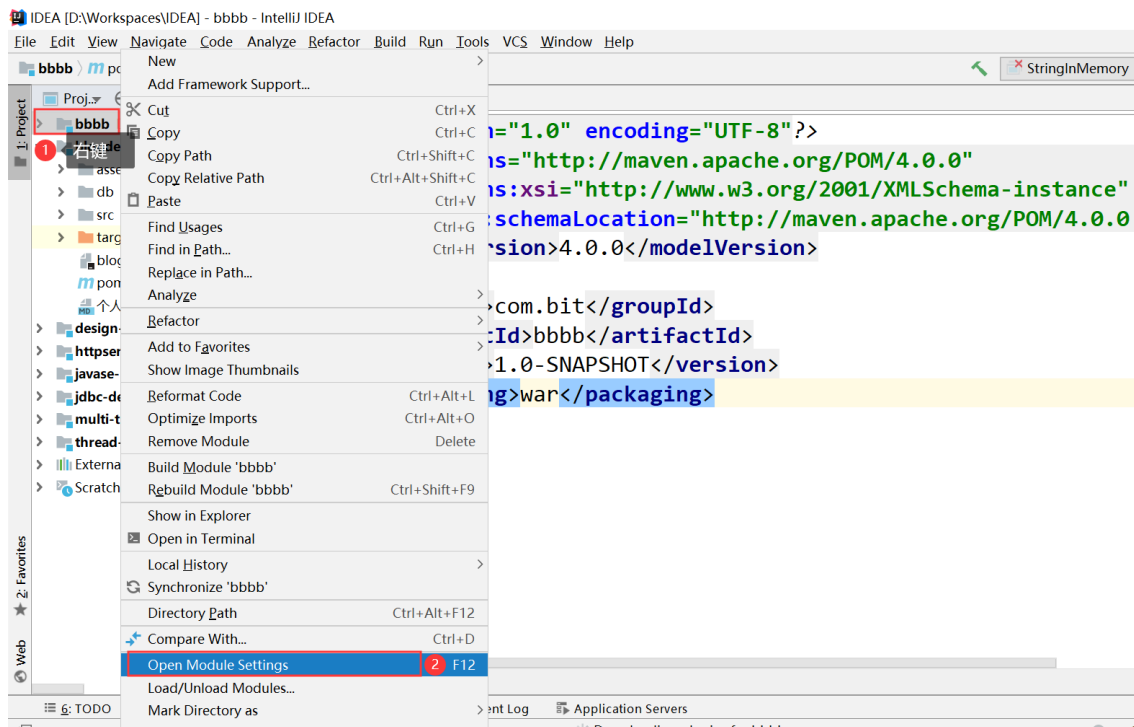
- tomcat部署项目自动发布设置

