

Servidor SIP Brekeke

Versión 3

Complemento de autenticación

Guía del desarrollador

Brekeke Software, Inc.

BREKEKE SOFTWARE, INC.

Versión

Brekeke SIP Server Versión 3 Autenticación Plug-in Guía del desarrollador

Derechos de autor

Este documento tiene derechos de autor de Brekeke Software, Inc.

Copyright © 2013 Brekeke Software, Inc.

Este documento no puede ser copiado, reproducido, reimpresso, traducido, reescrito o reenviado total o parcial sin el consentimiento expreso y por escrito de Brekeke Software, Inc.

Descargo de responsabilidad

Brekeke Software, Inc. se reserva el derecho de cambiar cualquier información que se encuentre en este documento

sin previo aviso por escrito al usuario.

Reconocimiento de marca registrada



Oracle y Java son marcas registradas de Oracle y / o sus filiales. Otros nombres pueden ser marcas registradas de sus respectivos dueños.



Otros logotipos y nombres de productos y servicios contenidos en este documento son las propiedades de sus respectivos dueños.

1

Brekeke SIP Server Authentication Plug-in Guía del desarrollador

s-aplug3.1.0_130328

BREKEKE SOFTWARE, INC.

1)

VENIMOS 4

2)

REQUISITOS 5

2.1.

Directorio de usuarios 5

2.2.

Enchufar 5

3)

FLUJO DEL PROCESO..... 6

4)

CLASES E INTERFACES 7

4.1.

Interfaz UserDir	7
4.2.	
Clase UserRecord	7
4.3.	
Clase Envrnmt	8
1	
getStr	8
2	
getStr	8
3	
getInt	9
4 4	
getLong	9
4.4.	
Clase de registro	9
1	
impresión.....	9
2	
println	9
3	
println	10
4.5.	
Nivel de registro.....	10
1	
Nivel de registro.....	11
5)	
MÉTODOS	11
5.1.	
en eso	11
5.2.	
cerrar	11
5.3.	
buscar.....	11

2
Brekeke SIP Server Authentication Plug-in Guía del desarrollador
s-aplug3.1.0_130328

BREKEKE SOFTWARE, INC.

5.4.	
adjuntar.....	12
5.5.	
eliminar.....	12
5.6.	
eliminar.....	13
5.7.	
getCount ()	13
6)	
INSTALACIÓN ENCHUFABLE	13
7)	
CÓDIGO DE MUESTRA ENCHUFABLE	

14

3
Brekeke SIP Server Authentication Plug-in Guía del desarrollador
s-aplug3.1.0_130328

BREKEKE SOFTWARE, INC.

1. Welcome

El servidor SIP de Brekeke autentica los métodos de solicitud SIP, como INVITAR y REGISTRARSE enviado desde los agentes de usuario del cliente SIP. El servidor consulta la base de datos del directorio de usuarios y valida

la consulta a través del complemento de autenticación predeterminado. El directorio de usuarios es una base de datos de usuarios que

tiene un nombre de usuario y contraseña.

Si desea utilizar otro servicio de directorio de usuarios existente (una base de datos de usuarios o autenticación

servidor) para la autenticación en el servidor SIP de Brekeke, es posible creando su propio servidor

Complementos de autenticación.

En este documento, presentamos las clases, interfaces y métodos necesarios para desarrollar

Complementos de autenticación.

4 4

Brekeke SIP Server Authentication Plug-in Guía del desarrollador
s-aplug3.1.0_130328

BREKEKE SOFTWARE, INC.

2. Requisitos

2.1. Directorio de usuarios

Como requisito mínimo, el directorio de usuarios debe incluir un nombre de usuario y contraseña para cada usuario. El servidor SIP de Brekeke solo realiza consultas pero no agrega ni elimina datos de usuario. Por lo tanto, debes

crear y eliminar usuarios con otra herramienta.

2.2. Enchufar

El complemento se cargará como una clase Java. Se recomienda que use Java versión 1.6 o

más tarde para desarrollar un complemento. `UserDir` debe implementar la clase para el complemento interfaz (consulte la Sección 4.1).

