



序号:

廣東工業大學

课 程 实 验

IDEA 构建基于 Maven 的 web 项目

课程名称 软件系统设计与体系结构

专业班级 17 软件工程 2 班

学 号 3117007142

学生姓名 陈悦演

指导教师 郝彦军

2020 年 7 月

目 录

一、实验目的.....	1
二、实验平台.....	1
三、实验前准备.....	1
四、实验内容.....	2
4.1 IntelliJ IDEA 安装.....	2
4.2 MySQL 安装.....	3
4.3 IDEA 创建 web 项目并编写代码.....	5
4.4 实现效果截图.....	10
(1) 个人博客系统首页.....	10
(2) 搜索查询功能实现.....	11
(3) 博主信息和博客分类信息展示.....	11
五、实验总结.....	12

一、实验目的

- 1.学习 IDEA 工具的使用，了解如何创建基于 maven 的 web 项目。
- 2.学习 mysql 数据库的使用，清楚如何与存储的数据进行交互。
- 3.学习利用 springboot 框架快速搭建开发环境，采用 maven 进行依赖管理，进行 web 项目开发。

二、实验平台

- 1.操作系统: windows 10, windows 7, windows XP。（以自身操作系统为准）

- 2.使用软件:

代码编写: IntelliJ IDEA 编辑器

数据库: MYSQL 数据库管理系统

建模工具: 在线网站 draw.io, 使用简单, 功能齐全

文档编写工具: Microsoft office Word 2019, Typora (markdown 语法编辑器)

项目管理工具: 采用 maven 对项目依赖进行管理

3. 开发语言与技术:

后台语言开发使用 Java 语言，前端语言使用 html、js、css 结合使用，使用了 SpringBoot+thymeleaf 完成了相关功能的开发，使用 MyBatis 框架连接数据库进行 Sql 语句的编写。

三、实验前准备

1. 翻阅相关资料，掌握 springboot 框架的基础知识和使用，采用 mvc 模式进行开发。
2. 翻阅相关资料，掌握 maven 对项目依赖的管理和配置。
3. 翻阅相关资料，掌握 MySQL 数据库的基础知识，以及如何使用 MyBatis 框架进行数据增删改查。
4. 安装必要的软件和开发工具。

四、实验内容

4.1 IntelliJ IDEA 安装

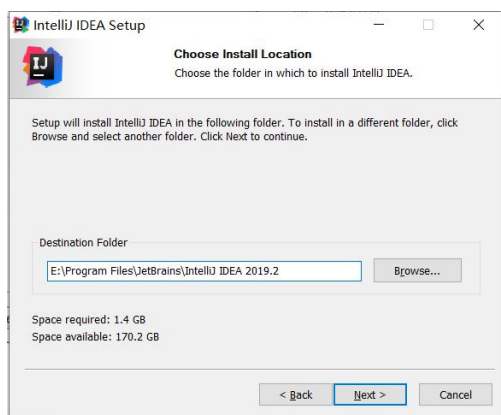
(1) 进入 [IntelliJ IDEA 官网](#)，根据自己的操作系统下载相应的安装包。