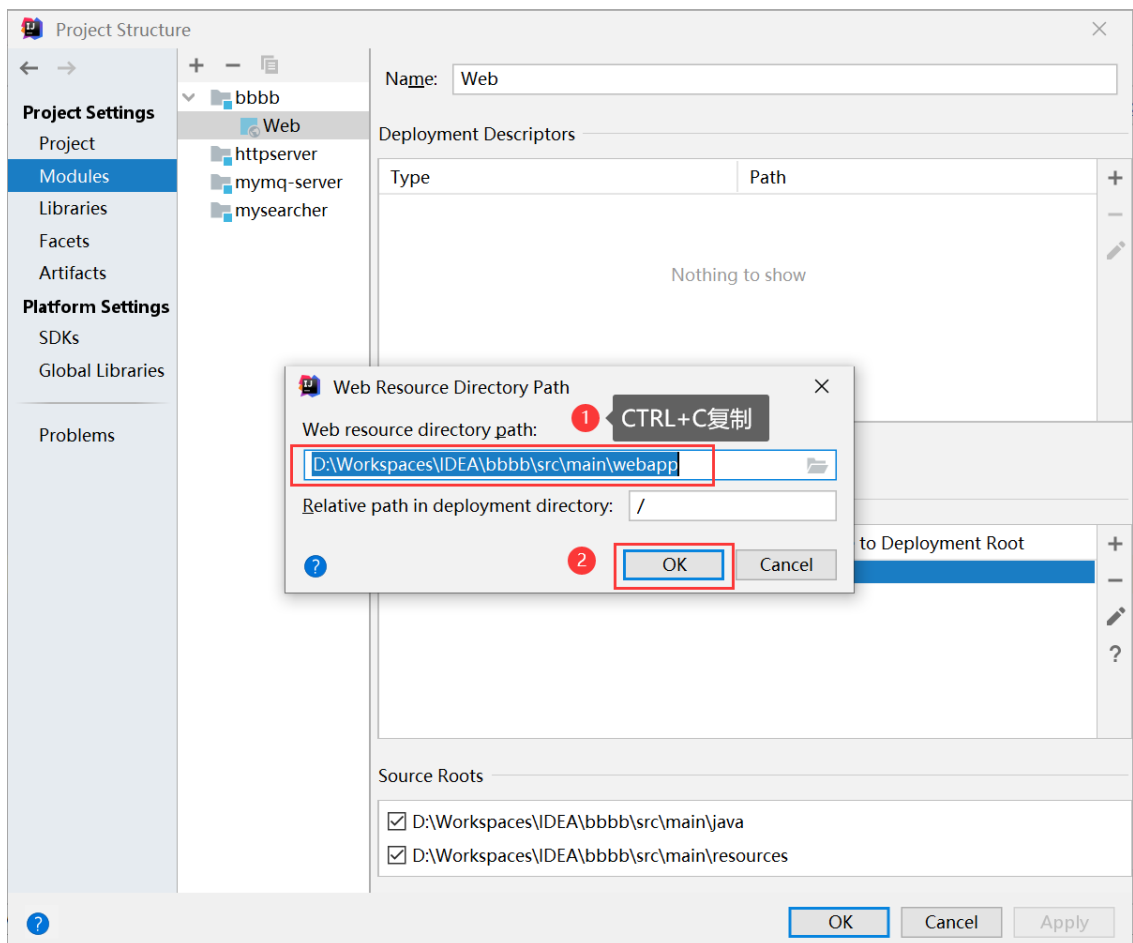
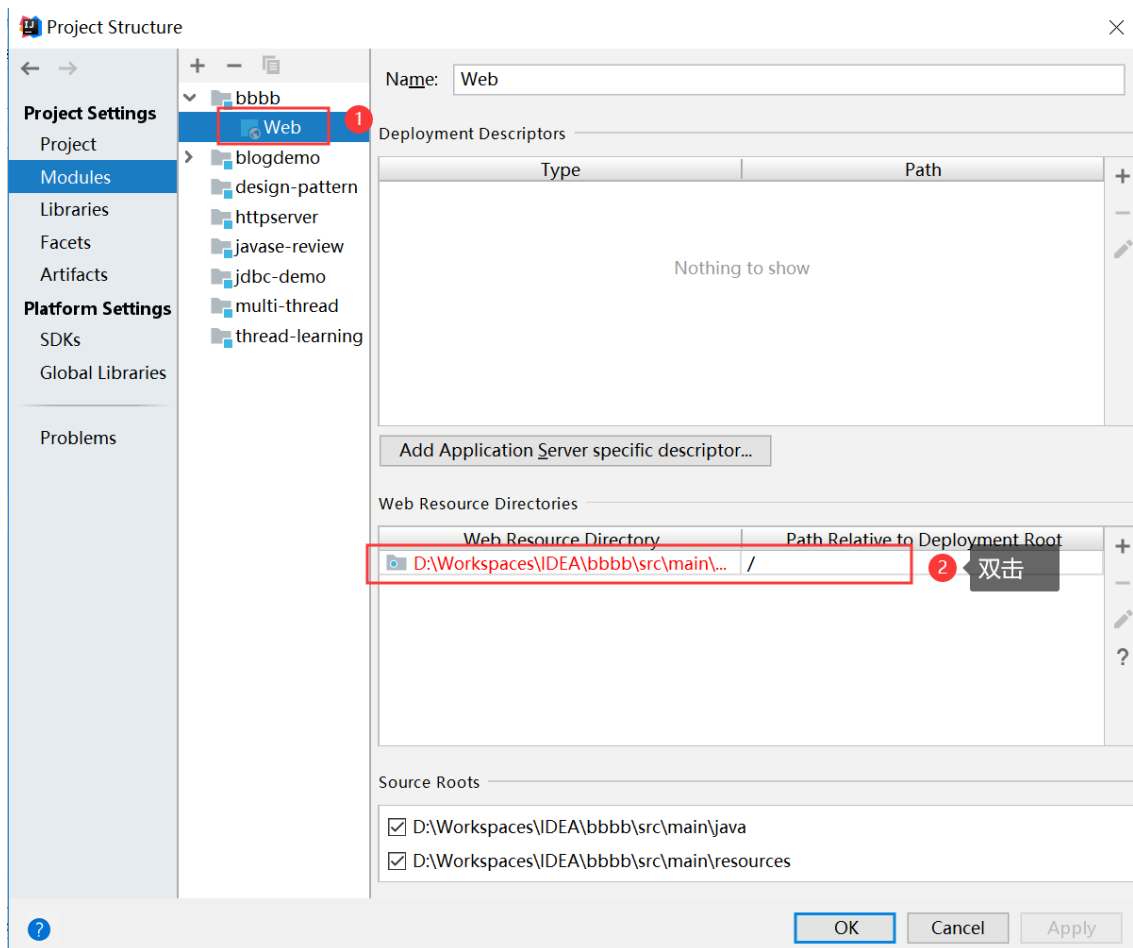