Los métodos que deben implementarse en la clase de complemento se enumeran a continuación:

Nombre del método

Acción

en eso()

Inicializar el complemento

cerrar()

Cerrar el complemento

buscar()

Busca un usuario

adjuntar()

Agregar un usuario

eliminar()

Eliminar un usuario

getCount ()

Obtenga un número total de usuarios

Consulte la sección 5 Métodos para obtener detalles adicionales.

Para compilar su programa, incluya la siguiente ruta en CLASSPATH.

`<SIP_server_install_directory> \ webapps \ sip \ WEB-INF \ lib \ ondosip.jar`

5 5

Brekeke SIP Server Authentication Plug-in Guía del desarrollador

BREKEKE SOFTWARE, INC.

3. Flujo de proceso

UA

Servidor SIP Brekeke

- 1) Un agente de usuario SIP envía una solicitud SIP como REGISTRARSE o INVITAR a Brekeke SIP Servidor.
- 2) Brekeke SIP Server pasa el nombre de usuario, el nombre del método SIP, el SIP-URI de destino y Valor de encabezado de **autorización de proxy** o valor de encabezado de **autorización** que se encuentran en Solicitud SIP al método de **búsqueda** () del complemento .
- 3) El complemento consulta el directorio de usuarios para el registro del nombre de usuario.
- 4) El directorio del usuario devuelve un registro apropiado.
- 5) El complemento coloca el registro en un objeto **UserRecord** (consulte la Sección 4.2, UserRecord Class) y devuelve el objeto al Servidor SIP Brekeke. Si no se encuentra ningún registro, se devuelve un valor nulo.
- 6) Brekeke SIP Server compara el resultado devuelto con la información de autenticación en el Solicitud SIP si se **devuelve** un objeto **UserRecord** del complemento.

Enchufar

- 1) solicitud
- 2) búsqueda
- 5) Objeto de registro de usuario
- 4)
- 3)

Directorio de usuarios

6 6

Brekeke SIP Server Authentication Plug-in Guía del desarrollador
s-aplug3.1.0_130328

BREKEKE SOFTWARE, INC.

4. Clases e interfaces

Esta sección explica las clases e interfaces que necesita para desarrollar complementos.

4.1. Interfaz UserDir

Paquete: **com.brekeke.net.usrdir**

La clase de complemento debe implementarse mediante la interfaz **UserDir** .

Consulte la sección 5 Métodos para obtener detalles adicionales sobre los métodos de **UserDir** interfaz.

4.2. Clase UserRecord

Paquete: **com.brekeke.net.usrdir**

UserRecord es la clase que contiene la información de un usuario. Este objeto se devuelve desde un método de **búsqueda** () cuando el servidor SIP de Brekeke llama al método

La clase **UserRecord** no contiene métodos. Solo contiene las siguientes variables miembro.

Tipo

Nombre de la variable

Contenido

Cuerda

nombre de usuario

Nombre de usuario

Cuerda

contraseña

Contraseña en texto plano

booleano

b Autorizado

Una bandera sobre si está autorizado o no

largo

uid

ID de usuario (opcional)

largo

gid

ID de grupo (opcional)

Cuerda

nombre largo

Nombre de usuario largo (opcional)

Cuerda

correo electrónico

Dirección de correo electrónico (opcional)

Cuerda

desc

Descripción del usuario (Opcional)

largo

expirar el tiempo

Vencimiento (opcional)

largo

timemake

Fecha y hora de cuando este registro fue creado (Opcional)

Objeto

ext

Para extensión (opcional)

7 7

Brekeke SIP Server Authentication Plug-in Guía del desarrollador
s-aplug3.1.0_130328

BREKEKE SOFTWARE, INC.

Todas las variables, excepto nombre de usuario, contraseña y **bAuthorized**, son opcionales. Si el servidor SIP de Brekeke

necesita calcular una contraseña cifrada a partir de la contraseña simple y autorizar comparando

la contraseña calculada con un campo de encabezado (**Autorización Proxy o Autorización**) en la solicitud del agente de usuario, establezca **bAuthorized = false**. El valor predeterminado

de **bAuthorized** es

falso. Si la autorización se realiza en el complemento y Brekeke SIP Server no necesita hacer un autorización, establecer **bAuthorized = true**.