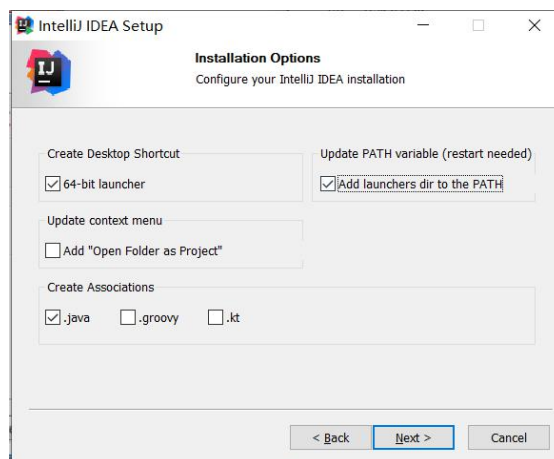
(2) 下载之后双击, 开始安装, 点击 next



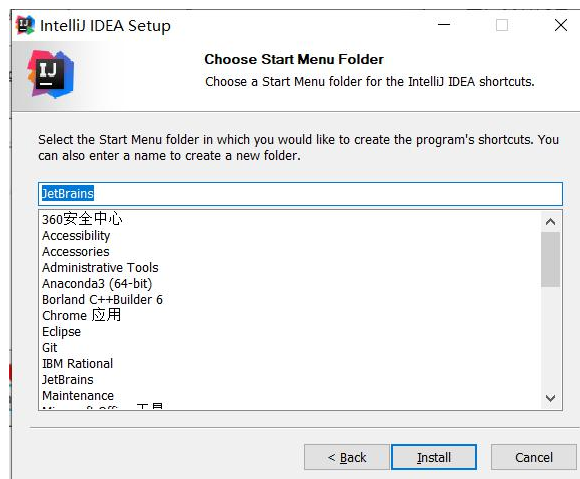
(3) 选择安装路径



(4) 选择一些配置属性, 添加变量配置到 PATH



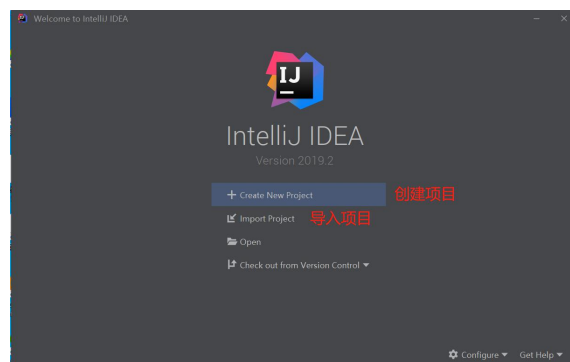
(5) 默认选择 JetBrains, 点击 install



(6) 等待安装完毕，勾选启动 IDEA，点击 finish。

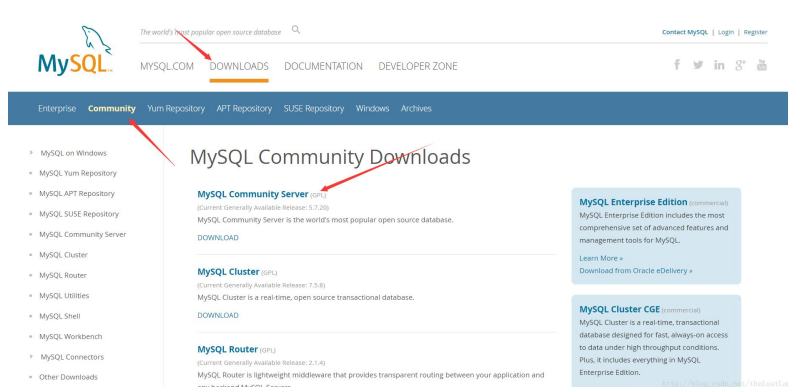


(7) 在进行一个软件激活，插件安装后会出现 IDEA 的软件界面，视为安装启动成功。

