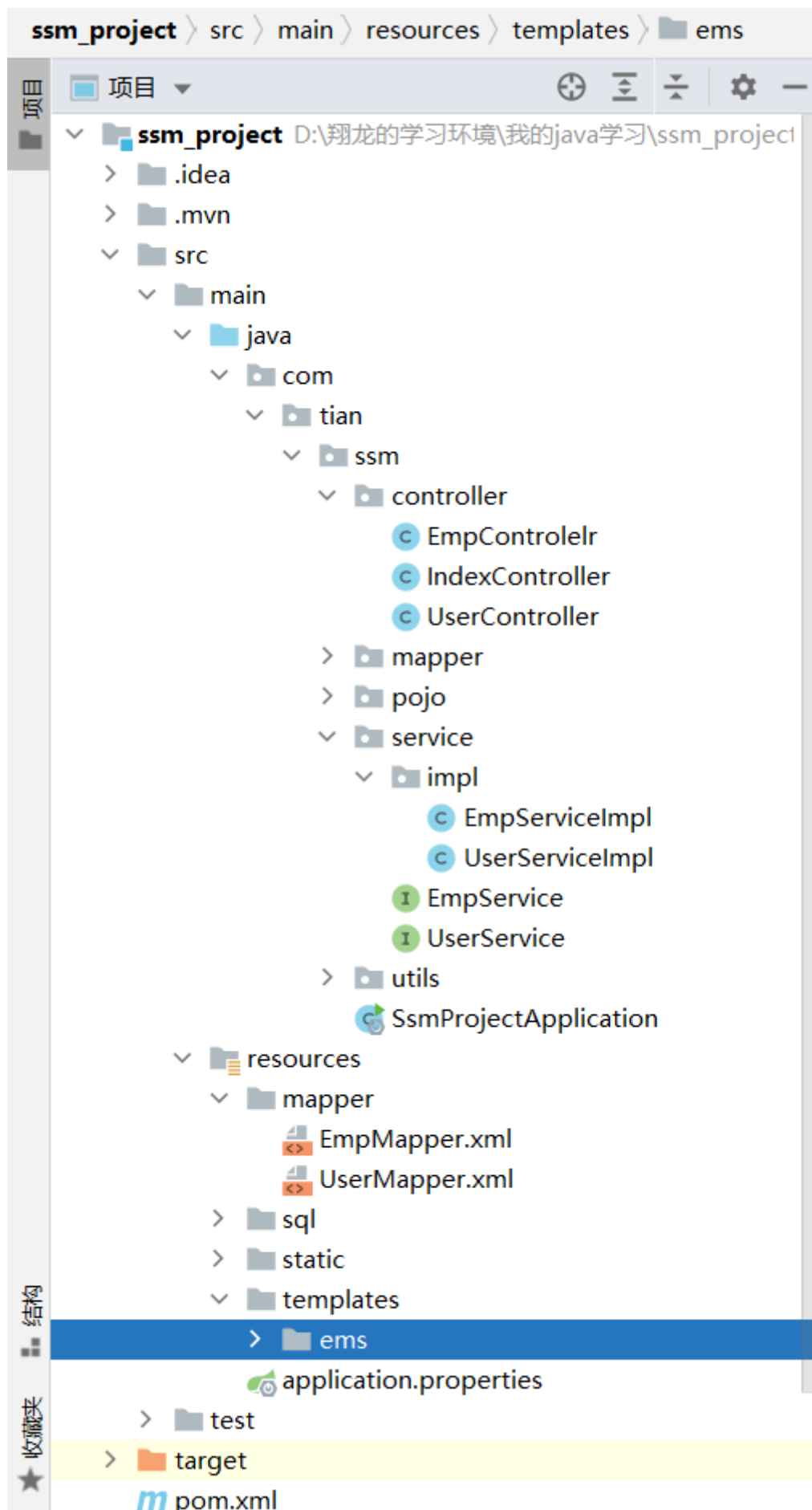


完整的企业级小型SSM (spring+springboot+mybatis)项目

1、项目结构图



2、建立一个springboot项目并导入依赖！

以下为本次项目所需要的依赖

```
<?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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.2.7.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.tian</groupId>
  <artifactId>ssm</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>ssm_project</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>1.8</java.version>
  </properties>
  <dependencies>

    <!-- thymeleaf语法，主要获取后台的数据放到前端-->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>

    <!-- spring-boot-starter-web依赖-->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!-- mybatis框架-->
    <dependency>
      <groupId>org.mybatis.spring.boot</groupId>
      <artifactId>mybatis-spring-boot-starter</artifactId>
      <version>2.2.0</version>
    </dependency>

    <!-- mysql数据库依赖-->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <scope>runtime</scope>
    </dependency>

    <!-- Lombok依赖-->
    <dependency>
      <groupId>org.projectlombok</groupId>
      <artifactId>lombok</artifactId>
      <optional>true</optional>
```

```

        </dependency>

        <!--spring-boot-starter-test依赖-->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
                <configuration>
                    <excludes>
                        <exclude>
                            <groupId>org.projectlombok</groupId>
                            <artifactId>lombok</artifactId>
                        </exclude>
                    </excludes>
                </configuration>
            </plugin>
        </plugins>
    </build>

</project>

```

3、配置application.properties文件

```

server.port=8888
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/my_learn?
useUnicode=true&characterEncoding=UTF-8&useSSL=true&serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=123456

mybatis.mapper-locations=classpath:mapper/*.xml
mybatis.type-aliases-package=com.tian.ssm.pojo
mybatis.configuration.log-impl=org.apache.ibatis.logging.stdout.StdOutImpl

