上面的生成验证码的类放在工具包utils中就行

@Controller

@RequestMapping("code")

public class CodeController {

@RequestMapping(value="getCode")

public void getCode(@RequestParam(value = "time") String time, HttpServletRequest request, HttpServletResponse response) {

//这些数据用来调控验证吗的大小个数等信息

ValidateCode code = new ValidateCode(60, 20, 4, 30, 25, "validateCode");

code.getCode(request, response);

}

}

提交后台

@RequestMapping("login")

public String login(String username, String password, String path, HttpSession session, RedirectAttributes attributes){

String validateCode = (String)session.getAttribute("validateCode");

session.removeAttribute("validateCode");

if(!validateCode.equalsIgnoreCase(path)){

attributes.addFlashAttribute("errorMsg","验证码错误");

return "redirect:/login";

}

//如果验证码正确，验证用户名和密码

Employee emp = empService.login(username,password);

if(emp != null){

session.setAttribute("loginUser",emp);

return "redirect:/index.jsp";

}else{

attributes.addFlashAttribute("error","用户名或密码错误");

return "redirect:/login";

}

}

使用RedirectAttributes attributes的话需要在springmvc中配置一个<mvc:view-controller path="/login" view-name="login"></mvc:view-controller>