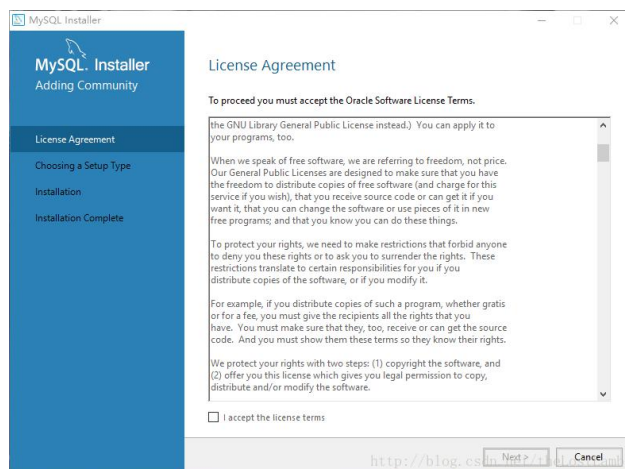


4.2 MySQL 安装

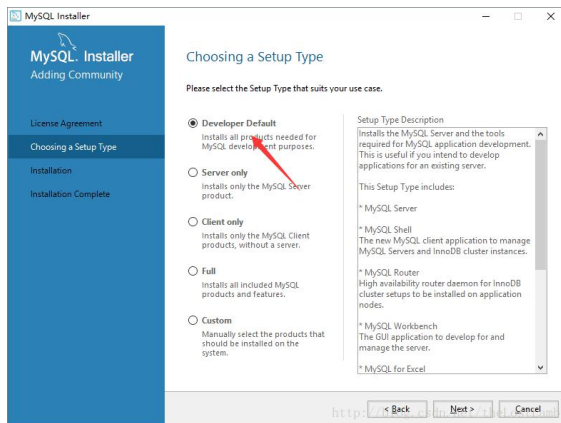
(1) 首先去 MySQL 官网浏览，根据操作系统下载相应版本的 MySQL。



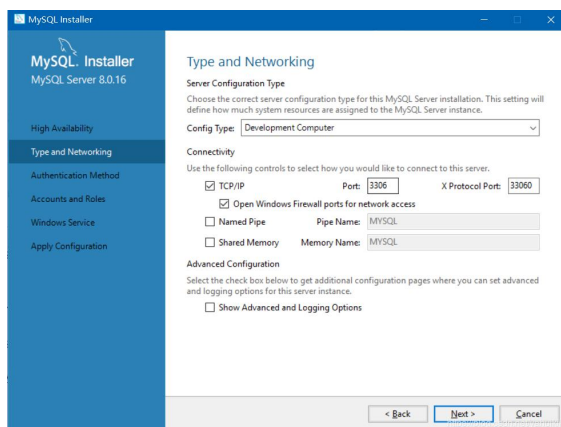
(2) 双击打开安装包，阅读软件安装许可证。



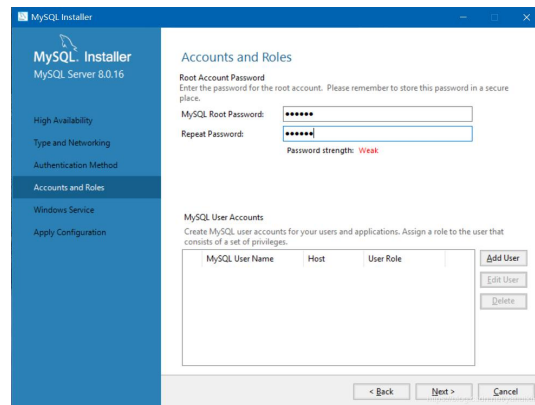
(3) 点击 **next** 进入下一步，选择所要安装的数据库模式



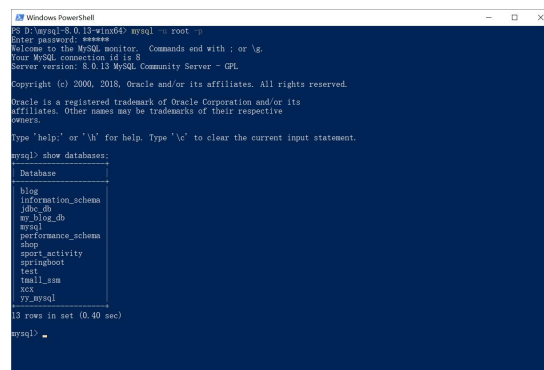
(4) **next** 继续进入下一步，选择默认的 Config Type 即可。