spring.resources.static-locations=classpath:/templates/,classpath:/static/
spring.thymeleaf.cache=false

```

4、创建数据库(sql)

```

//user表，用于登录和注册
CREATE table t_user(
    id VARCHAR(40) primary key,
    username VARCHAR(40),
    realname VARCHAR(40),
    password VARCHAR(40),

```

```

        sex VARCHAR(8)
    );

//emp用于进行CRUD的操作
CREATE table t_emp(
    id VARCHAR(40) primary key,
    name VARCHAR(60),
    salary double(7,2),
    age int(3),
    bir date
);

```

5、开发实体类(pojo)

use类

```

package com.tian.ssm.pojo;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.ToString;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class User {
    private String id;
    private String username;
    private String realname;
    private String password;
    private String sex;
}

```

emp类

```

package com.tian.ssm.pojo;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.ToString;

import java.util.Date;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class Emp {
    private String id;
    private String name;
    private Double salary;
    private Integer age;
    private Date bir;
}

```

```
}
```

6、开发持久层（Mapper）

usermapper层

```
package com.tian.ssm.mapper;

import com.tian.ssm.pojo.User;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Param;

@Mapper
public interface UserMapper {
    //注册用户
    void save(User user);

    //在mybatis中传递多个参数需要参数的绑定
    User login(@Param("username") String username, @Param("password") String password);
}
```

empmapper层

```
package com.tian.ssm.mapper;

import com.tian.ssm.pojo.Emp;
import org.apache.ibatis.annotations.Mapper;

import java.util.List;

@Mapper
public interface EmpMapper {

    List<Emp> findAll();

    void save(Emp emp);

    void delete(String id);

    Emp find(String id);

    void update(Emp emp);
}
```

7、开发对应的Mapper.xml文件（mybatis框架）

UserMapper.xml文件

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd" >
<mapper namespace="com.tian.ssm.mapper.UserMapper">
```

```

<!-- 注册-->
<insert id="save" parameterType="User">
    insert into t_user values({id},{username},{realname},{password},{sex})
</insert>

