10、单点登录密码管理

# 接9中代码

# 密码管理特性

cas目前已经支持很多常用的特性，那么密码管理也是支持得不错的

* 邮件发送验证
* 短信发送验证
* 问题回答验证
* 忘记密码/重置密码UI

# 首先说明我的不知道为啥没有成功，但是过程觉得没错，有信心的你们试试，肯定可以成功

# 1、打开qq的smpt并且得到密码

# 2、这个时候，不使用我们自定义的主题登录，而且不使用rest登录，而使用数据库的MD5加密登录，具体如下

##  
# CAS Server Context Configuration  
#  
server.context-path=/cas  
server.port=8443  
  
  
#SSL配置  
server.ssl.enabled=true  
server.ssl.key-store=classpath:tomcat.keystore  
server.ssl.key-store-password=123456  
server.ssl.keyAlias=passport.sso.com  
#SSL配置  
  
  
  
#4.jdbc  
#  
#Query Database Authentication 数据库查询校验用户名开始  
#查询账号密码sql，必须包含密码字段  
cas.authn.jdbc.query[0].sql=select \* from sys\_user where username=?  
#指定上面的sql查询字段名（必须）  
cas.authn.jdbc.query[0].fieldPassword=password  
#指定过期字段，1为过期，若过期不可用  
cas.authn.jdbc.query[0].fieldExpired=expired  
#为不可用字段段，1为不可用，需要修改密码  
cas.authn.jdbc.query[0].fieldDisabled=disabled  
#数据库方言hibernate的知识  
cas.authn.jdbc.query[0].dialect=org.hibernate.dialect.MySQL5Dialect  
#数据库驱动  
cas.authn.jdbc.query[0].driverClass=com.mysql.jdbc.Driver  
#数据库连接  
cas.authn.jdbc.query[0].url=jdbc:mysql://localhost:3306/cas?useUnicode=true&characterEncoding=UTF-8&autoReconnect=true&useSSL=true  
#数据库用户名  
cas.authn.jdbc.query[0].user=root  
#数据库密码  
cas.authn.jdbc.query[0].password=123456  
#默认加密策略，通过encodingAlgorithm来指定算法，默认NONE不加密  
cas.authn.jdbc.query[0].passwordEncoder.type=DEFAULT  
cas.authn.jdbc.query[0].passwordEncoder.characterEncoding=UTF-8  
cas.authn.jdbc.query[0].passwordEncoder.encodingAlgorithm=MD5  
#Query Database Authentication 数据库查询校验用户名结束  
  
##  
##4、加盐处理  
##Encode Database Authentication 开始  
##加密次数  
#cas.authn.jdbc.encode[0].numberOfIterations=2  
##该列名的值可替代上面的值，但对密码加密时必须取该值进行处理  
#cas.authn.jdbc.encode[0].numberOfIterationsFieldName=  
## 盐值固定列  
#cas.authn.jdbc.encode[0].saltFieldName=username  
##静态盐值  
#cas.authn.jdbc.encode[0].staticSalt=.  
#cas.authn.jdbc.encode[0].sql=select \* from sys\_user\_encode where username=?  
##对处理盐值后的算法  
#cas.authn.jdbc.encode[0].algorithmName=MD5  
#cas.authn.jdbc.encode[0].passwordFieldName=password  
#cas.authn.jdbc.encode[0].expiredFieldName=expired  
#cas.authn.jdbc.encode[0].disabledFieldName=disabled  
##数据库连接  
#cas.authn.jdbc.encode[0].url=jdbc:mysql://localhost:3306/cas?useUnicode=true&characterEncoding=UTF-8&autoReconnect=true&useSSL=true  
#cas.authn.jdbc.encode[0].dialect=org.hibernate.dialect.MySQL5Dialect  
#cas.authn.jdbc.encode[0].driverClass=com.mysql.jdbc.Driver  
#cas.authn.jdbc.encode[0].user=root  
#cas.authn.jdbc.encode[0].password=123456  
##Encode Database Authentication 结束  
  