(5) 配置 **root** 用户密码，还可以添加其他用户。记住所配置的 **root** 用户密码。

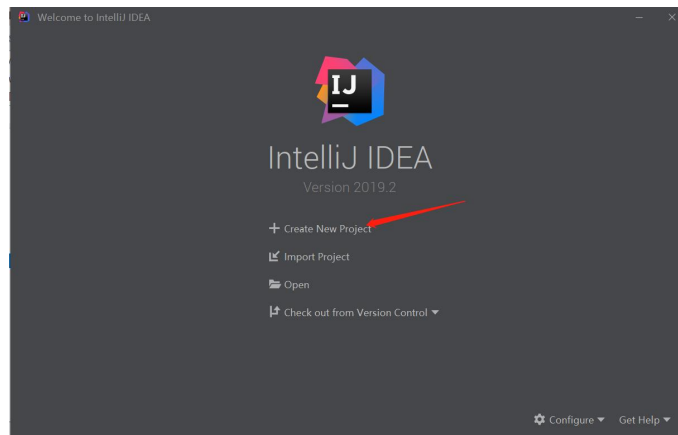


(6) 然后继续 **next**，选择安装的位置，直到所有都 **finish**。配置好 **mysql** 安装的 **bin** 目录到电脑环境变量中，输入账号密码后，使用相关指令就可以进入 **mysql** 操作界面。