<!--登录-->
<select id="login" resultType="User">
    select id,username,realname,password,sex from t_user
    where username={username} and password={password}
</select>
</mapper>

```

EmpMapper.xml文件

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd" >
<mapper namespace="com.tian.ssm.mapper.EmpMapper">

    <!-- 查询所有-->
    <select id="findAll" resultType="Emp">

        select id,name,salary,age,bir from t_emp
    </select>

    <!-- 添加员工-->
    <insert id="save" parameterType="Emp">
        insert into t_emp values ({id},{name},{salary},{age},{bir})
    </insert>

    <!-- 删除员工-->
    <delete id="delete" parameterType="String">
        delete from t_emp where id={id}
    </delete>

    <!-- 根据ID查询员工-->
    <select id="find" parameterType="String" resultType="Emp">
        select id,name,salary,age,bir from t_emp where id = {id}
    </select>

    <!-- 修改员工-->
    <update id="update" parameterType="Emp">
        update t_emp set name={name},salary={salary},age={age},bir={bir}
        where id={id}
    </update>
</mapper>

```

8、开发业务层 (service)

UserService

```

package com.tian.ssm.service;

import com.tian.ssm.pojo.User;

```

```

public interface UserService {

    /**
     * 注册
     * @param user 用户
     */
    void register(User user);

    /**
     * 登录
     * @param username 用户名
     * @param password 密码
     * @return
     */
    User login(String username, String password);
}

```

EmpService

```

package com.tian.ssm.service;

import com.tian.ssm.pojo.Emp;

import java.util.List;

public interface EmpService {
    //查询全部用户
    List<Emp> findAll();

    //添加用户
    void save(Emp emp);

    //删除用户
    void delete(String id);

    //查找指定用户
    Emp find(String id);

    //更新用户
    void update(Emp emp);
}

```

9、开发业务实现类 (servecimpl)

UserServiceImpl

```

package com.tian.ssm.service.impl;

import com.tian.ssm.mapper.UserMapper;
import com.tian.ssm.pojo.User;
import com.tian.ssm.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

```



```

import java.util.UUID;

@Service
@Transactional
public class UserServiceImpl implements UserService {
    @Autowired
    private UserMapper userMapper;

    @Override
    public void register(User user) {
        user.setId(UUID.randomUUID().toString());
        userMapper.save(user);
    }

    @Override
    public User login(String username, String password) {
        return userMapper.login(username, password);
    }
}

```

EmpServiceImpl

```

package com.tian.ssm.service.impl;

import com.tian.ssm.mapper.EmpMapper;
import com.tian.ssm.pojo.Emp;
import com.tian.ssm.service.EmpService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.UUID;

@Service
@Transactional
public class EmpServiceImpl implements EmpService {

    @Autowired
    private EmpMapper empMapper;

    @Override
    @Transactional(propagation = Propagation.SUPPORTS)
    public List<Emp> findAll() {
        return empMapper.findAll();
    }

    @Override
    public void save(Emp emp) {
        emp.setId(UUID.randomUUID().toString());
        empMapper.save(emp);
    }

    @Override

```

```

    public void delete(String id) {
        empMapper.delete(id);
    }

    @Override
    @Transactional(propagation = Propagation.SUPPORTS)
    public Emp find(String id) {
        return empMapper.find(id);
    }

    @Override
    public void update(Emp emp) {
        empMapper.update(emp);
    }
}

```

10、开发视图跳转层 (controller)

UserController

```

package com.tian.ssm.controller;

import com.tian.ssm.pojo.User;
import com.tian.ssm.service.UserService;
import com.tian.ssm.utils.ValidateImageCodeUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.Mapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

import javax.imageio.ImageIO;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import java.awt.image.BufferedImage;
import java.io.IOException;

/**
 * @author 田智龙
 */
@Controller
@RequestMapping("/user")
public class UserController {

    @Autowired
    private UserService userService;

    /**
     * 注册方法
     *
     * @param user
     * @param code

