Autor:	Asunto: Documentación de SerCen
Revisado:	Fecha: 29 de julio de 2022

Documentación de SerCen	
Fecha : 29 de julio de 2022	**Número de revisión**: Versión 1
Objeto del documento :	
Actores (empresas): ▪ **MDTEL**	

1. Introducción

Este documento muestra información sobre la funcionalidad Sercen y describe los diferentes comandos existentes.

2. Descripción

Sercen (Servicios Centrales) es un demonio de VIVAit que provee servicios centralizados a muchos elementos de la arquitectura. Sirve para identificar a los usuarios y garantizar que los mismos sean quienes dicen ser.

Puede haber varias instancias en el, y diferentes servicios pueden estar en diferentes Sercen

Ejemplos de servicios:

- Autenticación, de factor simple o doble factor
- Integración con proveedores de autenticación
- Click2talk (Demonio dentro de Vivait)

Sercen permite autenticar un usuario (simple factor o doble factor), utilizando para ello la base de datos de vivait y/u otro mecanismo externo, como puede ser el directorio activo de Microsoft.

SerCen se relaciona con :

- BBDD
- Asterisk (para el click2talk)
-VIVAit Meet (para por ejemplo las funciones “botón colaborar” y ”Recursos compartidos”.
- Con los portales a los que proporciona los servicios vía webservice

3. Interfaz para autenticación (app-webfon / serCen)

Este interfaz permite autenticar un usuario (simple factor o doble factor), utilizando para ello la base de datos de vivait y/u otro mecanismo externo, como puede ser el directorio activo de Microsoft. El objetivo es que app-webfon obtenga un token que debe incluirse en las comunicaciones con el resto de elementos para que estos conozcan que las solicitudes proceden de un usuario autenticado. Este token también puede utilizarse para que otras aplicaciones puedan se iniciadas desde app-webfon y no se requiera de un proceso específico de autenticación para cada aplicación. De modo recíproco, app-webfon debe soportar el ser invocado con un token y, en tal caso, considerar que el usuario ya está autenticado.

Por ello, se asume que cada usuario conoce su clave. Independientemente de quién gestione la clave, esta puede tener asociados periodos de expiración y de caducidad. Cuando finaliza el periodo de expiración, el usuario tiene que cambiar la clave obligatoriamente para poder acceder a sus credenciales (token). Tras la caducidad, ya no puede obtener credenciales válidas.

Cuando la autenticación utiliza las claves almacenadas en la base de datos vivait, serCen permite cambiar la clave. Si la validación de la clave se apoya en mecanismos externos (como puede ser el directorio activo), entonces serCen no soporta el cambio de claves, informando de ello en las respuestas correspondientes a los comandos de autenticación.

4. Comandos de SerCen

Los diferentes comandos de SerCen son:

COMANDOS	DESCRIPCIÓN
autenticar1	Para autenticar en base al primer factor (usuario y clave).
autenticar2	Para, si procede, autenticar en base al segundo factor (pin adicional).
autenticartokenaad	Autenticar empleando un token de "azure active directory"
validarToken	Para autenticar a un usuario en base a un token suministrado por otra aplicación
revalidarToken	Para ampliar el periodo de expiración de un token válido.
revocarToken	Para que un token deje de ser válido.
cambiarClave	Permite que una clave pueda se cambiada. Esto es posible cuando el usuario ha sido autenticado y, además, la clave tiene que estar siendo gestionada por vivait. Este cambio no es posible, por ejemplo, si se trata de una clave validada por un directorio activo.

4.1 Comandos autenticar1

Comandos autenticar1 y respuestas posibles:

autenticar1. Autenticación primer factor.

url (POST): https://<servidor_webfon>/sercen/postautenticar1

datos de entrada: {"cuenta": "fulano","clave$1aA": "dificil","aplicacion": "webfon","expira": 7200,"clvCambiar": "masdificil$1aA"}

- "expira" es opcional y siempre posible. Permite solicitar un periodo de expiración (segs) para el token devuelto.
- "clvCambiar" es opcional y solo se tiene en cuenta si no hay doble factor. Permite cambiar la clave del usuario, si es autenticado.