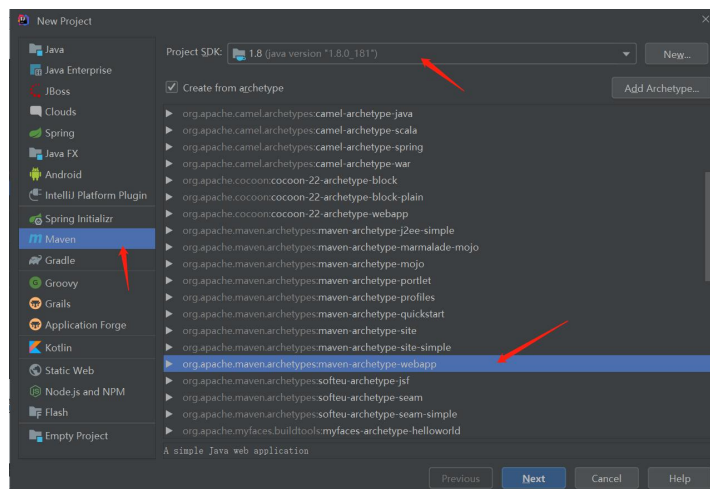


4.3 IDEA 创建 web 项目并编写代码

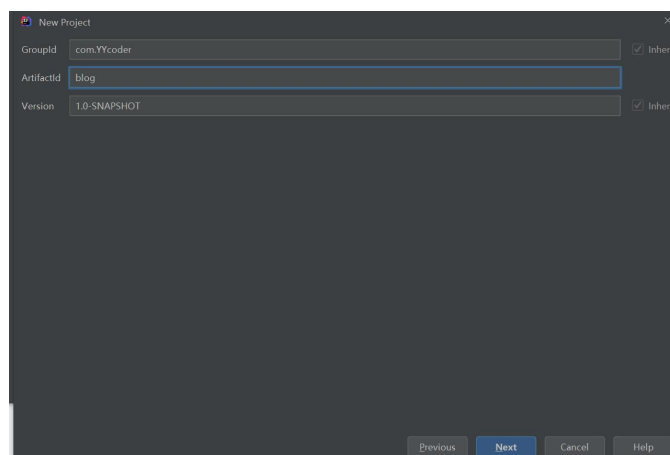
(1) 打开 IDEA 软件，进入操作界面，点击新建项目。



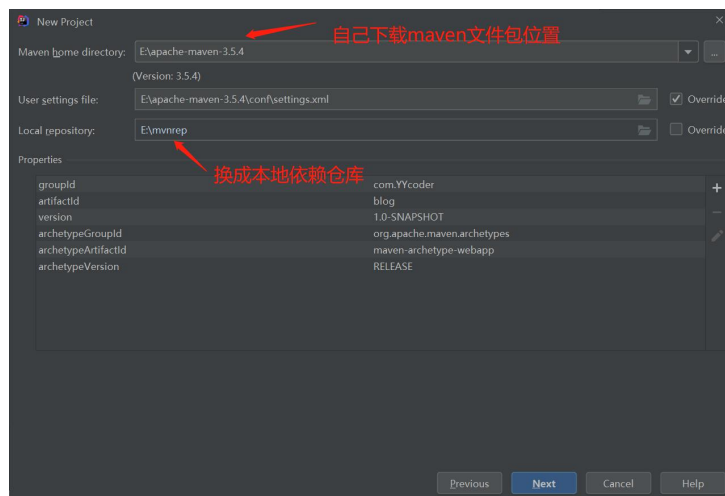
(2) 选择创建一个 maven 类型的项目，并选择安装好了的 JDK。



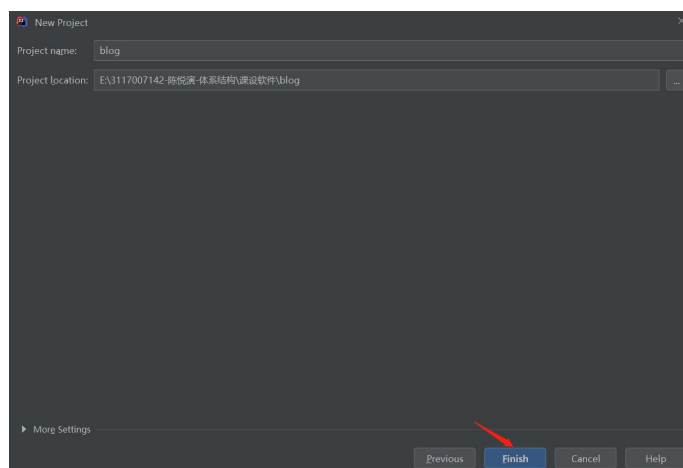
(3) 输入组织名称、模块名称、项目版本号等信息



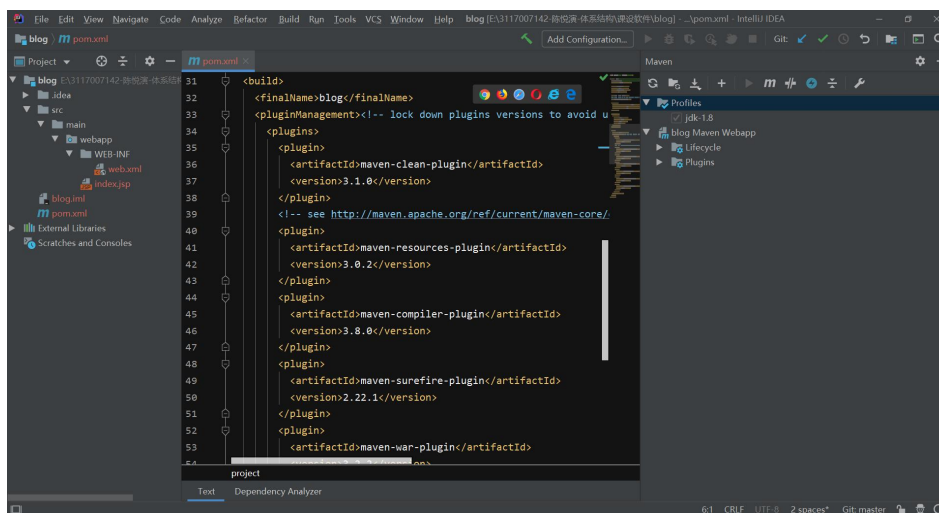
(4) 配置自己的 maven，以及相关的配置信息



(5) 点击 next 进入下一步操作，选择好安装路径，点击 finish。



(6) 创建完毕后打开就可以看到相关的 pom 文件和目录结构了。

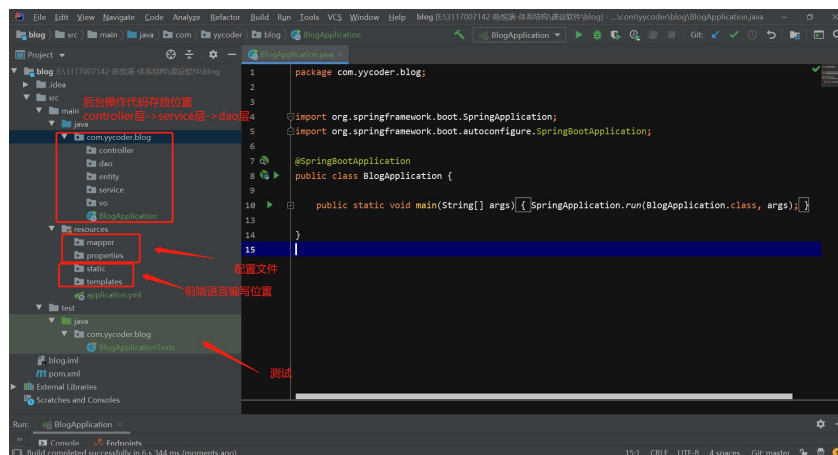


(7) 这样创建出来的 maven 项目，可以进行 web 项目的开发了，为了方便开发，我们引入 pom 依赖，将 maven 项目转变为 springboot 项目。