4.3. Clase de Medio Ambiente

Paquete: **com.brekeke.common**

Envrnmnt es una clase que contiene propiedades de entorno. El objeto **Envrnmnt** se pasa al plug-in como argumento del método **init ()**.

Las propiedades del entorno deben establecerse en la **[Configuración]** de la herramienta de administración del servidor SIP de Brekeke ->

[Avanzado] página de antemano. Reinicie el servidor SIP de Brekeke después de modificar la propiedad variables

1

getStr

Método:

public String getStr (clave de cadena)

Descripción: busque la propiedad Tipo de **cadena** con la clave especificada en las propiedades.

El método devuelve nulo si no se encuentra la clave.

Parámetros: **clave** : la clave de propiedad.

Devoluciones:

el valor de **cadena** con el valor clave especificado.

2

getStr

Método:

public String getStr (clave String, String defStr)

Descripción: Busque la propiedad de tipo de cadena con la clave especificada.

El método devuelve el argumento del valor predeterminado si no se encuentra la clave.

Parámetros: **clave** : la clave de propiedad.

defStr : el valor predeterminado.

Devoluciones:

El valor de **cadena** con el valor clave especificado.

8

Brekeke SIP Server Authentication Plug-in Guía del desarrollador
s-aplug3.1.0_130328

BREKEKE SOFTWARE, INC.

3

getInt

Método:

public int getInt (clave de cadena, int defNum)

Descripción: Busque la propiedad de tipo int con la clave especificada.

El método devuelve el argumento del valor predeterminado si no se encuentra la clave.

Parámetros: **clave** : la clave de propiedad.

defNum : el valor predeterminado.

Devoluciones:

El valor **int** con el valor clave especificado.

4 4

getLong

Método:

public long getLong (clave de cadena, defNum largo)

Descripción: Busque la propiedad de tipo **largo** con la clave especificada.

El método devuelve el argumento del valor predeterminado si no se encuentra la clave.

Parámetros : **clave** : la clave de propiedad.

defNum : el valor predeterminado.

Devoluciones :

El valor **largo** con el valor clave especificado.

4.4. Clase de registro

Paquete: **com.brekeke.common**

El **registro** es una clase para generar registros. Este objeto se pasa al complemento a través del método **init ()**

como argumento

1

impresión

Método:

impresión vacía pública (String str, LogLevel loglevel, int require)

Descripción: si el nivel de registro actual cumple con el nivel de registro requerido dado, se genera la cadena.

Parámetros: **str** : la cadena que se generará

loglevel : nivel de registro actual

require - Nivel de registro requerido

2

println

Método:

```
public void println (LogLevel loglevel, int require)
```

Descripción: si el nivel de registro actual cumple con el nivel de registro requerido dado, un nuevo carácter de línea (\ n) es salida

Parámetros: **loglevel** - Nivel de registro actual

require - Nivel de registro requerido

9 9

Brekeke SIP Server Authentication Plug-in Guía del desarrollador
s-aplug3.1.0_130328

BREKEKE SOFTWARE, INC.

3

println

Método:

```
public void println (String str, LogLevel loglevel, int require)
```

Descripción: si el nivel de registro actual cumple con el nivel de registro requerido dado, entonces la cadena y un nuevo

se emite el carácter de línea (\ n).

Parámetros: **str** : la cadena que se generará

loglevel : nivel de registro actual

require - Nivel de registro requerido

4.5. Nivel de registro

Paquete: **com.brekeke.common**

LogLevel es una clase que contiene valores de nivel de registro.

Esto se utiliza para iniciar sesión. El nivel se establece para la salida de la consola (salida estándar) o la salida del archivo.

Los valores de nivel de registro son los siguientes:

Nombre de la variable

Valor

Nivel

LOG_LEVEL_SILENT

0 0

Sin registro

LOG_LEVEL_SYSTEM

1

Sistema

LOG_LEVEL_ERROR

2

Error

LOG_LEVEL_WARNING

4 4

Advertencia

LOG_LEVEL_EXCEPTION

8

Excepción

LOG_LEVEL_STATUS

dieciséis

Estado

LOG_LEVEL_DETAIL

32

Detalle

LOG_LEVEL_DEBUG

64

Depurar

LOG_LEVEL_ALL

255

Todas

Puede especificar varios niveles de registro mediante la adición lógica (OR). Por ejemplo, si solo desea generar resultados

Registros de sistema y detalle, el nivel de registro será 34 (2 o 32).