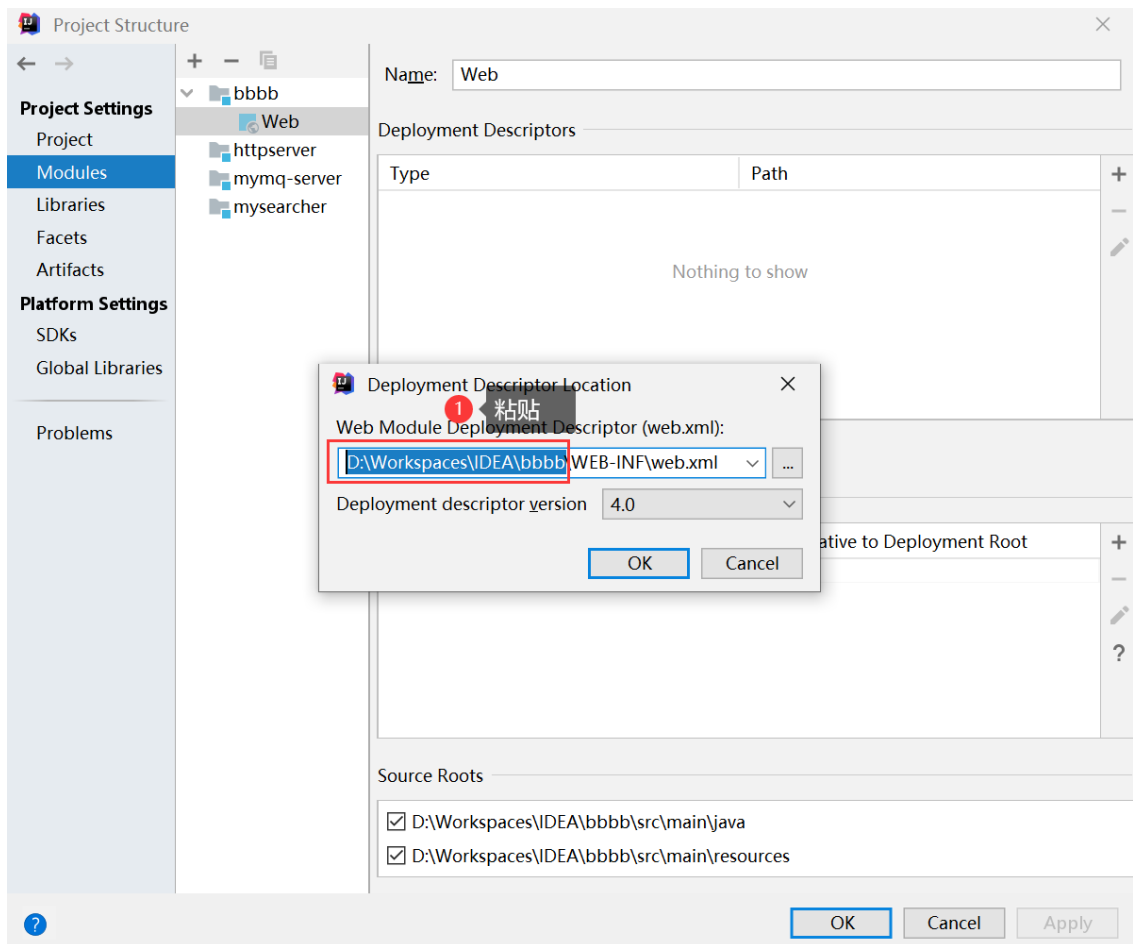