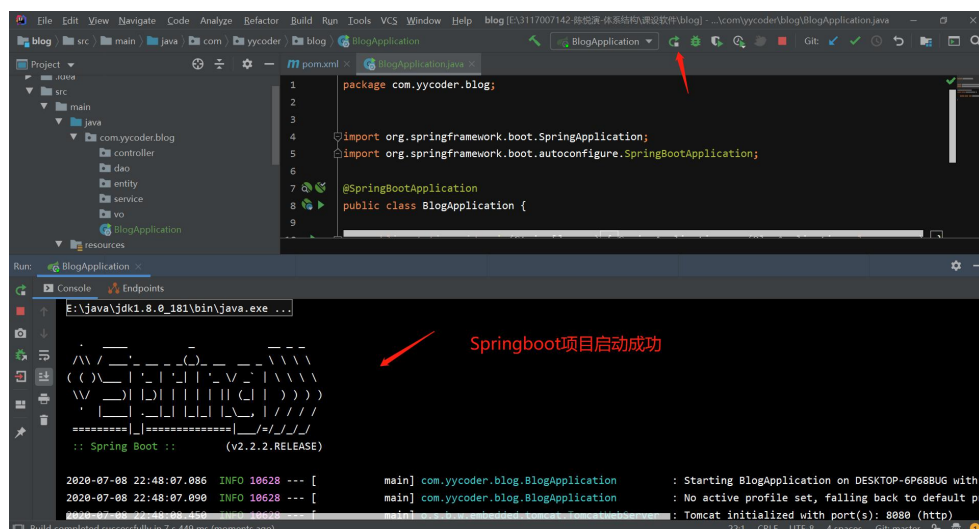
```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.2.2.RELEASE</version>
  <relativePath/>
</parent>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
</dependencies>
```

