

SEDEB2B ONLINE

(Versión JAVA)

Intercambio electrónico de Datos

Guía de Acceso batch.

	SEDEB2B On Line (versión Java)	Edición	Página
		1	2

Índice

1.	Introducción. _____	3
2.	Funcionamiento. _____	3
3.	Paso de parámetros. _____	3
4.	Configuración. _____	4
5.	Recepción de información. _____	5
5.1	Ejemplo de recepción. _____	5
6.	Envío de información. _____	5
6.1	El fichero de control. _____	6
6.2	Ejemplos de envío. _____	6
7.	Acuses de recibo. _____	7
7.1	Ejemplo recepción del listado de acuses. _____	7
8.	Código resultado de la operación. _____	8
9.	Tarea de mantenimiento. _____	8

	SEDEB2B On Line (versión Java)	Edición	Página
		1	3

1. Introducción.

Con el fin de posibilitar el acceso de la VAN SEDEB2B desde cualquier aplicación, EDICOM ha desarrollado un ejecutable con la funcionalidad necesaria para conseguir la total integración en la red. SEDEB2BONLINE (en lo sucesivo SB2BO) permite recibir y enviar información, así como consultar el estado de los datos transmitidos. Toda la comunicación se puede hacer de forma segura utilizando encriptación SSL.

Además, la implementación en el lenguaje Java aporta importantes ventajas a nuestros productos, tales como potencia y robustez, así como una amplia integración con cualquier tipo de sistema gracias a su capacidad de ejecución multiplataforma.

Nota: SEDEB2B On Line Java requiere JRE 1.4.2 o versiones superiores.

2. Funcionamiento.

Esta aplicación funciona pasándole una serie de parámetros que en este documento se explican. En cada llamada únicamente podemos indicar que realice una cierta tarea, es decir, en la misma llamada no podemos solicitar que envíe y reciba.

3. Paso de parámetros.

Existen 2 formas de pasar los parámetros a la aplicación:

- Mediante parámetros en línea de comando. Cada parámetro debe ir precedido de una barra (/). Si además dispone de algún valor adicional, este debe estar separado por dos puntos (:). El valor si tiene algún espacio en blanco debe estar entre comillas doble. Ejemplo:

```
[user@edicom:~]$ java -jar SEDEB2BONLINE.jar /SERVER:TEST
/PORT:8019 /USER:USER1 /PWD:PASS1 ...
```

Observese que la siguiente línea es incorrecta por el espacio en blanco a partir de los dos puntos (:)

```
[user@edicom:~]$ java -jar SEDEB2BONLINE.jar /SERVER: _TEST ...
```

- Mediante un parámetro que indique un fichero de configuración. En el fichero debe de estar cada parámetro en una línea y precedido del símbolo. Ejemplo:

```
[user@edicom:~]$ java -jar SEDEB2BONLINE.jar
/INI:/SB2BO/INIS/SB2BO.INI
```

Ejemplo del contenido del fichero /SB2BO/INIS/SB2BO.INI:

```
/SERVER:TEST
/PORT:8019
/USER:USER1
/PWD:PASS1
```

	SEDEB2B On Line (versión Java)	Edición	Página
		1	4

Además, ambas formas de pasar los parámetros se pueden combinar. Podemos tener un fichero de configuración común y luego únicamente variar la función a utilizar.

4. Configuración.

Los siguientes parámetros definen el acceso al servicio SEDEB2B y permiten configurar ciertas opciones en la aplicación.

Datos de acceso al servidor	
/INI: <FichIni>	Fichero de configuración, en cual se le pueden pasar los parámetros de la llamada.
/SERVER: <Server>	Nombre o dirección IP donde está ubicado el servicio: adapters.sedeb2b.com
/PORT: <NroPort>	Puerto donde está escuchando el servicio: 9020
/SSL	Indica si la comunicación con el servidor va a ser SSL. En este caso hacen falta 2 librerías que deberán estar en el mismo directorio que SB2BO: ssleay32.dll y libeay32.dll. De momento es la única forma de acceder a SEDEB2B
/PROXYSERVER:	Indica el servidor PROXY a utilizar.
/PROXYPORT:	Puerto PROXY
/PROXYUSER:	Usuario de acceso al PROXY
/PROXYPWD:	Contraseña de acceso al PROXY

Datos del buzón	
/USER: <user>	Usuario de acceso al buzón
/PWD: <pwd>	Password de acceso.