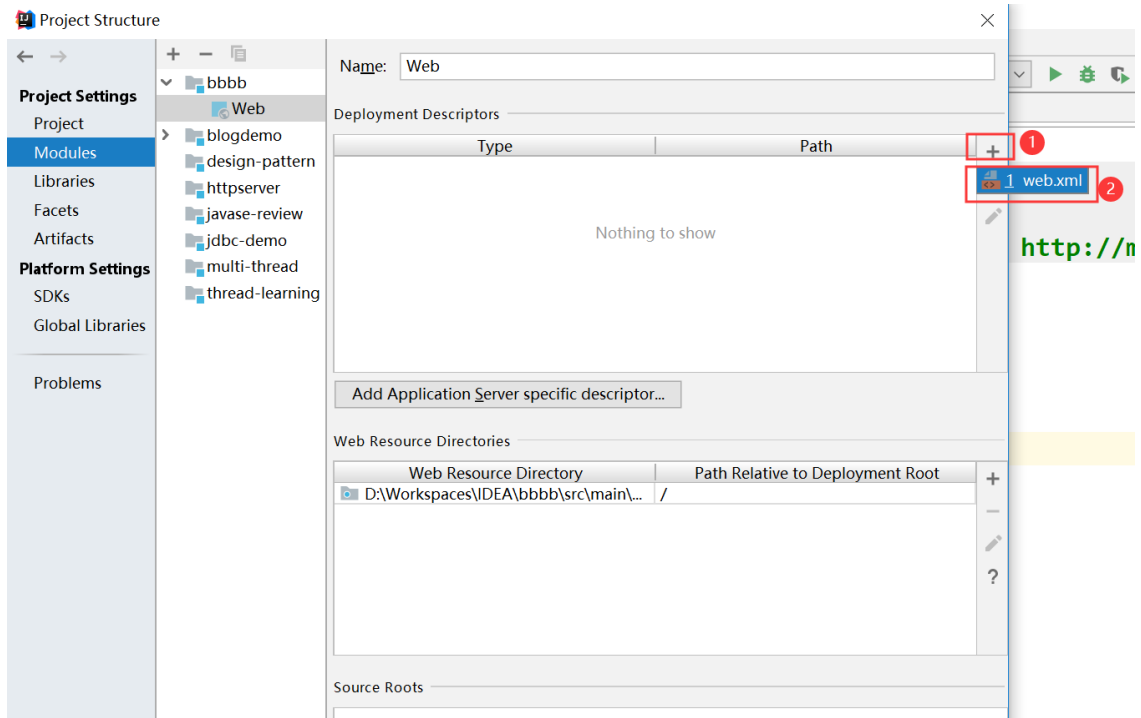