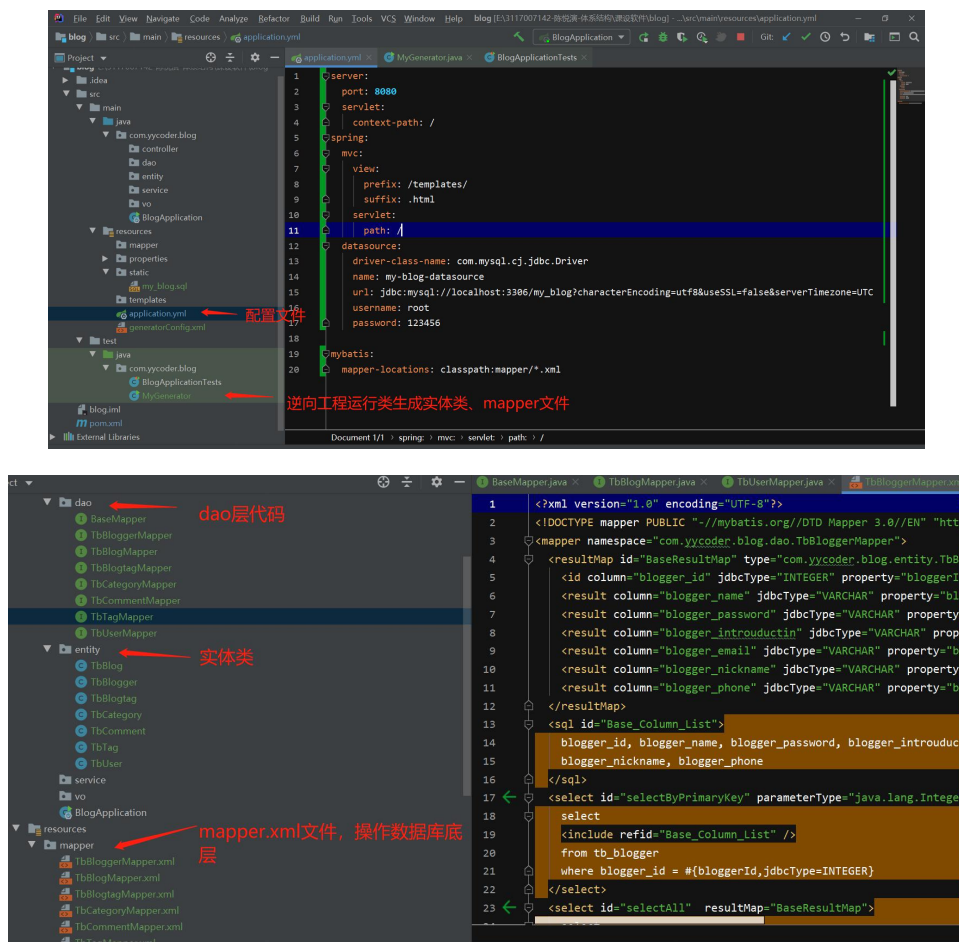
(8) 按照 mvc 模式对目录文件进行修改，让结构层次更加清晰。



(9) 启动项目成功，表示基于 maven 的 springboot 项目构建成功。



(10) 在 application.yml 进行数据库的配置，以及 mybatis 与数据库 mysql 的连接保持正常，提前创建好数据库的数据，通过反向代理方式生成数据的基本操作方法。



(11) 编写相应的 controller 代码、service 层代码以及前端代码，通过 controller 运用 springMvc 的访问方式与前端代码进行数据的获取与交互，实现整个 web 项目功能的编写实现。

Controller 层代码

```
@Controller
@RequestMapping("/blogHome")
public class HomeController {
    @Resource
    private TbBlogService tbBlogService; // 处理博客信息的逻辑层
    @Resource
    private TbBloggerService tbBloggerService; // 处理博主信息的逻辑层
    @Resource
    private TbCategoryService tbCategoryService; // 处理博客分类的逻辑层
    @Resource
    private TbCommentService tbCommentService; // 处理评论的逻辑层
```

```

@RequestMapping("/search")//实现搜索匹配
public String search(@RequestParam String keyWord, Model model) {
    keyWord="%" +keyWord+"%";
    List<BlogDetailVo> detailVoList = tbBlogService.queryAllBlogByTitle(keyWord);
    model.addAttribute("blogList", detailVoList);
    HashMap<String, Object> map = new HashMap<>();
    for(BlogDetailVo b:detailVoList) {
        map.put(b.getBlogId()+"", tbCommentService.getCommentCount(b.getBlogId()));
    }
    model.addAttribute("countList", map);
    if (detailVoList.isEmpty()) {
        return "blank";
    }
    return "content";//跳转到 content.html
}

@RequestMapping("/home")//博客首页
public String home(Model model) {
    TbBlogger blogger = tbBloggerService.selectAll();
    model.addAttribute("blogger", blogger);
    List<TbCategory> categoryList = tbCategoryService.selectAll();
    model.addAttribute("categoryList", categoryList);
    return "home";//跳转到主页面 home.html
}

@RequestMapping("/toContent")//博客数据的显示
public String toContent(Model model) {
    List<BlogDetailVo> tbBlogList = tbBlogService.queryAllBlog();
    model.addAttribute("blogList", tbBlogList);
    HashMap<String, Object> map = new HashMap<>();
    for(BlogDetailVo b:tbBlogList) {
        map.put(b.getBlogId()+"", tbCommentService.getCommentCount(b.getBlogId()));
    }
    model.addAttribute("countList", map);
    return "content";//跳转到 content.html
}
}