```

```

    * @param session
    * @return
    */
@RequestMapping("/register")
public String register(User user, String code, HttpSession session) {
    String sessionCode = (String) session.getAttribute("code");
    if (sessionCode.equalsIgnoreCase(code)) {
        userService.register(user);
        return "redirect:/index";
    } else {
        return "redirect:/toRegister";
    }
}

/**
 * 生成验证码
 *
 * @param session
 * @param response
 * @throws IOException
 */
@GetMapping("/code")
public void getVerification(HttpSession session, HttpServletResponse
response) throws IOException {
    //生成验证码
    String securityCode = ValidateImageCodeUtils.getSecurityCode();
    BufferedImage image = ValidateImageCodeUtils.createImage(securityCode);
    //存入session中
    session.setAttribute("code", securityCode);
    //响应图片
    ServletOutputStream os = response.getOutputStream();
    ImageIO.write(image, "png", os);
}

@PostMapping(value = "/login")
public String login(String username, String password) {
    //查询数据库登录系统
    User login = userService.login(username, password);
    if (login != null) {
        return "redirect:/emp/findAll";
    } else {
        return "redirect:/index";
    }
}
}
}

```

IndexController

```

package com.tian.ssm.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

```

```

/**
 * @author huanm
 * <p>
 * thymeleaf所有操作都需要通过controller实现跳转
 */
@Controller
public class IndexController {

    @GetMapping(value = "/index")
    public String toIndex() {
        return "ems/login";
    }

    @GetMapping(value = "/toRegister")
    public String toRegister() {
        return "ems/regist";
    }

    @GetMapping(value = "toSave")
    public String toSave() {
        return "ems/addEmp";
    }

}

```

EmpController

```

package com.tian.ssm.controller;

import com.tian.ssm.pojo.Emp;
import com.tian.ssm.service.EmpService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.Mapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

import java.util.List;

@Controller
@RequestMapping(value = "/emp")
public class EmpController {

    @Autowired
    private EmpService empService;

    @GetMapping(value = "/findAll")
    //find和findAll不是重定向，直接到跳转页面，并且直接跳转classpath下template自带/
    //重定向的话不是从template下，是需要加/的
    //查找不是重定向，不加/；重定向需要加/
    public String findAll(Model model) {
        List<Emp> allList = empService.findAll();
        model.addAttribute("emps", allList);
    }
}

```

```

        return "ems/emplist";
    }

    @GetMapping(value = "/save")
    public String save(Emp emp) {
        empService.save(emp);
        return "redirect:/emp/findAll";
    }

    @GetMapping(value = "/delete")
    public String delete(String id) {
        empService.delete(id);
        return "redirect:/emp/findAll";
    }

    @GetMapping(value = "/find")
    public String find(String id, Model model) {
        Emp emp = empService.find(id);
        model.addAttribute("emp", emp);
        return "ems/updateEmp";
    }

    @PostMapping(value = "update")
    public String update(Emp emp) {
        empService.update(emp);
        return "redirect:/emp/findAll";
    }
}

```

11、导入静态资源以及对应的页面

这里我就只展示页面吧！

addEmp.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>add Emp</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" type="text/css"
        href="css/style.css"/>
</head>

<body>
<div id="wrap">
    <div id="top_content">
        <div id="header">
            <div id="righthead">
                <p>
                    2020/5/1
                <br/>
            </p>
        </div>
    </div>

```