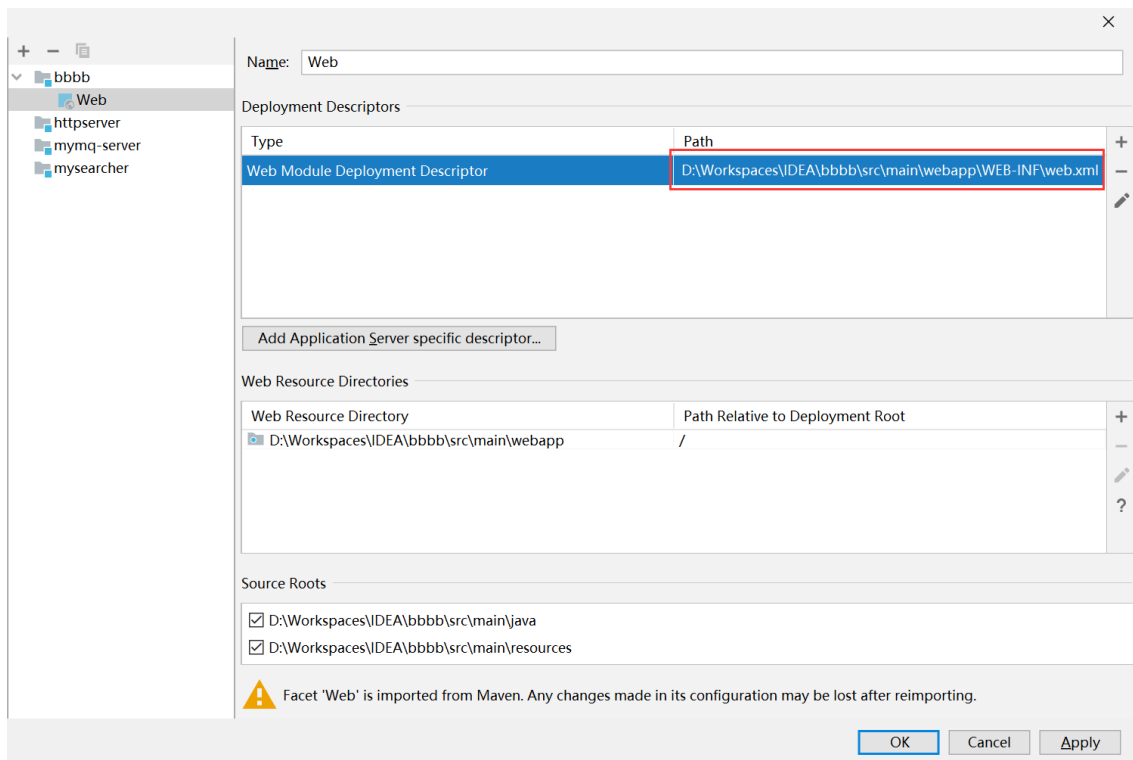


- 项目的web.xml生成









数据库设计

- 创建数据库

- 设计用户表

需要id、姓名、密码字段

- 设计文章表

需要id、文章标题、文字内容字段，且需要关联用户表，关系表现为1个用户多篇文章，所以在文章表建立外键user_id关联用户表的主键id

- 设计文章图片表

需要id，图片地址，且需要关联文章表，一篇文章对应多张图片，所以建立外键article_id关联文章表主键id

```
drop database if exists blogdemo;
create database blogdemo default charset utf8mb4;

use blogdemo;

-- -----
-- Table structure for user
-- -----
DROP TABLE IF EXISTS user;
CREATE TABLE user (
  id int(255) PRIMARY KEY AUTO_INCREMENT,
  name varchar(255) NOT NULL UNIQUE,
  password varchar(255) NOT NULL
);

-- -----
-- Table structure for article
-- -----
DROP TABLE IF EXISTS article;
CREATE TABLE article (
```

```

id int(11) PRIMARY KEY AUTO_INCREMENT,
title varchar(255) NOT NULL,
content mediumtext NOT NULL,
user_id int(11) NOT NULL,
CONSTRAINT fk_article_user_id FOREIGN KEY (user_id) REFERENCES user (id)
);

-- -----
-- Table structure for img
-- -----

DROP TABLE IF EXISTS img;
CREATE TABLE img (
  id int(11) PRIMARY KEY AUTO_INCREMENT,
  article_id int(11) NOT NULL,
  path varchar(255) NOT NULL,
  CONSTRAINT fk_img_article_id FOREIGN KEY (article_id) REFERENCES article
(id)
);

```

开发步骤

pom.xml配置

- web项目需要打包为war
- 引入依赖
- 使用jdk1.8版本来编译项目

```

<?xml version="1.0" encoding="UTF-8"?>
<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/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.bit</groupId>
  <artifactId>blogdemo</artifactId>
  <version>1.0-SNAPSHOT</version>
  <!-- web项目需要配置为war，表示打包为war文件 -->
  <packaging>war</packaging>

  <dependencies>
    <!-- MySQL数据库JDBC驱动包 -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>5.1.38</version>