```

service 层实现（主要方法）

```

public class TbBlogServiceImpl implements TbBlogService { //获取博客信息
    @Autowired(required = true)
    private TbBlogMapper tbBlogMapper; //有关博客信息的操作 dao 层
    @Autowired
    private TbTagMapper tbTagMapper; //有关博客分类的 dao 层
    @Override
    public List<BlogDetailVo> queryAllBlogByTitle(String keyWord) { //按博客标题匹
        配搜索
    }
}

```

```

        List<BlogDetailVo> list=tbBlogMapper.queryAllBlogByTitle(keyWord);
        for(BlogDetailVo bd:list){
            bd.setTagList(tbTagMapper.queryAllTagByBlogId(bd.getBlogId()));
        }
        return list;
    }
    @Override
    public List<BlogDetailVo> queryAllBlog() { //查询所有的博客信息
        List<BlogDetailVo> list=tbBlogMapper.queryAllBlog();
        for(int i=0;i<list.size();i++){
            list.get(i).setTagList(tbTagMapper.queryAllTagByBlogId(list.get(i).getBlogId()));
        }
        return tbBlogMapper.queryAllBlog();
    }
}

public class TbBlogCategoryServiceImpl implements TbCategoryService {
    @Autowired
    private TbCategoryMapper tbCategoryMapper; //处理分类数据的 dao 层
    @Override
    public List<TbCategory> selectAll() {
        return tbCategoryMapper.selectAll(); //调用 dao 代码实现博客分类查询
    }
}

```

(12) 编写相关前端 HTML 代码，等所有代码编写完毕无误后，点击运行项目，打开浏览器查看效果。

4.4 实现效果截图

(1) 个人博客系统首页

项目正常运行后，由于配置的是 8080 端口开启，所以访问[博客首页](#)即可进入系统，查看到博主发布的相关博客。

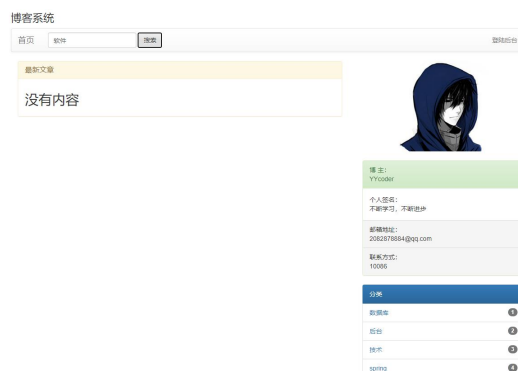


(2) 搜索查询功能实现

游客还可以在搜索框输入关键字，如输入“学习”点击查询，数据库中有含该关键字的博客标题，就展示相关的博客。



若数据库中没有相关的数据，则展示没有内容，表示没有对应包含该关键字的博客信息。



(3) 博主信息和博客分类信息展示

可以看到我们在访问博客首页的时候，也可以看到博主个人的信息，在该系统下所有的博客信息都是由博主所发布的。除此之外，为了方便游客查看，列举了博客分类列表，可以根据分类定位到特定类型的博客。



五、实验总结

本次的实验内容是借助 IDEA 编辑器，构建了一个基于 Maven 的 web 项目，经过 pom.xml 依赖引入以及目录的调整搭建后，最后将项目改为 springboot 项目，运用 springboot 进行开发，可以减少以往以 SSM 框架开发时一些繁琐的配置，借助 MVC 开发模式，进行代码的编写，最后结合 thymeleaf 模板语言编写前端相关语言，成功实现了一个页面的正常访问，和后台数据库数据之间的交互。

所编写的功能是自己选定的课题题目--个人博客系统中的，简单实现了博客系统首页、博主信息展示、博客搜索功能实现，理解了 MVC 模式下的代码编写，熟悉了 IDEA 软件的使用，也对 mysql 数据库的使用更加熟练。在本次实验中，我收获了很多，不仅学习到了 IDEA 工具的使用，了解如何创建基于 maven 的 web 项目，还学习了 mysql 数据库的使用，清楚如何与存储的数据进行交互。以及学习如何利用 springboot 框架快速搭建开发环境，采用 maven 进行依赖管理，进行 web 项目开发。对整个 web 项目开发的各个环境更加明白，并也尝试着使用 git 来进行项目的管理。