#  
## 5、shiro ,下面这里用的普通md5加密  
## Shiro Authentication 开始  
##允许登录的用户，必须要有以下角色，否则拒绝，多个逗号隔开  
#cas.authn.shiro.requiredRoles=admin  
##允许登录的用户，必须要有以下权限，否则拒绝，多个逗号隔开  
#cas.authn.shiro.requiredPermissions=staff  
##shir配置文件位置  
##由于目前classpath:shiro.ini会去临时目录找一遍，会发生一些错误，若用shiro，建议放到硬盘其他地方，写其他绝对路径  
#cas.authn.shiro.config.location=classpath:shiro.ini  
##shiro name 唯一  
#cas.authn.shiro.name=cas-shiro  
## 与Query Authentication一致的加密策略  
#cas.authn.shiro.passwordEncoder.type=DEFAULT  
#cas.authn.shiro.passwordEncoder.characterEncoding=UTF-8  
#cas.authn.shiro.passwordEncoder.encodingAlgorithm=MD5  
## Shiro Authentication 结束  
  
  
  
##5、REST 认证开始  
#cas.authn.rest.uri=http://localhost:8881/login  
#cas.authn.rest.passwordEncoder.type=DEFAULT  
#cas.authn.rest.passwordEncoder.characterEncoding=UTF-8  
#cas.authn.rest.passwordEncoder.encodingAlgorithm=MD5  
##REST 结束  
  
  
# 6、 service  
#开启识别json文件，默认false  
cas.serviceRegistry.initFromJson=true  
#自动扫描服务配置，默认开启  
#cas.serviceRegistry.watcherEnabled=true  
#120秒扫描一遍  
#cas.serviceRegistry.repeatInterval=120000  
#延迟15秒开启  
#cas.serviceRegistry.startDelay=15000  
#默认json/yml资源加载路径为resources/services  
#cas.serviceRegistry.config.location=classpath:/services  
  
  
# 8、修改默认的主题  
#通过service中确定主题的id（这里的id其实为thyme的值）  
#cas.theme.defaultThemeName=demo  
# 由于第一次访问的时候会对页面进行缓存，为了防止不显示我们修改过的主题，这里设置为false  
#spring.thymeleaf.cache=false  
  
# 9、单点登出  
#允许发出退出控制退出后转发url  
cas.logout.followServiceRedirects=true  
  
  
  
#10、密码管理 开始  
#允许内置密码管理  
cas.authn.pm.enabled=true  
#另外：由于已经发布的代码默认提交是false，导致不能修改密码成功请添加以下配置  
cas.authn.pm.jdbc.autocommit=true  
#重置信息 https://apereo.github.io/cas/5.1.x/installation/Configuration-Properties.html#password-management  
cas.authn.pm.reset.from=${spring.mail.username}  
#发送邮件标题  
cas.authn.pm.reset.subject=SSO DEMO 重置密码  
#邮件内容，必须要有%s，因为会生成一个连接并且带了token，否则无法打开链接，当然这个链接也和cas.server.prefix有关系  
cas.authn.pm.reset.text=打开以下链接重置您的密码（SSO-DEMO）: %s  
#token失效分钟数  
cas.authn.pm.reset.expirationMinutes=10  
cas.authn.pm.reset.emailAttribute=mail  
#是否开启问题回答  
cas.authn.pm.reset.securityQuestionsEnabled=true  
#新密码必须匹配表达式  
cas.authn.pm.policyPattern=\\d{3,10}  
  
#发送邮件  
spring.mail.host=smtp.qq.com  
spring.mail.port=465  
#邮箱用户名  
spring.mail.username=1318830916@qq.com  
#邮箱授权码  
spring.mail.password=bnnpxbbmnslxh  
 #mjwompxtqeaziaeg  
spring.mail.testConnection=false  
spring.mail.properties.mail.smtp.auth=true  
#必须ssl  
spring.mail.properties.mail.smtp.ssl.enable=true  
  