    </dependency>
    <!-- jackson: 提供Java对象与JSON数据格式进行序列化
      及反序列化的支持 -->
    <dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-databind</artifactId>
      <version>2.8.9</version>
    </dependency>

    <!-- 使用官方的servlet依赖，并配置为scope=provided，表示在打包时不打入war包中 -->

```

```

<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>3.0.1</version>
  <scope>provided</scope>
</dependency>

<!-- 日志记录框架: logback -->
<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-classic</artifactId>
  <version>1.2.3</version>
</dependency>
</dependencies>

<build>
  <!-- 最后使用mvn package命令打包的文件名称 -->
  <finalName>${project.artifactId}</finalName>
  <plugins>
    <plugin>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.1</version>
      <configuration>
        <source>1.8</source>
        <target>1.8</target>
        <encoding>UTF-8</encoding>
      </configuration>
    </plugin>
  </plugins>
</build>
</project>

```

设计数据库实体类

- 用户类，对应数据库用户表
- 文章类，对应数据库文章表，且文章图片表不做单独使用，需要配合文章来使用，所以可以不创建文章图片类，而直接再文章类中建立关联

设计数据库连接工具类

- 使用DataSource作为数据库连接的创建方式
- 设计释放jdbc资源的方法

设计用户注册、登录功能

原理：需要先注册用户才能登陆访问，登录后需要建立用户的session信息，用来在访问文章新增等功能判断用户是否登录。

目标：注册后在数据库建立用户，登录后创建用户session数据，并跳转到新增文章页面

设计用户新增文章功能

原理：在新增文字页面中，访问新增文章接口，上传文章标题、内容和图片文件

目标：用户没有登录时，跳转到用户登录页面。登录后允许访问，新增文章成功，保存数据到文章表，文章图片表。图片文件需要上传到指定的目录下，数据库记录图片在服务器的访问地址。新增文章成功后，跳转到文章详情页面。

设计文章详情功能

原理：显示文章的信息，包括标题、具体内容，图片

目标：可以单独输入url访问，也可以从新增文章成功后跳转进入

设计文章列表功能

原理：显示所有的文章信息，以json类型的数据返回，并展示

目标：后台需要获取数据库中的所有文章数据，并序列化为json格式，并返回给客户端浏览器显示

设计文章列表统计访问量功能

原理：统计文章列表的访问次数，当每次访问文章列表接口时，次数加1

目标：需要保证在多线程环境下访问文章列表时，设置次数的线程安全性，在访问统计访问量接口后，显示总的访问量