```

        <div id="topheader">
            <h1 id="title">
                <a href="#">添加页面</a>
            </h1>
        </div>
        <div id="navigation">
        </div>
    </div>
    <div id="content">
        <p id="whereami">
        </p>
        <h1>
            填写以下信息完成添加
        </h1>
        <form th:action="@{/emp/save}" method="get">
            <table cellpadding="0" cellspacing="0" border="0"
                class="form_table">
                <tr>
                    <td valign="middle" align="right">
                        姓名:
                    </td>
                    <td valign="middle" align="left">
                        <input type="text" class="inputgri" name="name"/>
                    </td>
                </tr>
                <tr>
                    <td valign="middle" align="right">
                        薪水:
                    </td>
                    <td valign="middle" align="left">
                        <input type="text" class="inputgri" name="salary"/>
                    </td>
                </tr>
                <tr>
                    <td valign="middle" align="right">
                        年龄:
                    </td>
                    <td valign="middle" align="left">
                        <input type="text" class="inputgri" name="age"/>
                    </td>
                </tr>
                <tr>
                    <td valign="middle" align="right">
                        生日:
                    </td>
                    <td valign="middle" align="left">
                        <input type="text" class="inputgri" name="bir"/>
                    </td>
                </tr>
            </table>
            <p>
                <input type="submit" class="button" value="确认添加"/>
            </p>
        </form>
    </div>
</div>
<div id="footer">
    <div id="footer_bg">

```

```
2455404279@qq.com
    </div>
  </div>
</div>
</body>
</html>
```

emplist.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>emplist</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <link rel="stylesheet" type="text/css" th:href="@{/css/style.css}"/>
</head>
<body>
<div id="wrap">
  <div id="top_content">
    <div id="header">
      <div id="righthead">
        <p>
          2020/5/1
        <br/>
        </p>
      </div>
      <div id="topheader">
        <h1 id="title">
          <a href="#">主界面</a>
        </h1>
      </div>
      <div id="navigation">
      </div>
    </div>
    <div id="content">
      <p id="whereami">
      </p>
      <h1>
        welcome!
      </h1>
      <table class="table">
        <tr class="table_header">
          <td>
            ID
          </td>
          <td>
            Name
          </td>
          <td>
            Salary
          </td>
          <td>
            Age
          </td>
        </tr>
      </table>
```

```

        Birth
    </td>
    <td>
        Operation
    </td>
</tr>
<tr class="row1" th:each="emp:${emps}">
    <td>
        <span th:text="${emp.id}"></span>
    </td>
    <td>
        <span th:text="${emp.name}"></span>
    </td>
    <td>
        <span th:text="${emp.salary}"></span>
    </td>
    <td>
        <span th:text="${emp.age}"></span>
    </td>
    <td>
        <!--日期格式化-->
        <span th:text="${#dates.format(emp.bir,'yyyy-MM-dd')}">
</span>
    </td>

    <td>
        <a th:href="@{/emp/delete(id=${emp.id})}">删除信息
</a>&nbsp;   
        <a th:href="@{/emp/find(id=${emp.id})}">修改信息</a>
    </td>
</tr>

</table>
<p>
    <input type="button" class="button" value="添加员工"
onclick="location.href='/toSave'"/>
</p>
</div>
</div>
<div id="footer">
    <div id="footer_bg">
        Tian Zhilong
    </div>
</div>
</div>
</body>
</html>

```

login.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>login</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

```



```

        <div id="footer">
            <div id="footer_bg">
                2455404279@qq.com
            </div>
        </div>
    </div>
</div>
</body>
</html>

```

regist.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>regist</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" type="text/css" href="css/style.css"/>
</head>
<body>
<div id="wrap">
    <div id="top_content">
        <div id="header">
            <div id="righthead">
                <p>
                    2020/5/1
                <br/>
            </p>
        </div>
        <div id="topheader">
            <h1 id="title">
                <a href="#">欢迎注册</a>
            </h1>
        </div>
        <div id="navigation">
        </div>
    </div>
    <div id="content">
        <p id="whereami">
        </p>
        <h1>
            注册
        </h1>
        <form th:action="@{/user/register}" method="post">
            <table cellpadding="0" cellspacing="0" border="0"
                class="form_table">
                <tr>
                    <td valign="middle" align="right">
                        用户名:
                    </td>
                    <td valign="middle" align="left">
                        <input type="text" class="inputgri
name="username"/>
                    </td>
                </tr>
            </table>
        </form>
    </div>