#邮箱查找  
#根据用户名查找问题  
cas.authn.pm.jdbc.sqlSecurityQuestions=select question, answer from sys\_user\_question where username=?  
#根据用户名查找邮箱  
cas.authn.pm.jdbc.sqlFindEmail=select email from sys\_user where username=?  
#修改密码  
cas.authn.pm.jdbc.sqlChangePassword=update sys\_user set password=? where username=?  
cas.authn.pm.jdbc.url=${cas.authn.jdbc.query[0].url}  
cas.authn.pm.jdbc.user=${cas.authn.jdbc.query[0].user}  
cas.authn.pm.jdbc.password=${cas.authn.jdbc.query[0].password}  
cas.authn.pm.jdbc.dialect=${cas.authn.jdbc.query[0].dialect}  
cas.authn.pm.jdbc.driverClass=${cas.authn.jdbc.query[0].driverClass}  
#密码修改加密规则，这个必须要和原始密码加密规则一致  
cas.authn.pm.jdbc.passwordEncoder.type=${cas.authn.jdbc.query[0].passwordEncoder.type}  
cas.authn.pm.jdbc.passwordEncoder.characterEncoding=${cas.authn.jdbc.query[0].passwordEncoder.characterEncoding}  
cas.authn.pm.jdbc.passwordEncoder.encodingAlgorithm=${cas.authn.jdbc.query[0].passwordEncoder.encodingAlgorithm}  
cas.authn.pm.jdbc.passwordEncoder.secret=${cas.authn.jdbc.query[0].passwordEncoder.secret}  
#密码管理 结束  
  
  
  
server.max-http-header-size=2097152  
server.use-forward-headers=true  
server.connection-timeout=20000  
server.error.include-stacktrace=NEVER  
  
server.tomcat.max-http-post-size=2097152  
server.tomcat.basedir=build/tomcat  
server.tomcat.accesslog.enabled=true  
server.tomcat.accesslog.pattern=%t %a "%r" %s (%D ms)  
server.tomcat.accesslog.suffix=.log  
server.tomcat.max-threads=10  
server.tomcat.port-header=X-Forwarded-Port  
server.tomcat.protocol-header=X-Forwarded-Proto  
server.tomcat.protocol-header-https-value=https  
server.tomcat.remote-ip-header=X-FORWARDED-FOR  
server.tomcat.uri-encoding=UTF-8  
  
spring.http.encoding.charset=UTF-8  
spring.http.encoding.enabled=true  
spring.http.encoding.force=true  
  
##  
# CAS Cloud Bus Configuration  
#  
spring.cloud.bus.enabled=false  
  
endpoints.enabled=false  
endpoints.sensitive=true  
  
endpoints.restart.enabled=false  
endpoints.shutdown.enabled=false  
  
management.security.enabled=true  
management.security.roles=ACTUATOR,ADMIN  
management.security.sessions=if\_required  
management.context-path=/status  
management.add-application-context-header=false  
  
security.basic.authorize-mode=role  
security.basic.enabled=false  
security.basic.path=/cas/status/\*\*  
  
##  
# CAS Web Application Session Configuration  
#  
server.session.timeout=300  
server.session.cookie.http-only=true  
server.session.tracking-modes=COOKIE  
  
##  
# CAS Thymeleaf View Configuration  
#  
spring.thymeleaf.encoding=UTF-8  
spring.thymeleaf.cache=true  
spring.thymeleaf.mode=HTML  
##  
# CAS Log4j Configuration  
#  
# logging.config=file:/etc/cas/log4j2.xml  
server.context-parameters.isLog4jAutoInitializationDisabled=true  
  