Ejemplos de posibles respuestas:

- **ejRes1**: {"errorNum": 101,"errorCad": "Clave incorrecta"}
- **ejRes2**: {"errorNum": 115,"errorCad": "Clave caducada"}
- **ejRes3**:{"errorNum": 116,"errorCad": "Clave expirada","dobleFactor": "no","clvLongitudMinima": 6,"clvRequiereNumeros": true, "clvRequiereMayusculas": true, "clvRequiereMinusculas": true, "clvRequiereCarsEspeciales": false}
- **ejRes4**:ejRes4: {"errorNum": 0,"errorCad": "OK","token": "1234567890","dobleFactor": "email","clvExpira": 86400}
- **ejRes5**:{"errorNum":0,"errorCad":"OK","token": "1234567890","dobleFactor": "email","clvExpira": 86400}

- **ejRes6**:{"errorNum": 0,"errorCad": "OK","token": "1234567890","dobleFactor": "no","token2": "0987654321","expira": 3600,"clvExpira": 86400,"clvPuedeCambiar": true,"clvHaCambiado": true,"clvLongitudMinima": 6,"clvRequiereNumeros": true, "clvRequiereMayusculas": true, "clvRequiereMinusculas": true, "clvRequiereCarsEspeciales": false}

Este comando se invoca para autenticar al usuario en base a cuenta y clave. Opcionalmente, se soporta un doble factor en base a un pin que se comunica al usuario mediante correo electrónico u otro medio.

El campo "expira" en la respuesta informa sobre el token devuelto. Dicho token es válido durante un periodo que se expresa en segundos a transcurrir desde el momento en que se informa.

El campo "clvExpira" en la respuesta informa en segs. sobre cuando va a expirar la clave introducida y validada (errorNum=0). Si devuelve cero o no devuelve nada, quiere decir que no va a expirar por estar configurado de esa manera o porque es un autenticación ldap.

Al invocar este comando, app-webfon solicita, opcionalmente (campo "expira"), un periodo (en segundos) cuyo valor debe formar parte de su configuración. El valor finalmente válido es el que aparece con el mismo nombre en la respuesta (si finalmente es positiva) y si no hay doble factor.

Es responsabilidad de app-webfon el efectuar una solicitud para revalidar el token, mediante un procedimiento que se describirá posteriormente, tantas veces como considere necesario.

En lo que se refiere a las posibles respuestas a autenticar1, pueden darse varios casos:

Se produce un error: ejRes1 y ejRes2.

La clave ha expirado y no hay doble factor (ejRes3). Se invita a cambiar la clave invocando de nuevo a autenticar1 con "clvCambiar", para lo cuál se aportan datos de la complejidad requerida.

La clave es válida y hay doble factor en ejRes4.

La clave es válida y hay doble factor "totp" con usuario no enrolado en ejRes6. Se incluye un código QR que permite enrolar al usuario

La clave es válida y no hay doble factor en ejRes6. El usuario ha sido autenticado. Se devuelven "token" y "token2". También se informa sobre si es posible cambiar la clave ("clvPuedeCambiar") y sobre sus requerimientos de complejidad. Si el comando autenticar1 hubiese incluido un cambio de clave mediante el campo "clvCambiar" y este cambio hubiese sido completado, se devolverá un booleano "clvHaCambiado" para informar de ello.

4.2 Comandos autenticar2

Comandos autenticar2 y respuestas posibles:

autenticar2. Autenticación doble factor.

url (POST): https://<servidor_webfon>/sercen/postautenticar2

datos de entrada: {"token": "1234567890","pin": "1234","expira": 7200,"clvCambiar": "masdificil$1aA"}

- "expira" es opcional.
- "clvCambiar" es opcional. Permite cambiar la clave del usuario, si es autenticado.

Ejemplos de posibles respuestas:

- **ejRes1:** {"errorNum": 102,"errorCad": "Pin incorrecto"}
- **ejRes2:** {"errorNum": 115,"errorCad": "Clave caducada"}
- **ejRes3:** {"errorNum": 116,"errorCad": "Clave expirada","clvLongitudMinima": 6,"clvRequiereNumeros": true, "clvRequiereMayusculas": true, "clvRequiereMinusculas": true, "clvRequiereCarsEspeciales": false}

- **ejRes4:**{"errorNum": 0,"errorCad": "OK","token2": "0987654321","expira": 3600,"clvExpira": 86400,"clvPuedeCambiar": true,"clvHaCambiado": true,"clvLongitudMinima": 6,"clvRequiereNumeros": true, "clvRequiereMayusculas": true, "clvRequiereMinusculas": true, "clvRequiereCarsEspeciales": false}

Este comando se invoca para autenticar en base a un segundo factor: pin enviado por un medio de transporte independiente.

El campo "expira" en la respuesta informa sobre el token enviado y token2 devuelto. Dichos token son válidos durante un periodo que se expresa en segundos a transcurrir desde el momento en que se informa.

El campo "clvExpira" en la respuesta informa en segs. sobre cuando va a expirar la clave introducida y validada (errorNum=0). Si devuelve cero o no devuelve nada, quiere decir que no va a expirar por estar configurado de esa manera o porque es un autenticación ldap.

El campo que requiere explicación adicional es "token2". "token" puede ser traspasado a otras aplicaciones para evitar procesos de autenticación, pero esas aplicaciones (a las que no debe pasarse el valor de "token2") no podrán realizar operaciones relacionadas con el propio token, como puede ser revocarlo, cuando el usuario cierre el app-webfon. Es decir, el valor de "token2" sólo se requiere para operaciones de gestión del propio token. Sin "token2" tampoco es posible ejecutar el comando "cambiarClave".