```

```

        <td valign="middle" align="right">
            真实姓名:
        </td>
        <td valign="middle" align="left">
            <input type="text" class="inputgri"
name="realname"/>
        </td>
    </tr>
    <tr>
        <td valign="middle" align="right">
            密码:
        </td>
        <td valign="middle" align="left">
            <input type="password" class="inputgri"
name="password"/>
        </td>
    </tr>
    <tr>
        <td valign="middle" align="right">
            性别:
        </td>
        <td valign="middle" align="left">
            男
            <input type="radio" class="inputgri" name="sex"
value="男" checked="checked"/>
            女
            <input type="radio" class="inputgri" name="sex"
value="女"/>
        </td>
    </tr>
    <tr>
        <td valign="middle" align="right">
            验证码:
            
            <a href="javascript:;"
                onclick="document.getElementById('num').src =
'/ems/user/code?'+(new Date()).getTime()">换一张</a>
        </td>
        <td valign="middle" align="left">
            <input type="text" class="inputgri" name="code"/>
        </td>
    </tr>
</table>
<p>
    <input type="submit" class="button" value="提交 &raquo;"/>
</p>
</form>
</div>
</div>
<div id="footer">
    <div id="footer_bg">
        @田智龙
    </div>
</div>
</div>
</body>
</html>

```

updateEmp.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html xmlns:th="www.thymeleaf.org">
<head>
    <title>update Emp</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" type="text/css"
        th:href="@{/css/style.css}"/>
</head>

<body>
<div id="wrap">
    <div id="top_content">
        <div id="header">
            <div id="righthheader">
                <p>
                    2020/5/1
                <br/>
            </p>
        </div>
        <div id="topheader">
            <h1 id="title">
                <a href="#">员工修改</a>
            </h1>
        </div>
        <div id="navigation">
        </div>
    </div>
    <div id="content">
        <p id="whereami">
        </p>
        <h1>
            请修改员工信息:
        </h1>
        <form th:action="@{/emp/update}" method="post">
            <table cellpadding="0" cellspacing="0" border="0"
                class="form_table">
                <tr>
                    <td valign="middle" align="right">
                        id:
                    </td>
                    <td valign="middle" align="left">
                        <span th:text="${emp.id}"></span>
                        <input type="hidden" th:value="${emp.id}" name="id">
                    </td>
                </tr>
                <tr>
                    <td valign="middle" align="right">
                        姓名:
                    </td>
                    <td valign="middle" align="left">
                        <input type="text" class="inputgri" name="name"
th:value="${emp.name}"/>
                    </td>
                </tr>
            </table>
        </form>
    </div>
</div>
</body>
</html>
```

```

        </td>
      </tr>
      <tr>
        <td valign="middle" align="right">
          薪水:
        </td>
        <td valign="middle" align="left">
          <input type="text" class="inputgri" name="salary"
th:value="\${emp.salary}"/>
        </td>
      </tr>
      <tr>
        <td valign="middle" align="right">
          年龄:
        </td>
        <td valign="middle" align="left">
          <input type="text" class="inputgri" name="age"
th:value="\${emp.age}"/>
        </td>
      </tr>
      <tr>
        <td valign="middle" align="right">
          生日:
        </td>
        <td valign="middle" align="left">
          <input type="text" class="inputgri" name="bir"
th:value="\${#dates.format(emp.bir,'yyyy/MM/dd')}"/>
        </td>
      </tr>
    </table>
    <p>
      <input type="submit" class="button" value="Confirm"/>
    </p>
  </form>
</div>
</div>
<div id="footer">
  <div id="footer_bg">
    2455404279@qq.com
  </div>
</div>
</div>
</body>
</html>

```

对应，整个项目就开发完成了！

12、运行并测试项目

启动项目访问localhost:8888/index



点击注册



用户名和密码进行登录



然后就是普通的操作！

看了这么久了，是时候来自己动手做一下了吧！

QQ邮箱: 2455404279@qq.com

作者: SmartDragon

欢迎来进行交流!