Datos de configuración	
/TIMEOUT: <Seg>	Tiempo máximo en segundos que la aplicación estará activa. Transcurrido ese tiempo abortará cualquier operación.
/UPDATELEVEL: <Num>	<p>Tipo de actualización en función del número que se le pase:</p> <ul style="list-style-type: none"> 0. No se realiza ningún chequeo, solo se obtiene el número de versión. 1. Se comprueba a nivel de build. 2. Se comprueba la release (por defecto). 3. Se comprueba la menor versión. 4. Se comprueba la mayor versión.

	SEDEB2B On Line (versión Java)	Edición	Página
		1	5

5. Recepción de información.

A continuación se describe los parámetros para recibir los mensajes del buzón:

Acción a realizar	
/GET: <dirIn>	Recepción de los mensajes del buzón. El parámetro indica la ruta en al que se dejarán los ficheros
Parámetros adicionales	
/NUM: <Nro>	Permite bajar los mensajes en bloques.
/PARSEMIME	Permite extraer el contenido de mensajes en formato MIME. Funcionamiento: <ol style="list-style-type: none"> 1. Si el fichero ORIGINAL está en formato MIME recorre todas las partes y las va escribiendo en ficheros numerados con el siguiente patrón: <i>NomFichero_XXX.IN</i> donde XXX es el número de parte del mensaje MIME. Finalmente se elimina el fichero ORIGINAL. 2. Si el fichero ORIGINAL no está en formato MIME o es SMIME lo dejará con el nombre <i>NomFichero.IN</i> para no volver a procesarlo. 3. Si hay algún error al procesar el fichero lo dejará con la extensión <i>.ERR</i>.

5.1 Ejemplo de recepción.

En este ejemplo, usando la configuración contenida en el fichero Sedeb2bOnLine.Ini, recupera los mensajes del buzón, solicitándolos en bloques de 10 y dejándolos en el directorio /SB2BO/Inis/SB2BO.Ini.

```
[user@edicom:~]$ java -jar SEDEB2BONLINE.jar
/INI:/SB2BO/INIS/SB2BO.INI /GET:/SB2BO/ENTRADA/ /NUM:10
```

6. Envío de información.

A continuación se describe los parámetros para enviar información:

Acción a realizar	
/PUT: <MaskOut>	Envío de mensajes. El parámetro indica la máscara de los ficheros a enviar.
Parámetros adicionales globales al envío.	
/DIRCTR: <dir>	Directorio donde se encuentran los ficheros de control. Si se omite se asume que es el mismo donde está el fichero a procesar. Su uso se explica posteriormente.
/DESTINO: <Interloc>	Indica el interlocutor destino al que se le enviará el contenido del fichero. Se puede omitir con los mensajes EDI. De esta forma se obtiene de los segmentos de servicio. Los formatos EDI soportados son: EDIFACT, X12, TRADACOMS Y GENCOD.
/REF: <RefCli>	Referencia del cliente. Se puede usar las siguientes variables que serán sustituidas por sus correspondientes valores:

	SEDEB2B On Line (versión Java)	Edición	Página
		1	6

	@FILENAME: es el nombre del fichero que se está enviando.
/DEL	Borra todo fichero enviado correctamente.
/REN: <Mask>	Renombra según la máscara que se le pasa como parámetro todo fichero enviado correctamente. Ej /REN: *.ENV

El envío de ficheros tiene cierta complejidad añadida:

- Cada fichero puede ir a un destino diferente, y necesitamos poder indicarlo.
- SB2BO puede proporcionar el identificador del servidor asociado a un fichero enviado. Este identificador puede ser utilizado para enlazarlo con el acuse que posteriormente deberemos solicitar.

Por lo tanto, hace falta un mecanismo para intercambiar información a nivel de fichero de envío, el fichero de control.

6.1 El fichero de control.

Cada archivo a enviar puede tener un fichero de control que se llamará igual solo que con extensión .CTR. En él se puede redefinir ciertos valores como por ejemplo el destino del envío. Además se puede recuperar valores escritos por SB2BO como por ejemplo el identificador del servidor del envío.