Nivel de registro = 2 O 32 = 34

10

Brekeke SIP Server Authentication Plug-in Guía del desarrollador
s-aplug3.1.0_130328

BREKEKE SOFTWARE, INC.

1

Nivel de registro

Constructor: `public LogLevel (int levelConsole, int levelFile)`

Descripción: crea un nuevo objeto LogLevel.

Parámetros: `levelConsole` - Nivel de registro para la salida de la consola.

5. Métodos

Esta sección explica todos los métodos que debe implementar un complemento. Estos métodos se declaran en Interfaz `UserDir`.

5.1. en eso

Método:

`public void init (Envrnmnt env, Logging log) lanza Exception`

Descripción: se llama al método `init ()` cuando se inicia Brekeke SIP Server. Por favor incluya cualquier tipo

de inicialización con respecto al complemento, como conectar una base de datos o establecer niveles de registro.

Parámetros: `env` - objeto `Envrnmnt`

`log` : objeto de registro

5.2. cerrar

Método:

`public void close () lanza Exception`

Descripción: se llama al método `close ()` mientras Brekeke SIP Server se está cerrando. Por favor incluya cualquier tipo de terminación con respecto al complemento, como la desconexión con la base de datos.

5.3. buscar

Método:

`búsqueda pública de UserRecord (nombre de usuario de cadena, método de cadena, cadena`

`destino, cadena authinfo) lanza excepción`

Descripción:

Se llama al método de **búsqueda** cuando Brekeke SIP Server necesita hacer autenticación. Nombre de usuario, SIP

nombre del método de la solicitud, SIP-URI de destino y valor de encabezado de **Autorización de proxy** o

El valor del encabezado de **autorización** se pasa como argumento.

Se supone que el nombre de usuario debe usarse como clave de búsqueda para un directorio de usuarios. Nombre del método SIP y

El URI de destino se puede utilizar para que la información tome una decisión. Pon el resultado de búsqueda de la

usuario en un objeto **UserRecord** como valor de retorno.

11

Brekeke SIP Server Authentication Plug-in Guía del desarrollador
s-aplug3.1.0_130328

BREKEKE SOFTWARE, INC.

Debe establecer la variable de **nombre de usuario** en el objeto **UserRecord** . Si la autorización se realiza aquí, y

El servidor SIP de Brekeke no necesita hacer una autorización, no necesita establecer una **contraseña** variable, sin embargo, debe establecer **bAuthorized = true** . De lo contrario, se debe establecer una contraseña simple

en **contraseña** variable. Si la contraseña en el directorio del usuario está encriptada, debe desencriptarse.

El valor de retorno es un objeto **UserRecord** . Si desea fallar la autenticación independientemente de resultado de búsqueda, devuelve nulo.

Si se produce una excepción, Brekeke SIP Server falla la autenticación y no ejecuta el solicitud.

Parámetros: nombre de usuario

- Nombre de usuario del usuario que envió la solicitud.

método

- Nombre del método SIP de la solicitud

destino - Destino SIP-URI de la solicitud

authinfo: valor de encabezado de autorización de proxy (INVITE) o

Valor de encabezado de autorización (REGISTRO)

Devoluciones:

Objeto **UserRecord**

nulo por falla de autorización

5.4. adjuntar

Método:

public boolean append (registro de UserRecord) produce una excepción

Descripción: Brekeke SIP Server no llama a este método de agregar. Se supone que esto es llamado desde otras herramientas. Cuando se llama a este método, agregue el registro al directorio del usuario.

Parámetros: record - objeto UserRecord

Devoluciones:

cierto si tiene éxito

falso si no tiene éxito

5.5. eliminar

Método:

La eliminación pública booleana (registro de registro de usuario) genera una excepción

Descripción: Brekeke SIP Server no llama a este método de eliminación. Se supone que esto es llamado desde otras herramientas. Cuando se llama a este método, elimine el registro del directorio de usuarios.

