### 1、实现登录成功/失败、登出处理逻辑

#### （1）登录成功

/\*\*  
\* @Author   
\* @Description: 登录成功处理逻辑  
\* @Date Create in    
\*/  
@Component  
public class CustomizeAuthenticationSuccessHandler implements AuthenticationSuccessHandler {  
  @Autowired  
  SysUserService sysUserService;  
​  
  @Override  
  public void onAuthenticationSuccess(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, Authentication authentication) throws IOException, ServletException {  
      //更新用户表上次登录时间、更新人、更新时间等字段  
      User userDetails = (User)SecurityContextHolder.getContext().getAuthentication().getPrincipal();  
      SysUser sysUser = sysUserService.selectByName(userDetails.getUsername());  
      sysUser.setLastLoginTime(new Date());  
      sysUser.setUpdateTime(new Date());  
      sysUser.setUpdateUser(sysUser.getId());  
      sysUserService.update(sysUser);  
         
      //此处还可以进行一些处理，比如登录成功之后可能需要返回给前台当前用户有哪些菜单权限，  
      //进而前台动态的控制菜单的显示等，具体根据自己的业务需求进行扩展  
​  
      //返回json数据  
      JsonResult result = ResultTool.success();  
      //处理编码方式，防止中文乱码的情况  
      httpServletResponse.setContentType("text/json;charset=utf-8");  
      //塞到HttpServletResponse中返回给前台  
      httpServletResponse.getWriter().write(JSON.toJSONString(result));  
  }  
}  
​

#### （2）登录失败

登录失败处理器主要用来对登录失败的场景（密码错误、账号锁定等…）做统一处理并返回给前台统一的json返回体。还记得我们创建用户表的时候创建了账号过期、密码过期、账号锁定之类的字段吗，这里就可以派上用场了.

/\*\*  
\* @Author: Hutengfei  
\* @Description: 登录失败处理逻辑  
\* @Date Create in 2019/9/3 15:52  
\*/  
@Component  
public class CustomizeAuthenticationFailureHandler implements AuthenticationFailureHandler {  
​  
​  
  @Override  
  public void onAuthenticationFailure(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, AuthenticationException e) throws IOException, ServletException {  
      //返回json数据  
      JsonResult result = null;  
      if (e instanceof AccountExpiredException) {  
          //账号过期  
          result = ResultTool.fail(ResultCode.USER\_ACCOUNT\_EXPIRED);  
      } else if (e instanceof BadCredentialsException) {  
          //密码错误  
          result = ResultTool.fail(ResultCode.USER\_CREDENTIALS\_ERROR);  
      } else if (e instanceof CredentialsExpiredException) {  
          //密码过期  
          result = ResultTool.fail(ResultCode.USER\_CREDENTIALS\_EXPIRED);  
      } else if (e instanceof DisabledException) {  
          //账号不可用  
          result = ResultTool.fail(ResultCode.USER\_ACCOUNT\_DISABLE);  
      } else if (e instanceof LockedException) {  
          //账号锁定  
          result = ResultTool.fail(ResultCode.USER\_ACCOUNT\_LOCKED);  
      } else if (e instanceof InternalAuthenticationServiceException) {  
          //用户不存在  
          result = ResultTool.fail(ResultCode.USER\_ACCOUNT\_NOT\_EXIST);  
      }else{  
          //其他错误  
          result = ResultTool.fail(ResultCode.COMMON\_FAIL);  
      }  
      //处理编码方式，防止中文乱码的情况  
      httpServletResponse.setContentType("text/json;charset=utf-8");  
      //塞到HttpServletResponse中返回给前台  
      httpServletResponse.getWriter().write(JSON.toJSONString(result));  
  }  
}  
​

#### （3）登出

同样的登出也要将登出成功时结果返回给前台，并且登出之后进行将cookie失效或删除

/\*\*  
\* @Author: Hutengfei  
\* @Description: 登出成功处理逻辑  
\* @Date Create in 2019/9/4 10:17  
\*/  
@Component  
public class CustomizeLogoutSuccessHandler implements LogoutSuccessHandler {  
  @Override  
  public void onLogoutSuccess(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, Authentication authentication) throws IOException, ServletException {  
      JsonResult result = ResultTool.success();  
      httpServletResponse.setContentType("text/json;charset=utf-8");  
      httpServletResponse.getWriter().write(JSON.toJSONString(result));  
  }  
}  
​