El formato es con secciones, como los ficheros de configuración. Todos los parámetros deben estar bajo la sección [Control]. El nombre del parámetro y el valor están separado con el carácter igual =.

Parámetros de configuración particulares al fichero.	
Si se incluyen, redefinen los valores de los parámetros globales al envío.	
DESTINO=	Indica el interlocutor destino de este fichero.
REF=	Indica la referencia del cliente.

Datos devueltos por SB2BO	
IdServer=	Indica el identificador de envío del servidor.
Enviado=	Indica la fecha y hora del envío.

6.2 Ejemplos de envío.

- Sin fichero de control:

```
[user@edicom:~]$ java -jar SEDEB2BONLINE.jar
/INI:/SB2BO/INIS/SB2BO.INI /PUT:/SB2BO/SALIDA/*.EDI /DESTINO:USER2
```

```
[user@edicom:~]$ java -jar SEDEB2BONLINE.jar
/INI:/SB2BO/INIS/SB2BO.INI /PUT:/SB2BO/SALIDA/S121212.EDI
/DESTINO:USER2
```

- Con fichero de control:

```
[user@edicom:~]$ java -jar SEDEB2BONLINE.jar
/INI:/SB2BO/INIS/SB2BO.INI /PUT:/SB2BO/SALIDA/*.EDI
```

	SEDEB2B On Line (versión Java)	Edición	Página
		1	7

```
[user@edicom:~]$ java -jar SEDEB2BONLINE.jar
/INI:/SB2BO/INIS/SB2BO.INI /PUT:/SB2BO/SALIDA/S121212.EDI
```

Ejemplo de fichero de control S121212.CTR

```
[Control]
DESTINO=USER2
```

Tras el envío, SB2BO añadirá el Identificador del servidor y la fecha hora de envío.

```
[Control]
DESTINO=USER2
IdServer=0601021736098729754
Enviado=02/01/2006 17:36:10
```

7. Acuses de recibo.

Se puede solicitar al servidor un listado de los ficheros que han sido recuperados por el destinatario. Solo se recibe una vez el acuse.

Acción a realizar	
/Ack: <DirOut>	Ruta (y nombre opcional) donde se dejará el listado de acuses.

El listado tiene el siguiente formato:

- Cada línea es un acuse. La línea termina con los caracteres #13+#10 (CR+LF).
- Un acuse está formado por distintos campos separados por el carácter ~.

Significado de los campos:

IdServer	Identificador del servidor.
Mensaje	Tipo de mensaje
Origen	Emisor del mensaje. Viene vacío.
Destino	Receptor del mensaje.
Fecha	Fecha de cuando ha recibido el mensaje el destinatario.
ReferenciaCliente	Referencia del cliente.
Tipo	Tipo de mensaje.
Reservado	Reservado.
Tamaño	Tamaño en bytes del mensaje.

7.1 Ejemplo recepción del listado de acuses.

```
[user@edicom:~]$ java -jar SEDEB2BONLINE.jar
/INI:/SB2BO/INIS/SB2BO.INI /ACK: /SB2BO/TEMP/121212.ACK
```

Ejemplo del contenido del fichero 121212.ACK:

```
0601251759107788480~SM~~PEDRO4@SEDEB2B.COM~25/01/2006 17:59:24~~6~~1122
```

	SEDEB2B On Line (versión Java)	Edición	Página
		1	8

8. Código resultado de la operación.

Tras la ejecución el programa devuelve un código de resultado, con el que podemos comprobar si ha terminado correctamente o hemos tenido algún error.

Códigos de terminación	
0	Todo ha funcionado correctamente.
1	Programa finalizado con avisos. Para más información, consultar el log de sesión.
5	Programa finalizado con errores. Para más información, consultar el log de sesión.

9. Tarea de mantenimiento.

Sirve para eliminar los ficheros temporales cuya antigüedad sea superior a 2 días. Además si se incluye el parámetro `DiasLog` indica que elimine los ficheros de log cuya antigüedad sea superior a `DiasLog` días.

Acción a realizar	
/Mant:[DiasLog]	Elimina los ficheros temporales que tengan más de 2 días. Si se incluye el parámetro <code>DiasLog</code> eliminará los ficheros log de más de <code>DiasLog</code> días.