Código Java

import javax.security.auth.callback.\*;

import javax.security.auth.login.LoginException;

import javax.security.auth.spi.LoginModule;

import java.util.Map;

public class SimpleLoginModule implements LoginModule {

private CallbackHandler callbackHandler;

private boolean succeeded = false;

@Override

public void initialize(Subject subject, CallbackHandler callbackHandler, Map<String, ?> sharedState, Map<String, ?> options) {

this.callbackHandler = callbackHandler;

}

@Override

public boolean login() throws LoginException {

Callback[] callbacks = new Callback[2];

callbacks[0] = new NameCallback("Username: ");

callbacks[1] = new PasswordCallback("Password: ", false);

try {

callbackHandler.handle(callbacks);

String username = ((NameCallback) callbacks[0]).getName();

char[] password = ((PasswordCallback) callbacks[1]).getPassword();

// Aqui você deve implementar a lógica de autenticação.

// Por exemplo, verificar se o usuário e a senha são válidos.

// Neste exemplo, apenas verifica se o usuário é "admin" e a senha é "password".

if ("admin".equals(username) && "password".equals(new String(password))) {

succeeded = true;

} else {

succeeded = false;

throw new FailedLoginException("Login falhou");

}

} catch (Exception e) {

throw new LoginException(e.getMessage());

}

return succeeded;

}

@Override

public boolean commit() throws LoginException {

return succeeded;

}

@Override

public boolean abort() throws LoginException {

return false;

}

@Override

public boolean logout() throws LoginException {

return false;

}

}

import javax.security.auth.Subject;

import javax.security.auth.login.LoginContext;

import javax.security.auth.login.LoginException;

public class JaasExample {

public static void main(String[] args) {

System.setProperty("java.security.auth.login.config", "jaas.config");

try {

LoginContext loginContext = new LoginContext("JaasExample", new SimpleCallbackHandler());

loginContext.login();

Subject subject = loginContext.getSubject();

System.out.println("Usuário autenticado: " + subject.getPrincipals());

// Aqui você pode adicionar lógica de autorização, se necessário.

loginContext.logout();

} catch (LoginException e) {

e.printStackTrace();

}

}

}

import javax.security.auth.callback.Callback;

import javax.security.auth.callback.CallbackHandler;

import javax.security.auth.callback.NameCallback;

import javax.security.auth.callback.PasswordCallback;

public class SimpleCallbackHandler implements CallbackHandler {

@Override

public void handle(Callback[] callbacks) {

for (Callback callback : callbacks) {

if (callback instanceof NameCallback) {

NameCallback nameCallback = (NameCallback) callback;

nameCallback.setName("admin");

} else if (callback instanceof PasswordCallback) {

PasswordCallback passwordCallback = (PasswordCallback) callback;

passwordCallback.setPassword("password".toCharArray());

}

}

}

}