En lo que se refiere a las posibles respuestas a autenticar2, pueden darse varios casos:

Se produce un error: ejRes1 y ejRes2.

La clave ha expirado (ejRes3). Se invita a cambiar la clave invocando de nuevo a autenticar2 con "clvCambiar", para lo cuál se aportan datos de la complejidad requerida.

El pin del doble factor es válido en ejRes4. El usuario ha sido autenticado. Se devuelve "token2" ("token" ya se tiene). También se informa sobre si es posible cambiar la clave ("clvPuedeCambiar") y sobre sus requerimientos de complejidad. Si el comando autenticar2 hubiese incluido un cambio de clave mediante el campo "clvCambiar" y este cambio hubiese sido completado, se devolverá un booleano "clvHaCambiado" para informar de ello.

4.3 Comando autenticartokenaad

Autenticar empleando un token de "Azure active directory"

url (POST): https://<servidor_webfon>/sercen/postautenticartokenaad

datos de entrada: {"token": "tokenrequetelargao","aplicacion": "webfon","expira": 7200} ("expira" es opcional).

Ejemplos de posibles respuestas:

```
{"errorNum": 101,"errorCad": "Token incorrecto"}
```

```
{"errorNum": 0,"errorCad": "OK","token": "1234567890","dobleFactor": "email"}
```

```
{"errorNum": 0,"errorCad": "OK","token": "1234567890","dobleFactor": "no","token2": "0987654321","expira": 3600}
```

4.4 Comando validartoken

Es un comando muy simple, que permite conocer si un token es válido y su periodo de expiración.

url (POST): https://<servidor_webfon>/sercen/postvalidartoken

datos de entrada: {"token": "1234567890"}

en datos de entrada puede ir opcionalmente "token2". Si es así, se validan ambos.

Ejemplos de posibles respuestas:

{"errorNum": 100,"errorCad": "Token expirado"}

{"errorNum": 103,"errorCad": "Token incorrecto"}

{"errorNum": 0,"errorCad": "OK","expira": 2538}

4.5 Comando revalidartoken

Permite ampliar el periodo de expiración de un token. El periodo ampliado es el que se indica en la respuesta.

url (POST): https://<servidor_webfon>/sercen/revalidartoken

datos de entrada: {"token": "1234567890","token2": "0987654321","expira": 7200} ("expira" es opcional).

Ejemplos de posibles respuestas:

{"errorNum": 100,"errorCad": "Token expirado"}

{"errorNum": 103,"errorCad": "Token incorrecto"}

{"errorNum": 0,"errorCad": "OK","expira": 3600}

4.6 Comando revocartoken

Permite revocar un token.

url (POST): https://<servidor_webfon>/sercen/postrevocartoken

datos de entrada: {"token": "1234567890","token2": "0987654321"}

Ejemplos de posibles respuestas:

{"errorNum": 103,"errorCad": "Token incorrecto"}

{"errorNum": 0,"errorCad": "OK"}

4.7 Comando cambiarClave

Permite que un usuario autenticado cambie su clave y amplíe el periodo de expiración y de caducidad.

url (POST): https://<servidor_webfon>/sercen/postcambiarclave

datos de entrada:{"token": "1234567890","token2": "0987654321","clvCambiar": "masdificil$1aA"}

Ejemplos de posibles respuestas:

{"errorNum": 102,"errorCad": "Clave incorrecta"}

{"errorNum": 117,"errorCad": "Clave demasiado simple"}

{"errorNum": 118,"errorCad": "Clave con caracteres no soportados"}

{"errorNum": 119,"errorCad": "Clave no admite cambio"}

{"errorNum": 120,"errorCad": "Clave repetida"}

{"errorNum": 0,"errorCad": "OK","clvExpira": 86400}

5. Posibles errores

Número de error	Descripción breve
0	OK
100	Token expirado
101	Pin expirado
102	Clave incorrecta
103	Token incorrecto
104	Token2 incorrecto
105	Pin incorrecto
106	Token comprometido
107	Token anulado
108	Token inválido
115	Clave caducada
116	Clave expirada
117	Clave demasiado simple
118	Clave con caracteres no soportados
119	Clave no admite cambio
120	Clave repetida
400	Error en datos de entrada
401	Error registro de extensión
402	Error en el estado del dispositivo
403	Error en el estado de la media con el navegador
404	No hay línea libre
405	La línea no está libre
406	La línea es errónea
407	Línea en estado erróneo
408	JSEP obligatorio
409	JSEP ‘offer’ obligatorio
410	Cifrado extremo a extremo no soportado
411	Canal de datos no soportado
412	Comando desconocido
413	No hay recursos
414	No hay conferencia libre
415	Error conferencia
416	Línea no conectada
500	Error interno