链滴

【类库教程】JAVA AD 域登录验证

作者：moonce
原文链接：https://ld246.com/article/1539616877482
来源网站：链滴

AD域登录验证

作者:Moonce

AD域登录验证

作者：Grey

原文地址：http：//www.cnblogs.com/greyzeng/p/5799699.html

需求

系统在登录的时候，需要根据用户名和密码验证连接域服务器进行验证此用户是否为域用户。

条件

- 域服务器地址：XXXX
- 域验证端口：XXX
- AD域为：DC = adservice，DC = COM
- 某个域用户是：abc@adservice.com 密码：abc123。

实现

Java版

ADAuthJava.java

```
package com.hui.advalidationdemo;

import static com.hui.advalidationdemo.constant.ApplicationConstants.buildADPath;
import static com.hui.advalidationdemo.constant.ApplicationConstants.getConfig;
import static javax.naming.Context.INITIAL_CONTEXT_FACTORY;
import static javax.naming.Context.PROVIDER_URL;
import static javax.naming.Context.SECURITY_AUTHENTICATION;
import static javax.naming.Context.SECURITY_CREDENTIALS;
import static javax.naming.Context.SECURITY_PRINCIPAL;

import java.util.Hashtable;

import javax.naming.directory.DirContext;
import javax.naming.directory.InitialDirContext;

public class ADAuthJava {

    public static boolean authenticate(String username, String password) {
        DirContext ctx = null;
        Hashtable<String, String> HashEnv = initADServer(username, password);
        try {
```

```
            ctx = new InitialDirContext(HashEnv);
            System.out.println("Authenticate Success!");
            return true;
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        } finally {
            if (null != ctx) {
                try {
                    ctx.close();
                    ctx = null;
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
    }

    private static Hashtable<String, String> initADServer(String username, String password) {
        String adPath = buildADPath(username);
        Hashtable<String, String> HashEnv = new Hashtable<String, String>();
        HashEnv.put(SECURITY_AUTHENTICATION, "simple");
        HashEnv.put(SECURITY_PRINCIPAL, adPath);
        HashEnv.put(SECURITY_CREDENTIALS, password);
        HashEnv.put(INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
        HashEnv.put("com.sun.jndi.ldap.connect.timeout", "3000");
        HashEnv.put(PROVIDER_URL, getConfig("ad.url"));
        return HashEnv;
    }
}
```

单元测试：ADAuthJavaTest.java

```
package com.hui.advalidationdemo;

import static com.hui.advalidationdemo.ADAuthJava.authenticate;
import static org.junit.Assert.assertTrue;

import org.junit.Test;
public class ADAuthJavaTest {
    @Test
    public void testAuthenticate() {
        assertTrue(authenticate("abc", "abc123."));
    }
}
```

Spring版（附源码，Maven的项目）

- Spring版本：3.2.3.RELEASE
- spring-ldap-core版本：2.0.2.RELEASE
- JDK1.7 +

pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
MLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/m
ven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.hui</groupId>
    <artifactId>advalidationdemo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>advalidationdemo</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.12</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.ldap</groupId>
            <artifactId>spring-ldap-core</artifactId>
            <version>2.0.2.RELEASE</version>
        </dependency>
        <dependency>
            <groupId>org.acegisecurity</groupId>
            <artifactId>acegi-security</artifactId>
            <version>1.0.7</version>
        </dependency>
        <dependency>
            <groupId>org.apache.commons</groupId>
            <artifactId>commons-lang3</artifactId>
            <version>3.4</version>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-beans</artifactId>
            <version>3.2.3.RELEASE</version>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-core</artifactId>
            <version>3.2.3.RELEASE</version>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
```

```
        <artifactId>spring-test</artifactId>
        <version>3.2.3.RELEASE</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>3.2.3.RELEASE</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context-support</artifactId>
        <version>3.2.3.RELEASE</version>
    </dependency>
  </dependencies>
</project>
```

的applicationContext-ldap.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN 2.0//EN" "http://www.springframework.or
/dtd/spring-beans-2.0.dtd">
<beans>
  <bean id="configBean" class="org.springframework.beans.factory.config.PropertyPlacehol
erConfigurer">
    <property name="location"><value>classpath:config.properties</value></property>
  </bean>
  <bean id="contextSource" class="org.springframework.ldap.core.support.LdapContextSour
e">
    <property name="url" value="${ad.url}" />
    <property name="base" value="${ad.base}" />
  </bean>
  <bean id="ldapTemplate" class="org.springframework.ldap.core.LdapTemplate">
    <constructor-arg ref="contextSource" />
  </bean>
  <bean id="adDao" class="com.hui.advalidationdemo.ADAuthSpring">
    <property name="ldapTemplate" ref="ldapTemplate" />
  </bean>
</beans>
```