Parámetros: record - objeto UserRecord

Devoluciones:

cierto si tiene éxito

falso si no tiene éxito

12

Brekeke SIP Server Authentication Plug-in Guía del desarrollador
s-aplug3.1.0_130328

BREKEKE SOFTWARE, INC.

5.6. eliminar

Método:

public boolean remove (String username) lanza Exception

Descripción: Brekeke SIP Server no llama a este método de eliminación. Se supone que esto es

llamado desde otras herramientas. Cuando se llama a este método, elimine el registro del nombre de usuario dado de

El directorio de usuarios.

Parámetros: **nombre de usuario** - nombre de usuario

Devoluciones:

cierto si tiene éxito

falso si no tiene éxito

5.7. getCount ()

Método:

public int getCount () lanza una excepción

Descripción: cuando se llama a este método, devuelve el número de usuarios en el directorio de usuarios.

Devoluciones: el número de usuarios en el directorio de usuarios

6. Instalación del complemento

Copie el complemento desarrollado en un directorio apropiado y agregue el directorio en la clase Java camino (CLASSPATH).

Además, establezca el nombre del complemento en la variable de propiedad "**net.usrdir.plugins**" en Brekeke SIP

Herramienta de administración del servidor **[Configuración]** -> página **[Avanzado]** .

Supongamos que el plugin **SampleUserDir.class** está en el directorio **/ plugins / usrdir** y el

El nombre del paquete de la clase de **complemento** es **com.domain.proxy.plugins**

Agregue "**/ plugins / usrdir**" a la variable de entorno **CLASSPATH**.

Por ejemplo:

CLASSPATH = / usr / java / jdk / lib: / plugins: / plugins / usrdir

Establezca el complemento en la variable de propiedad **net.usrdir.plugins** en el

[Configuración] -> página **[Avanzado]** .

Por ejemplo:

net.usrdir.plugins = com.domain.proxy.plugins.SampleUserDir

13

Brekeke SIP Server Authentication Plug-in Guía del desarrollador
s-aplug3.1.0_130328

BREKEKE SOFTWARE, INC.

7. Código de muestra de complemento

Aquí hay un código de muestra que explica cómo implementar init y el método de búsqueda.

```
paquete com.domain.proxy.plugins;
import com.brekeke.common. *;
import com.brekeke.net.usrdir. *;
La clase pública SampleUserDir implementa UserDir
{
    Envrmnt env = nulo;
    Registro de registro = nulo;
    LogLevel loglevel = null;
    // en eso
    public void init (Envrmnt env, Logging log) lanza Exception
    {
        this.env = env;
        this.log = log;
        // Configuración del nivel de registro
        // Se obtiene de la variable "net.userdir.loglevel.console"
        loglevel = new LogLevel (0,
        env.getInt ("net.userdir.loglevel.file", LogLevel.LOG_LEVEL_EXCEPTION));
        log.println ("SampleUserDir: start", loglevel, LogLevel.LOG_LEVEL_SYSTEM);
        //
        // Inicialización de la conexión de la base de datos, etc.
        //
```

```
}  
14  
Brekeke SIP Server Authentication Plug-in Guía del desarrollador  
s-aplug3.1.0_130328
```

BREKEKE SOFTWARE, INC.

```
15  
Brekeke SIP Server Authentication Plug-in Guía del desarrollador  
s-aplug3.1.0_130328  
// buscar  
búsqueda pública de UserRecord (nombre de usuario de cadena, método de cadena, destino  
de cadena,  
String authinfo) lanza Exception  
{  
//  
// consulta de la base de datos para obtener información del usuario por clave de  
nombre de usuario  
//  
if ( la información del usuario no se encuentra ) {  
retorno (nulo);  
}  
si ( el usuario doesn 't tiene la autoridad para ejecutar el método ) {  
retorno (nulo);  
}  
si ( el usuario doesn 't tiene la autoridad para acceder al destino SIP-URI ) {  
retorno (nulo);  
}  
Registro UserRecord = nuevo UserRecord ();  
record.username = nombre de usuario;  
record.password = la contraseña simple obtenida de la base de datos  
volver (registro);  
}  
}
```