##  
# CAS AspectJ Configuration  
#  
spring.aop.auto=true  
spring.aop.proxy-target-class=true  
  
##  
# CAS Authentication Credentials  
#  
#cas.authn.accept.users=casuser::Mellon

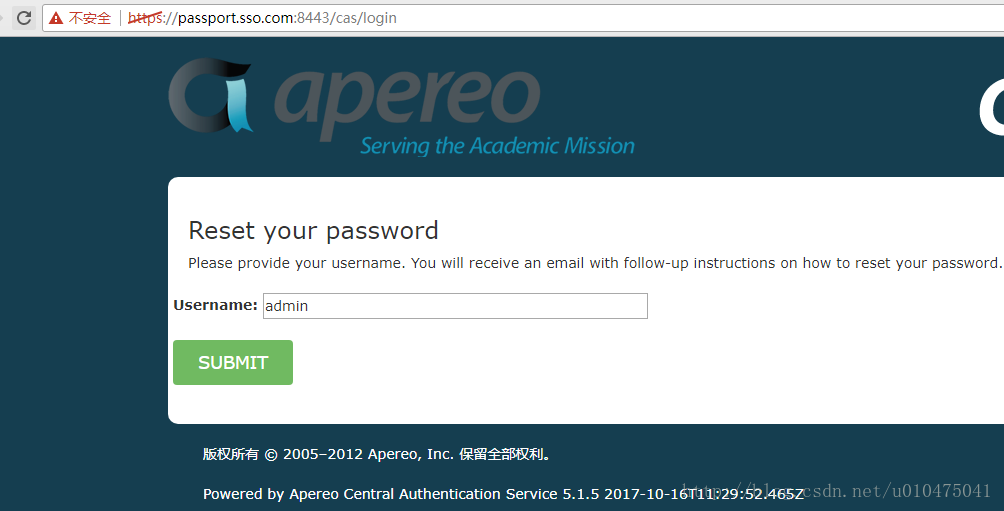
# 3、密保问题，简历数据库表和插入测试数据（如果测试的话，记得修改下之前我设置的用户邮箱）

/\*  
密码重置问题表  
\*/  
CREATE TABLE SYS\_USER\_QUESTION (  
 USERNAME VARCHAR(30) NOT NULL,  
 QUESTION VARCHAR(200) NOT NULL,  
 ANSWER VARCHAR(100) NOT NULL  
);

---问题数据  
INSERT INTO SYS\_USER\_QUESTION VALUES ('admin', '使用过的密码是？', '123');  
INSERT INTO SYS\_USER\_QUESTION VALUES ('admin', '你的年龄是？', '24');  
INSERT INTO SYS\_USER\_QUESTION VALUES ('zhangsan', '我的名字是？', 'zhangsan');  
INSERT INTO SYS\_USER\_QUESTION VALUES ('zhangsan', '我在哪里工作？', 'guangzhou');

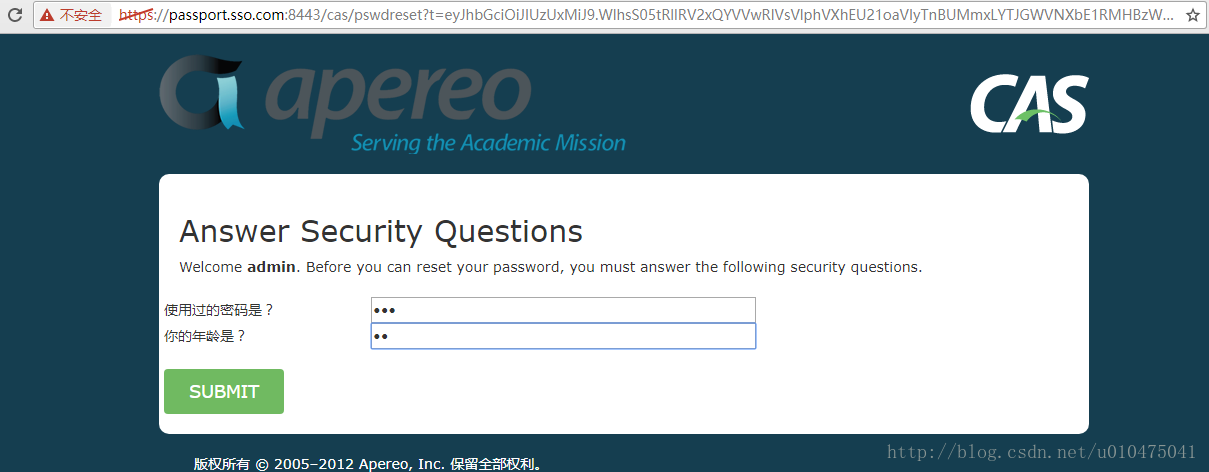
# 4、开始测试（我没有经历过下面的测试）

打开登录页，点击登录按钮下发的重置密码按钮



邮箱收取打开验证   


# 5、打开验证并回答问题



提交修改   


任务完成   