ADAuthSpring.java

```
package com.hui.advalidationdemo;

import static com.hui.advalidationdemo.constant.ApplicationConstants.buildADPath;
import static org.acegisecurity.ldap.LdapUtils.closeContext;

import javax.naming.directory.DirContext;

import org.springframework.ldap.core.LdapTemplate;

public class ADAuthSpring {
   private LdapTemplate ldapTemplate;

   public void setLdapTemplate(LdapTemplate ldapTemplate) {
```

```
        this.ldapTemplate = ldapTemplate;
    }

    public boolean authenticate(String userName, String password) {
        DirContext ctx = null;
        String distinguishedName = null;
        distinguishedName = buildADPath(userName);
        System.out.println("userName:" + userName + " map distinguishedName:" + distinguish
dName);
        try {
            distinguishedName = buildADPath(userName);
            System.out.println("userName:" + userName + " map distinguishedName:" + distingui
hedName);

            ctx = ldapTemplate.getContextSource().getContext(distinguishedName, password);
            System.out.println("authenticate success distinguishedName:" + distinguishedName +
" userName:" + userName);
            return true;
        } catch (Exception e) {
            System.out.println("authenticate fail distinguishedName:" + distinguishedName + " us
rName:" + userName);
            return false;
        } finally {
            closeContext(ctx);
        }
    }

}
```

config.properties

```
# AD Validation#
ad.url=ldap://x.x.x.x:xxx
ad.base=DC=adservice,DC=com
ad.path.template=%s@adservice.com
```

单元测试：

ADAuthSpringTest.java

```
package com.hui.advalidationdemo;

import org.junit.Assert;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = { "classpath:applicationContext-ldap.xml" })
public class ADAuthSpringTest {

    @Autowired
    public ADAuthSpring adValidation;
```

```
    @Test
    public void testAuth() {
        Assert.assertTrue(adValidation.authenticate("abc", "123abc."));
    }

}
```

ApplicationConstants.java

```
package com.hui.advalidationdemo.constant;

import static java.lang.String.format;
import static java.lang.Thread.currentThread;
import static org.apache.commons.lang3.StringUtils.isBlank;

import java.io.IOException;
import java.io.InputStream;
import java.util.HashMap;
import java.util.Map;
import java.util.Properties;

import org.apache.log4j.Logger;

public class ApplicationConstants {

    private static final String CONFIG_FILE = "config.properties";
    private static Map<String, Object> configs = new HashMap<String, Object>();

    private static final Logger log = Logger.getLogger(ApplicationConstants.class);
    static {
        InputStream in = null;
        Properties p = new Properties();
        try{
            in = currentThread().getContextClassLoader().getResourceAsStream(CONFIG_FILE);
            p.load(in);
            for(Object k : p.keySet()){
                String key = (String) k;
                configs.put( key, p.getProperty(key));
            }
            log.info("config.properties is loaded!"  );
        } catch (IOException e){
            log.error("Unable to read config.properties");
        } finally{
            if(in != null)
                try {
                    in.close();
                } catch (IOException e) {
                    log.error("Unable to close inputstream");
                }
        }
    }

    public static String getConfig(String key){
```

```
        return (String) configs.get(key);
    }
    public static  String buildADPath(String userName) {
        String adPathTemplate = getConfig("ad.path.template");
        if (isBlank(adPathTemplate)) {
            log.error("ad.path template do not exist in config.properties please config it");
            return null;
        }
        log.debug("ad.path template is "+adPathTemplate);
        try {
            String adPath = format(adPathTemplate, userName);
            log.debug("adPath is:"+adPath);
            return adPath;
        } catch (Exception e) {
            log.error("ad path template format error");
            return null;
        }

    }

}
```

== ==注意：在测试的时候需要将XXXX，XXX，ABC，123ABC替换成相应的域服务器IP，域服务端口，域用户名，域用户密码